МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«____»_____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Аналіз та розробка програмного забезпечення системи екологічного моніторингу промислового регіону

Освітньо-кваліфікаційний рівень "Магістр"
Спеціальність 123 "Комп'ютерна інженерія"
(освітня програма - "Системне програмування")

Науковий керівник роботи: _____ (підпис) С.О. Сафонова (ініціали, прізвище)

Консультант з охорони праці: _____ (підпис) Я.О.Критська (ініціали, прізвище)

Студент: _____ (підпис) І.Л. Кончик (ініціали, прізвище)

Група: СП-16дм

Сєвєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерноих наук та інженерії
Освітньо-кваліфікаційний рівень "магістр"
Спеціальність 123 – "Комп'ютерна інженерія"
(шифр і назва)
Спеціалізація "Системне програмування"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри комп'ютерної інженерії
д.т.н., доц. *І.С. Скарга-Бандурова*
«_____» _____20____р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Кончику Іллі Леонідовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз та розробка програмного забезпечення системи екологічного моніторингу промислового регіону

керівник проекту (роботи) Сафонова Світлана Олександрівна, к.т.н., доцент
(прізвище, ім.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «18» 10 2017 р. №
2. Строк подання студентом роботи 20.01.2018
3. Вихідні дані до роботи Матеріали науково-дослідної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Огляд методів розрахунку основних показників екологічної системи. Вибір варіантів виконання програмного забезпечення серверу системи регіонального екологічного моніторингу для прогнозування витрат від забруднення атмосферного повітря. Дослідження методів та технологій розробки програмного забезпечення. Розробка програмного забезпечення серверу регіональної системи екологічного моніторингу для пронозування витрат від забруднення атмосферного повітря. Охорона праці та безпека в надзвичайних ситуаціях.Екологія

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	*Критська Я.О.*		

7. Дата видачі завдання *20.10.2017*

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Формування технічного завдання	*21.10.17-25.10.17*	
2	Збір необхідної інформації щодо екологічного стану регіону та пошук існуючих методів розрахунку	*26.10.17-03.11.17*	
3	Дослідження та аналіз існуючих методів вирішення завдання	*04.11.17-14.11.17*	
4	Дослідження та аналіз існуючих технологій вирішення завдання	*15.11.17-30.11.17*	
5	Розробка програмного забезпечення серверу регіональної системи екологічного моніторингу	*01.12.17-19.12.17*	
6	Охорона праці	*20.12.17-28.12.17*	
7	Оформлення пояснювальної записки	*02.01.18-18.01.18*	

Студент _____ *Кончик І.Л.*
(підпис) (прізвище та ініціали)

Науковий керівник _____ *Сафонова С.О.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кончик І.Л. Аналіз та розробка програмного забезпечення системи екологічного моніторингу промислового регіону.

Метою магістерської атестаційної роботи роботи є дослідження математичних аспектів інформаційного забезпечення екологічного моніторингу та розробка програмного забезпечення системи екологічного моніторингу промислового регіону. Результатом роботи є працездатний тестовий сервер регіональної системи екологічного моніторингу, що забезпечує реалізацію вимог завдання.

Ключові слова: модель, сервер, код, програмне забезпечення, екологічний моніторинг, база даних.

АННОТАЦИЯ

Кончик И.Л. Анализ и разработка программного обеспечения системы экологического мониторинга промышленного региона.

Целью магистерской аттестационной работы работы является исследование математических аспектов информационного обеспечения экологического мониторинга и разработка программного обеспечения системы экологического мониторинга промышленного региона. Результатом работы является работоспособный тестовый сервер региональной системы экологического мониторинга, который обеспечивает реализацию требований задания.

Ключевые слова: модель, сервер, код, программное обеспечение, экологический мониторинг, база данных.

ABSTRACT

Konchik I.L. Analysis and development of software for the system of environmental monitoring of the industrial region.

The aim of certification diploma is to study the mathematical aspects of information provision of environmental monitoring and the development of software for the environmental monitoring system of the industrial region. The result of the work is an efficient test server of the regional environmental monitoring system, which ensures the implementation of the requirements of the task.

Key words: model, server, code, software, ecological monitoring, database.

ЗМІСТ

ВСТУП

Обґрунтування вибору теми дослідження. Науково-технічна діяльність людства в кінці XX століття стала відчутним чинником дії на довкілля. Теплове, хімічне, радіоактивне і інші забруднення довкілля в останні десятиліття знаходяться під пильною увагою фахівців і викликають справедливу заклопотаність, а інколи - і тривогу громадськості.

По багатьом прогнозам проблема захисту довкілля в XXI столітті стане найбільш значимою для більшості промислово розвинених країн. У подібній ситуації налагоджена широкомасштабна і ефективна мережа контролю достатку довкілля, особливо в крупних містах і довкола екологічно небезпечних об'єктів, може з'явитися важливим елементом забезпечення екологічної безпеки і заставою стійкого розвитку суспільства.

У останні десятиліття суспільство все ширше використовує в своїй діяльності зведення про достаток природної середи. Ця інформація потрібна в повсякденному житті людей, при господарюванні, в будівництві, при надзвичайних обставинах - для сповіщення про небезпечні явища, що насуваються, природи. Але зміни в стані довкілля відбуваються і під впливом біосферних процесів, пов'язаних з діяльністю людини. Визначення вкладу антропогенних змін є специфічним завданням.

Вже більше 100 років спостереження за зміною погоди, кліматом ведуться регулярно на цивілізованому світі. Все ширше стає круг спостережень, число вимірюваних параметрів, все густіше мережа наглядових станцій. Все більшою складністю володіють проблеми, пов'язані з моніторингом довкілля.

Все вищенаведене свідчить про те, що проблема спостереження за зміною погоди, моніторингом довкілля є актуальною на даний момент.

Тому обґрунтованою є тема магістерської роботи.

Об'єктом дослідження магістерської роботи є можливість створення розрахованого на багато користувачів програмного забезпечення, що здійснює збір інформації, обробку та дозволяє отримати об'єктивну оцінку про екологічний стан міста і регіону в цілому у будь-який час..

Предметом дослідження є методи та технології розробки програмних систем екологічного моніторингу промислового регіону.

Мета і задачі дослідження. Метою роботи є дослідження математичних аспектів інформаційного забезпечення екологічного моніторингу та розробка програмного забезпечення системи екологічного моніторингу промислового регіону. Для досягнення поставленої мети в роботі сформульовані та вирішені наступні задачі:

- дослідження та аналіз математичних аспектів інформаційного забезпечення екологічного моніторингу, методів розрахунку екологічних показників та технологій розробки програмних систем;
- розробка алгоритмів забезпечення безпечної передачі інформації в системі, статистичного та екологічного аналізу, побудови графіків, підсистеми ухвалення рішень;
- визначення недоліків і переваг існуючих методів та технології розробки ПЗ; обґрунтування варіантів вирішення завдання, виходячи з результатів досліджень;
- розробка програмного забезпечення екологічного сервера.

Методи рішення поставлених задач базуються на комплексному використанні математичного апарату теорії ймовірностей, методів імітаційного моделювання, послідовної графічної регресії, теорії статичних ігор з ієрархічною структурою.

Наукова новизна одержаних результаті:.

- одержали подальший розвиток методи розрахунку екологічних показників;
- модифіковано існуючі методи статистичного та екологічного аналізу.

Апробація результатів роботи. Основні результати магістерської атестаційної роботи докладалися на VIII всеукраїнської науково-практичої конференції «Майбутній науковець – 2017» (м. Сєвєродонецьк).

Практичне значення отриманих результатів:

- запропоновані алгоритми забезпечення безпечної передачі інформації в системі, побудови графіків;
- модуль підтримує збір та обробку оперативної інформації про стан екології в місті і відображення її в зручному вигляді

Публікації. Основні результати магістерської атестаційної роботи опубліковано в 2 наукових працях: 2 матеріалах конференцій.

Структура і обсяг роботи.

Магістерська робота складається зі вступу, 5 розділів, висновків на 1 сторінці, списку використаних джерел з 12 найменувань на 1 сторінкі, додатків на 75 сторінках. Загальний обсяг роботи складає 197 сторінки. В магістерській роботі міститься 5 таблиці, 64 рисунки.

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ТА МЕТОДИ РОЗРАХУНКУ ОСНОВНИХ ПОКАЗНИКІВ ЕКОЛОГІЧНОЇ СИСТЕМИ

В Україні існує досить розгалужена система екологічного дослідження стану навколишнього природного середовища. На рівні держави, регіонів та міст інформація про стан довкілля за певний період формується різними державними установами та підпорядкованими їм підприємствами.

Розподіл функцій моніторингу по різних відомствах, які не зв'язані між собою, призводить до дублювання зусиль, знижує ефективність усієї системи моніторингу та затрудняє доступ до необхідної інформації як для громадян, так і для державних організацій. Тому в Україні було прийняте рішення про створення Державної системи моніторингу довкілля (ДСМД), яка повинна об'єднати можливості і зусилля різноманітних служб для вирішення задач комплексного спостереження, оцінки та прогнозу стану довкілля в Україні.

Державна система моніторингу довкілля - це система спостережень, збирання, оброблення, передавання, збереження та аналізу інформації про стан довкілля, прогнозування його змін і розроблення науково-обґрунтованих рекомендацій для прийняття рішень про запобігання негативним змінам стану довкілля та дотримання вимог екологічної безпеки.

Екологічний моніторинг довкілля здійснюється за довгостроковою Державною програмою, яка визначає спільні, узгоджені за цілями, завданнями, територіями та об'єктами, часом і засобами виконання дії відомчих органів державної виконавчої влади, підприємств, організацій та установ незалежно від форм власності.

Створення і функціонування Державної системи екологічного моніторингу довкілля повинно сприяти здійсненню державної екологічної політики, яка передбачає: екологічно раціональне використання природного та соціально-економічного потенціалу держави, збереження сприятливого середовища життєдіяльності суспільства; соціально-екологічне та економічно раціональне вирішення проблем, які виникають в результаті забруднення довкілля, небезпечних природних явищ, техногенних аварій та катастроф; розвиток міжнародного співробітництва щодо збереження біорізноманіття природи, охорони озонового шару атмосфери, запобігання антропогенній зміні клімату, захисту лісів і лісовідновлення, транскордонного забруднення довкілля, відновлення природного стану Дніпра, Дунаю, Чорного і Азовського морів.

Функціонування Державної системи екологічного моніторингу повинне більш якісно розвивати принципи: систематичності спостережень за станом довкілля та техногенними об'єктами, що впливають на нього; своєчасності отримання і оброблення даних спостережень на відомчих та узагальнюючих (локальному, регіональному та державному) рівнях; комплексності використання екологічної інформації, що надходить у систему від відомчих служб моніторингу й інших постачальників; об'єктивності первинної, аналітичної і прогнозної екологічної інформації та узгодженості нормативного, організаційного і методичного забезпечення екологічного моніторингу довкілля, що проводиться відповідними службами міністерств та відомств України, інших центральних органів виконавчої влади; сумісності технічного, інформаційного і програмного забезпечення її складових частин; оперативності доведення екоінформації до органів виконавчої влади, інших зацікавлених органів, підприємств, організацій та установ; доступності екологічної інформації населенню України та світовій спільноті.

1.1 Мета екологічного дослідження

Основною метою дослідження є забезпечення системи управління природоохоронної діяльності і екологічної безпеки достовірною інформацією, що дозволяє:

а) оцінити достаток місця існування людини, біологічних співтовариств;

б) виявити причини відхилення показників;

в) оцінити наслідки зміни показників;

г) визначити рішення, що управляють, для ліквідації причин відхилення показників;

д) прогнозувати витрати від забруднення атмосферного повітря.

Екологічний моніторинг має бути орієнтований на три основні показники:

а) дотримання встановлених національних і міжнародних вимог до антропогенної дії;

б) діагностика антропогенної дії;

в) запобігання наслідкам антропогенної дії.

1.2 Основі завдання екологічного дослідження

Основні завдання екологічного дослідження:

а) спостереження за джерелами антропогенної дії;

б) спостереження за чинниками антропогенної дії;

в) спостереження за достатком забруднення природної середи;

г) оцінка достатку забруднення природної середи;

д) прогноз забруднення природної середи, об'єктів природної середи;

е) інформаційна підтримка розробки і реалізації заходів по своєчасному прогнозуванню, виявленню і запобіганню погрозам і кризисним ситуаціям відносно можливого забруднення довкілля.

Об'єктом дослідження є екологічне перебування і екологічна обстановка на території спостереження.

Екологічний моніторинг може мати різний масштаб і охоплювати:

а) підприємство;

б) населений пункт;

в) територію і так далі

Інформаційні потоки, необхідні для здійснення екологічного моніторингу:

а) джерела вступу забруднюючих речовин в природне довкілля;

б) процеси перенесення і міграції забруднюючих речовин в природній середі;

в) достаток здоров'я людини;

г) відгук біологічних співтовариств на антропогенну дію.

Моніторингом довкілля називають регулярні, виконувані за заданою програмою спостереження природної середи, природних ресурсів, рослинного і тваринного миру, що дозволяють виділити їх достатки і процеси, що відбуваються в них, під впливом антропогенної діяльності.

Під екологічним моніторингом слід розуміти організований моніторинг природного довкілля, при якому, по-перше, забезпечується постійна оцінка екологічних умов місця існування людини і біологічних об'єктів (рослин, тварин, мікроорганізмів і т. д.), а також оцінка достатку і функціональної цінності екосистем, створюються умови для визначення дій, що коректують, в тих випадках, коли цільові показники екологічних умов не досягаються.

Відповідно до приведених визначень і покладених на систему функцій, моніторинг включає декілька основних процедур: виділення (визначення) об'єкту спостереження;

обстеження виділеного об'єкту спостереження; складання інформаційної моделі для об'єкту спостереження; планерування вимірів; оцінка достатку об'єкту спостереження і ідентифікації його інформаційної моделі; прогнозування зміни достатку об'єкту спостереження; представлення інформації в зручній для користувача формі і доведення її до споживача.

Слід взяти до уваги, що сама система моніторингу не включає діяльність по управлінню якістю середи, але є джерелом необхідною для прийняття екологічно значимих вирішень інформації.

Система екологічного моніторингу повинна нагромаджувати, систематизувати і аналізувати інформацію: про достаток довкілля; про причини спостережуваних і вірогідних змін достатку (тобто про джерела і чинники дії); про допустимість змін і навантажень на середу в цілому; про існуючі резерви біосфери.

Таким чином, в систему екологічного моніторингу входять спостереження за достатком елементів біосфери і спостереження за джерелами і чинниками антропогенної дії.

Відповідно до приведених визначень і покладених на систему функцій, моніторинг включає три основні напрями діяльності: спостереження за чинниками дії і достатком середи; оцінку фактичного достатку середи; прогноз достатку природного довкілля і оцінку прогнозованого достатку.

Слід взяти до уваги, що сама система моніторингу не включає діяльність по управлінню якістю середи, але є джерелом необхідною для прийняття екологічно значимих вирішень інформації.

Основні завдання екологічного моніторингу: спостереження за джерелами антропогенної дії; спостереження за чинниками антропогенної дії; спостереження за достатком природної середи і процесами, що відбуваються в ній , під впливом чинників антропогенної дії; оцінка фактичного достатку природної середи; прогноз зміни достатку природної середи під впливом чинників антропогенної дії і оцінка прогнозованого достатку природної середи.

Екологічні моніторинги довкілля можуть розроблятися на рівні промислового об'єкту, міста, області, краю, республіки у складі федерації. У зоні впливу джерел емісії організовується систематичне спостереження за наступними об'єктами і параметрами природного довкілля.

Атмосфера: хімічний і радіонуклідний склад газової і аерозольної фази повітряної сфери; тверді і рідкі осідання (сніг, дощ) і їх хімічний і радіонуклідний склад; теплове і влагостне забруднення атмосфери.

Системи моніторингу природної середи і екосистем включають засоби спостереження: екологічної якості повітряної середи, екологічного достатку поверхневих вод і водних екосистем, екологічного достатку геологічної середи і наземних екосистем.

Спостереження в рамках цього вигляду моніторингу проводяться без врахування конкретних джерел емісії і не пов'язані із зонами їх впливу. Основний принцип організації - природно-экосистемний.

Цілями спостережень, що проводяться в рамках моніторингу природної середи і екосистем, є:

1.1 оцінка достатку і функціональної цілісності місця існування і екосистем;

1.2 виявлення змін природних умов в результаті антропогенної діяльності на території;

1.3 дослідження змін екологічного клімату (багатолітнього екологічного достатку) територій;

1.4 основні принципи моніторингу

Відомо, що під моніторингом довкілля мають на увазі регулярні, виконувані за заданою програмою спостереження природної середи, природних ресурсів, рослинного і тваринного, що дозволяють оцінити достаток і зміни, що відбуваються, під впливом антропогенної діяльності. Моніторинг за своєю суттю є системою, що включає спостереження, оцінку спостереження, прогноз, оцінку прогнозу, дозволяє правильно управляти якістю природної середи.

Під екологічним моніторингом слід розуміти організований моніторинг об'єктів довкілля для забезпечення оцінки місця існування людини, біологічних співтовариств і екологічних систем з метою ухвалення управлінських рішень, коли показники достатку одного або декількох об'єктів не досягаються.

Основні принципи організації моніторингу: комплексність, систематичність, уніфікована.

Процедури моніторингу:

а) виділення об'єкту спостережень;

б) обстеження виділеного об'єкту;

в) складання інформаційної моделі для об'єкту спостережень;

г) планерування вимірів;

д) оцінка достатку об'єкту і ідентифікації його інформаційної моделі;

е) прогнозування достатку зміни об'єкту спостережень;

ж) представлення інформації в зручній для використання формі.

Основна середа моніторингу:

а) населення, демографічні чинники, здоров'я, соціально - економічні чинники;

б) атмосфера, всі види забруднення;

в) гідросфера, всі види забруднення;

г) ґрунти;

д) біота;

е) урбанізована середа.

Види забруднення:

а) хімічне;

б) радіоактивне;

в) теплове;

г) електромагнітне;

д) шумове.

1.3 Функції екологічного моніторингу

Система моніторингу повинна забезпечувати виконання наступних функцій:

а) збір, обробка, аналіз, зберігання і передача інформації про місце розташування, узагальнені параметри екологічного достатку і інших необхідних даних;

б) інформаційна підтримка робіт, що виконуються в цілях підготовки і реалізації заходів по забезпеченню безпечного функціонування об'єктів,попередженню і локалізації кризисних ситуацій, а також ліквідації їх наслідків;

в) підготовка інтегральних оцінок (моделей) кризисних ситуацій відносно об'єктів екологічного моніторингу і оцінка їх можливих наслідків;

г) прогнозування погроз забруднення і динаміки зміни достатку вірогідності цього під впливом природних, техногенних і інших чинників;

д) ведення інформаційних баз даних для забезпечення підтримки прийняття і реалізації управлінських рішень по захисту об'єктів моніторингу і природної середи вцілому;

е) надання в установленому порядку інформаційних ресурсів системи моніторингу, забезпечення захисту цих ресурсів від несанкціонованої дії;

ж) формування єдиного інформаційного простору системи моніторингу на основі уніфікації і сумісності інформаційних, програмних і апаратних засобів;

з) інформаційне забезпечення реалізації міжнародних договорів і угод в області об'єктів.

1.4 Склад та структура екологічного моніторингу

Система моніторингу передбачає спільний, міжрегіональний, регіональний, муніципальний і об'єктовий рівні.

При цьому до складу системи моніторингу кожного рівня мають бути включені:

а) центри системного моніторингу і оперативного управління (далі-центри моніторингу);

б) системи, комплекси і засоби здобуття інформації про узагальнені параметри достатку захищеності об'єктів моніторингу;

в) системи і засоби телекомунікації, збору, передачі даних і сповіщення.

Основними структурними елементами системи моніторингу, що забезпечують рішення покладеної на неї задачі, повинні стати центри моніторингу органів виконавчої влади і органів місцевої самоврядності.

Об'єднання інформаційних ресурсів центрів моніторингу різних рівнів здійснюється з використанням систем і засобів телекомунікації, при цьому враховується необхідність забезпечення конфіденційності інформації і санкціонованого видаленого доступу до їх баз даних.

При рішенні покладеної на систему моніторингу задачі має бути передбачена можливість інформаційної взаємодії центрів моніторингу різних рівнів з іншими державними і недержавними інформаційними системами спільного і спеціального призначення, а також з міжнародними інформаційними системами.

При створенні і використанні системи моніторингу необхідно керуватися наступними основними принципами:

а) забезпечення відповідності вирішуваною системою моніторингу завдання а також її структури і характеристик: рівню погроз відносно об'єктів моніторингу; структурі і завданням органів виконавчої влади, органів місцевої самоврядності і охорони довкілля;

б) організаційна, інформаційна і функціональна єдність системи моніторингу, основу якого складають: єдина система класифікації і кодифікування

погроз об'єктам моніторингу, показників і критеріїв оцінки;базові (типові) протоколи, алгоритми (програми) збору обробки і обміну інформацією, підготовки і автоматизованою підтримка прийняття і реалізації управлінських рішень на основі даних моніторингу; єдина геоінформаційна система;

в) ієрархічність побудови системи моніторингу, можливість централізованого і санкціонованого децентралізованого використання ресурсів системи моніторингу;

г) раціональна функціональна сумісність центрів моніторингу різних рівнів;

д) уніфікація програмних, інформаційних і технічних засобів забезпечення сумісності елементів системи моніторингу, можливості її модульного нарощування і модернізації;

е) можливість структурного і функціонального розвитку оптимізації складу користувачів системи моніторингу і спектру послуг, що надаються;

ж) багатофункціональність, що забезпечує одночасне рішення завдань на користь національної безпеки і соціально-економічного розвитку країни;

з) спадкоємність, заснована на інтеграції і вдосконаленні інших систем моніторингу;

и) гарантований захист інформації від несанкціонованого доступу, включаючи обмежений доступ до циркулюючої в системі моніторингу інформації;

к) недопущення залежності системи моніторингу від іноземних технологій.

1.5 Загальні напрямки роботи в галузі розробки, користування та розвитку системи моніторингу

Загальними напрямками робіт в галузі створення використання і розвитку системи моніторингу є:

а) організаційне і фінансово-економічне забезпечення системи моніторингу;

б) вдосконалення нормативно-правовий бази;

в) створення і впровадження перспективних науково-технічних розробок.

В області організаційного і фінансово-економічного забезпечення системи моніторингу необхідно вирішити наступні завдання:

а) розробка техніко-економічного обґрунтування заходів щодо створенню і вживанню системи моніторингу, комплексній ув'язці її складових частин;

б) аналіз інших систем моніторингу, забезпечення уніфікації технічних і організаційних рішень;

в) вироблення пропозицій по необхідних об'ємах і джерелах фінансових коштів (бюджетних і позабюджетних), їх виділенню (залученню) на реалізацію програм і планів заходів щодо створення використанню і розвитку системи моніторингу;

г) включення робіт із створення і розвитку системи моніторингу в загальнодержавні і регіональні цільові програми;

д) формування інституту загальнодержавних і регіональних операторів робіт (послуг) в області моніторингу;

е) розробка і реалізація механізмів залучення недержавних фінансових, матеріальних і інших ресурсів для вирішення інноваційних і інвестиційних завдань в області екологічного моніторингу.

Основними напрямами діяльності в області вдосконалення нормативно-правовий бази є:

а) забезпечення розробки проектів і прийняття нормативних правових актів держави в області моніторингу, у тому числі у відношенні встановлення відповідальності власників (балансоутримувачів) об'єктів моніторингу і експлуатуючих їх організацій за недотримання встановлених заходів по забезпеченню здійснення моніторингу;

б) гармонізація нормативно-правовий бази в області екологічного моніторингу з нормами міжнародного права і міжнародними договорами (угодами) в цій області;

в) введення норм, що передбачають обов'язкове оснащення об'єктів моніторингу технічними системами, комплексами і засобами моніторингу їх достатку (місця розташування).

Основними напрямами діяльності в області створення і впровадження перспективних науково-технічних розробок є:

а) підготовка і впровадження технічних регламентів для різних видів діяльності відносно об'єктів моніторингу, що забезпечують виконання вимог по організації і проведенню їх моніторингу;

б) відробіток типових (уніфікованих) технічних і організаційних вирішень створення і використання системи моніторингу в рамках регіональних цільових програм;

в) розробка єдиної системи критеріїв і комплексних методик аналізу узагальнених параметрів достатку захищеності об'єктів моніторингу.

г) забезпечення інформаційного обміну між центрами моніторингу різних рівнів;

д) створення в рамках системи моніторингу спеціалізованих засобів збору і передачі інформації, що функціонують на всій території України;

е) розвиток систем дистанційного моніторингу;

ж) створення мобільних (перебазуємих) центрів моніторингу що забезпечують інформаційну підтримку діяльності, здійснюваної при виникненні кризисних ситуацій;

з) формування вітчизняної науково-технічної кооперації в області створення і використання системи моніторингу, у тому числі визначення головних організацій, об'єднуючих розробників і виробників технічних систем, комплексів і засобів моніторингу, а також системи моніторингу.

1.6 Класифікація екологічного моніторингу

Існують різні підходи до класифікації моніторингу (по характеру вирішуваних завдань, по рівнях організації, по природній середі, за якою ведуться спостереження). Відображена на рисунку 1.1 класифікація охоплює весь блок екологічного моніторингу, спостереження за змінної абіотичної біосфери, що становить, і у відповідь реакцією екосистем на ці зміни.

Таким чином, екологічний моніторинг включає як геофизичні, так і біологічні аспекти, що визначає широкий спектр методів і прийомів досліджень, використовуваних при його здійсненні.

Рисунок 1.1 - Класифікація екологічного моніторингу

Як вже було відмічено, здійснення екологічного моніторингу входить в обов'язки різних державних служб. Це приводить до деякої невизначеності (принаймні, для громадськості) відносно розподілу обов'язків держслужби і доступності відомостей про джерела дії, про достаток довкілля і природних ресурсів. Ситуацію посилюють періодичні перебудови міністерств і відомств, їх злиття і розділення.

На регіональному рівні екологічний моніторинг і контроль зазвичай ставиться в обов'язок: Комітету з екології (спостереження і контроль за викидами і скиданнями підприємств, що діють). Комітету з гідрометеорології і моніторингу (импактний, регіональний і частково фоновий моніторинг) Санітарно-епідеміологічній службі Мінздорову (достаток робітників,рекреаційних зон, якість питної води і продуктів харчування). Міністерству природних ресурсів (перш за все, геологічні і гідрогеологічні спостереження).

Підприємствам, що здійснюють викиди і скидання в довкілля (спостереження і контроль за власними викидами і скиданнями). Різним відомчим структурам (підрозділам Мінсільгосппроду, Мінпаленерго, підприємствам водноканалізаційного господарства і ін.)

1.7. Екологічна експертиза

Екологічна експертиза може бути державною і суспільною. Суспільна екологічна експертиза проводиться за ініціативою громадян і громадських організацій (об'єднань), а також за ініціативою органів місцевої самоврядності громадськими організаціями (об'єднаннями).

Об'єктами державної екологічної експертизи є: проекти генеральних планів розвитку територій, всі види містобудівної документації (наприклад, генеральний план, проект забудови), проекти схем розвитку галузей народного господарства, проекти міждержавних інвестиційних програм, проекти комплексних схем охорони природи, схем охорони і використання природних ресурсів (в т.ч. проекти землекористування і лісовпорядження, матеріали, що обґрунтовують переклад лісових земель в нелісових), проекти міжнародних договорів, матеріали обґрунтування ліцензій на здійснення діяльності, здатної надати дію на довкілля, техніко-економічні обґрунтування і проекти будівництва, реконструкції, розширення, технічного переозброєння, консервації і ліквідації організацій і інших об'єктів господарської діяльності, незалежно від їх кошторисної вартості, відомчої приналежності і форм власності, проекти технічної документації на нову техніку, технологію, матеріали, речовини, товари, що

сертифікуються, і послуги.Суспільна екологічна експертиза може проводитися відносно тих же об'єктів, що і державна екологічна експертиза, за винятком об'єктів, відомості про яких складають державну, комерційну і (або) іншу таємницю, що охороняється законом.

Метою екологічної експертизи є запобігання можливим несприятливим діям наміченої діяльності на довкілля і пов'язаних з ними соціально-економічних і інших наслідків.

1.8. Рівні екологічного моніторингу

На рисунку 1.2 відображено рівні екологічного моніторингу. В ідеальному випадку система импактного моніторингу повинна нагромаджувати і аналізувати детальну інформацію про конкретні джерела забруднення і їх дію на довкілля. Але в системі зведення про діяльність підприємств, що склалася в країні, і про полягання середи в зоні їх дії здебільше усереднені або засновані на заявах самих підприємств. Велика частина доступних матеріалів відображає характер розсіяння забруднюючих речовин в повітрі і у воді, встановлений за допомогою модельних розрахунків, і результати вимірів (щоквартальних - по воді, щорічних або рідших - по повітрю). Достаток довкілля достатній повно описується лише в крупних містах і промислових зонах. В області регіонального моніторингу спостереження ведуться в основному, що має розгалужену мережу, а також деякими відомствами (агрохімслужба Мінсільгосппроду, водно-каналізаційна служба і ін.) І, нарешті, існує мережа фонового моніторингу.

Рисунок 1.2 – Рівні екологічного моніторингу

Практично не охопленими мережею спостережень залишаються малі міста і багато чисельні населені пункти, переважна більшість дифузних джерел забруднення. Моніторинг достатку водної середи, організований, перш за все, до деякої міри, санітарно-епідеміологічними (СЕС) і комунальними (Водоканал) службами, не охоплює переважну більшість малих річок. В той же час, забруднення великих річок в значній частині обумовлене вкладом розгалуженої мережі їх припливів і господарською діяльністю у водозборі. В умовах скорочення спільного чисел; постів спостережень очевидно, що держава в даний час не має в своєму розпорядженні ресурсів для організації скільки-небудь ефективної системи моніторингу достатку малих річок.

Система моніторингу реалізується на декількох рівнях, яким відповідають спеціально розроблені програми:

а) импактному (вивчення сильних дій локальному масштабі в- І); регіональному (прояв проблем міграції і трансформації забруднюючих речовин, спільної дії різних чинників, характерних для економіки регіону, - Р);

б) фоновому (на базі біосферних заповідників, де виключена всяка господарська діяльність - Ф).

Програма импактного моніторингу може бути направлена, наприклад, на вивчення скидань або викидів конкретного підприємства. Предметом регіонального моніторингу, як випливає з самого його назви, є достаток довкілля в межах того або іншого регіону. Нарешті, фоновий моніторинг, здійснюваний в рамках міжнародної програми Людина і біосфера, має на меті зафіксувати фоновий достаток довкілля, що необхідне для подальших оцінок рівнів антропогенної дії.

Програми спостережень формуються за принципом вибору забруднюючих речовин і відповідними ним характеристиками. Визначення цих забруднень при організації систем моніторингу залежить від мети і завдань конкретних програм: так, в територіальному масштабі пріоритет державних систем моніторингу відданий містам, джерелам питної води і місцям нерестовищ риб; відносно середи спостережень першочергової уваги заслуговують атмосферне повітря і вода прісних водоймищ. Пріоритетність інгредієнтів визначається з врахуванням критеріїв, що відображають токсичні властивості забруднюючих речовин, об'єми їх вступу в довкілля, особливості їх трансформації, частоту і величину дії на людину і біоту, можливість організації вимірів і інші чинники.

На рисунку 1.3 відображено ієрархічна структура потоку інформації.

Рисунок 1.3 – Ієрархічна структура потоку інформації

Загальні задачі що мають бути вирішені:

а) розробка програм спостережень за перебуванням природного довкілля на території України, в її окремих регіонах і районах; організація спостережень і проведення вимірів показників об'єктів екологічного моніторингу; забезпечення достовірності і порівнянності даних спостережень як в окремих регіонах і районах, так і по всій території України; збір і обробка даних спостережень; організація зберігання даних спостережень, ведення спеціальних банків даних, що характеризують екологічну обстановку на території України і в окремих її районах; гармонізація банків і баз екологічної інформації з міжнародними эколого-інформаційними системами; оцінка і прогноз антропогенних дій на них, природних ресурсів, відгуків екосистем і здоров'я населення; організація і проведення оперативного контролю і прецизійних змін радіоактивного і хімічного забруднення в результаті аварій і катастроф, а також прогнозування екологічної обстановки і оцінка нанесеного збитку;

б) забезпечення доступності інтегрованої екологічної інформації широкому кругу споживачів, включаючи населення, суспільні рухи і організації; інформаційне забезпечення органів управління, природних ресурсів і екологічною безпекою;

в) розробка і реалізація єдиною науково технічної політики в області екологічного моніторингу;

г) створення і вдосконалення організованого, правового, нормативного, методологічного, методичного, інформаційного, програмно-математичного, апаратурно-технічного забезпечення функціонування;

д) Повинно бути включено наступні основні компоненти: моніторинг джерел антропогенної дії на довкілля; моніторинг забруднення абіотичного компонента природного довкілля;

е) моніторинг біотичного компонента природного довкілля; соціально-гігієнічний моніторинг;

ж) забезпечення створення і функціонування екологічних інформаційних систем.

При організації довготривалого моніторингу особлива увага приділяється принципу уніфікації методів аналізу і контролю і забезпеченню якості даних. Далі ми детально охарактеризуємо кожен з цих принципів. Слід звернути увагу, що при проведенні комплексного дослідження використовуються не лише чисто екологічні знання і методи, але також знання і методи географії, геофізики, аналітичної хімії, програмування і ін.

1.9. Комплексна характеристика стану забруднення навколишнього середовища

Для того, щоб скласти попередню комплексну характеристику забруднення території, немає необхідності в довготривалому моніторингу. Поважно, щоб при проведенні дослідження, враховувалися основні вимоги і принципи, на яких будується концепція комплексності дослідження.

Принципи комплексної характеристики достатку забруднення природної середи. Комплексна характеристика достатку забруднення виходить з концепції все стороннього аналізу довкілля. Головною і обов'язковою умовою цієї концепції є розгляд всіх основних сторін взаємодій і зв'язків в природній середі і облік всіх аспектів забруднення природних об'єктів, а також поведінка забруднюючих речовин (ЗР) і прояву їх дії. При комплексній характеристиці забруднень ЗР відстежуються у всій середі, при цьому велике значення надається вивченню накопичення (акумуляція) того або іншого ЗР в природних об'єктах або певних ландшафтах, його переходу (транс локації) з однієї природної середи в іншу і змін, що викликаються під його дією (ефектів). Комплексні дослідження забруднень, що проводяться, покликані визначити джерело забруднення, оцінити його потужність і час дії і знайти дороги оздоровлення середи. Підхід, що враховує перераховані вимоги, прийнято вважати комплексним.

У зв'язку з цим, виділяють 4 основних принципу комплексності:

а) Інтегральність (спостереження за сумарними показниками);

б) Багатосередовищність (спостереження в основній природній середі);

в) Системність (відтворення біохімічних циклів забруднюючих речовин);

г) Багатокомпонентність (аналіз різних видів забруднюючих речовин).

При організації довготривалого моніторингу особлива увага приділяється п'ятому принципу - уніфікації методів аналізу і контролю і забезпеченню якості даних. Далі ми детально охарактеризуємо кожен з цих принципів.

Слід звернути увагу, що при проведенні комплексного дослідження використовуються не лише чисто екологічні знання і методи, але також знання і методи географії, геофізики, аналітичної хімії, програмування і ін.

а) Інтегральність

Особливість інтегрального підходу полягає у використанні для визначення наявності забруднень ознак реакцій різних природних об'єктів і біоіндикаторів.

Потрапляючи в незнайому місцевість, наглядова людина, а особливо натураліст, може по непрямих межах визначити полягання забруднення в даній місцевості. Неприродний запах, задимленість горизонту, сірий лютневий сніг, веселкова плівка на поверхні водоймища і багато інших меж підкажуть спостерігачеві підвищене промислове забруднення місцевості. У наведеному прикладі індикаторами достатку забруднення місцевості є неживі (абіотичні) об'єкти - приземне повітря, поверхня снігового покриву і водоймища. Найширше як абіотичного індикатор промислового забруднення території використовується сніговий покров і метод його вивчення - снігомірна зйомка.

При використанні інтегрального підходу особлива увага приділяється достатку живих організмів.

Так, відомо, що до забруднення повітря в нашій смузі найуразливішою виявляється сосна. При високому рівні забруднення повітря оксидами сірки, азоту і іншими токсичними з'єднаннями спостерігається спільне освітлення забарвлення хвої, суховершиність, пожовтіння країв хвоїнок. У підліску засихає ялівець. Через декілька годинників після кислотного дощу краю листя берези жовтіють, листя покривається сіро-жовтим нальотом мул цятками. При великій кількості оксидів азоту в повітрі на стволах дерев бурхливо розвиваються водорості, при цьому зникають епіфітні кущисті лишайники і так далі Наявність широкопалих рак у водоймищі свідчить про високу чистоту води.

Метод використання живих організмів як індикатори, що сигналізують про достаток природної середи, називається біоіндикацією, а сам живий організм, за достатком якого проводяться спостереження, називають біоіндикатором. У наведених вище прикладах біоіндикаторами служили живі об'єкти - береза, сосна, ялівець, епіфітні лишайники, широкопалі раки.

Використання біоіндикаторів засноване на реакції будь-якого біологічного організму на негативну дію. При цьому, набір реакцій на множинну, інтегральну, негативну дію навколишнього середовища, як правило, вельми обмежений. Організм або гине, або покидає (якщо може) дану місцевість, або тягне жалюгідне існування, що можна визначити візуально або з використанням різних тестів і серії спеціальних спостережень (методикам біоіндикації присвячено декілька посібників даної серії).

Підбір і використання біоіндикаторів - цілком в руслі экологічної науки, а біоіндикація - що інтенсивно розвивається в метод дослідження результатів дій. Так, наприклад, при спостереженнях за якістю повітря широко використовуються різні рослини. У лісі, в кожному ярусі, можна виділити певні види рослин, що реагують по своєму на достаток забруднення середи.

Таким чином, інтегральний підхід полягає у використанні природних об'єктів як індикатори забруднення середи.

При цьому, частенько, буває абсолютно неясно, яке конкретне речовина була причиною того або іншого ефекту і робити виводи про пряму залежність між виглядом-індикатором і забруднюючою речовиною не можна. Особливість інтегрального підходу полягає саме в тому, що той або інший об'єкт-індикатор лише сигналізує нам, що в даній місцевості щось не в порядку. Використання біоіндикаторів для характеристики достатку забруднення дозволяє ефективно (тобто швидко і дешево) визначати наявність спільної, інтегральної дії забруднення на середу і складати лише попередні уявлення про хімічну природу забруднення. На жаль, точно визначати хімічний склад забруднюючих речовин за допомогою методів біоіндикації не можна. Для того, щоб конкретно визначити, яка речовина або група речовин надає найбільш згубну дію, необхідно використовувати інші методи дослідження. Точне визначення вигляду впливаючого ЗР, його джерела і масштабів забруднення і поширення неможливо без проведення аналітичних довготривалих досліджень у всій природній середі;

б) Багатосередовищність

При проведенні моніторингових досліджень важливий обхват всієї основної природної середи: атмосфери, гідросфери, літосфери (головним чином ґрунтового покриву - сфери), а також біоти. Для аналізу міграцій ЗР, визначення місць їх локалізації і акумуляції і визначення лімітуючої середи необхідне проведення вимірів в об'єктах основної природної середи.

Особливо поважно визначити лімітуючу середу, тобто середу, забруднення якої визначає забруднення всієї іншої середи і природних об'єктів. Також вельми поважно

визначити дороги міграції ЗР і можливості і коефіцієнти переходу (транслокації) ЗР з однієї середи (або об'єкту) в іншу. Цим займається наука геофізика.

Основна середа (об'єкти), яка має бути охоплені при проведенні комплексного дослідження: повітря, ґрунт (як частина літосфери), поверхневі води і біота. Забруднення кожною з цієї середи характеризується за результатами аналізів ЗР в різних об'єктах в межах цієї середи, вибір якої має важливе значення для отримуваних результатів і виводів. Щоб отримати зведення про забруднення певного об'єкту потрібно відібрати пробу для аналізу. Основні принципи, якими необхідно керуватися при виборі об'єкту і відборі проб охарактеризовані нижче;

1) Атмосфера.

Основним об'єктом, по якому характеризується забруднення атмосфери є приземний шар повітря. Проби повітря для аналізу відбираються на рівні 1,5 - 2 м-коду від поверхні землі. Відбір проби повітря полягає, зазвичай в його прокачуванні через фільтри, сорбент (єднальна речовина) або вимірювальний пристрій. Особливі вимоги пред'являються до майданчика відбору. По-перше, майданчик має бути відкритим і видаленим більш ніж на 100 м-кодів від лісу. Виміру під запоною лісу дають, як правило, занижений результат і більш характеризують щільність крон, чим рівень забруднення повітря. Опосередковано про якість повітря можна судити по забрудненню атмосферних опадів (головним чином - снігу і дощу). Осідання відбирають, використовуючи великі воронки, спеціальні опадозбірники або просто тази, лише у момент їх випадання і в точці відбору проб повітря. Інколи для характеристики забруднення повітря використовують проби сухих випадань, тобто твердих часток пилу, що постійно осідають на підстилаючу поверхню. Методично це досить складне завдання, яке, проте, досить просто вирішується методом снігомарної зйомки;

1.10. Математичні аспекти інформаційного забезпечення екологічного моніторингу

У завданнях автоматизованого контролю і управління найбільш важливою є формалізація опису як наочної області, так і процесів управління.

Розглянемо математичну постановку завдання планування природоохоронних заходів. Хай деякий природо користувач (підприємство), що знаходиться на території, підконтрольній системі екологічного моніторингу (СЕМ), має N стаціонарних джерел дії на довкілля (ВОС) з рівнем дії Vij (i - індекс джерела, j - індекс инградиєнта), що

перевищує норму Vnij, -, встановлену в результаті рішення задачі нормування, і проводить на даних джерелах N природоохоронні заходи щодо досягнення норм ВОС.

Дії природокористувача регламентуються планом проведення заходів, погодженим з вимогою моніторингу. У плані заданий нормативний термін виконання всіх заходів T (плановий період), а також терміни виконання кожного із заходів(формула 1.1).

$$t_i^s, t_i^t, i = \overline{1, N} \tag{1.1}$$

Приймемо, що еколого-економічна ефективність заходів обґрунтована, а проект проведення заходів пройшов екологічну експертизу, тобто можна вважати, що задані наступні еколого-економічні характеристики кожного із заходів:

Ki - капітальні витрати, необхідні для його реалізації;

q0i - експлуатаційні витрати в одиницю часу;

τ_i - час проведення заходу відповідно до існуючих нормативів будівництва об'єктів;

c0i(t) - функція освоєння капіталовкладень (задана на тимчасовій шкалі відносно моменту початку проведення заходів, $t \in [0, \tau]$ визначається виходячи з інженерно-технічної специфіки проведення заходу);

Vdij — екологічна ефективність заходу (формула 1.2).

$$Vdij = Vij - Vnij \tag{1.2}$$

Проведенням заходів управляють дві сторони - адміністратор СЕМ і адміністрація підприємства. Розглянемо їх цілі і дії, що управляють.

Інтереси СЕМ виходячи з її макрозавдання - запобігання збитку довкіллю (ОС) - в рамках управління проведенням заходів полягають в мінімізації сумарного збитку D протягом періоду T, заподіюваного джерелами ВОС і об'єктами проведення заходів.

Для формального опису критерію D введемо функцію миттєвого (у одиницю часу) збитку i-го джерела (формула 1.3).

$$g_i(t) = g_i^d \left(1 - U\left(t - t_i^f\right)\right), t \in [0, T] \tag{1.3}$$

де g_i^d - збиток в одиницю часу в і вартісному вираженні (формула 1.4);

$$g_i^d = \sum_i P_y^d V_y^d \tag{1.4}$$

pdij - норматив плати за збиток;

U(t) - одинична ступінчаста функція(формула 1.5).

$$U(t) = \begin{cases} 0, t < 0 \\ 1, t \geq 0 \end{cases} \tag{1.5}$$

Метою системи экомоніторингу є мінімізація критерію D (формула 1.6).

$$D \rightarrow min \tag{1.6}$$

Для досягнення своєї мети СЕМ використовує дію, що управляє (формула 1.7).

$$G_m = \{g_i^M(t), i = \overline{1,N}, t \in [0,T]\} \tag{1.7}$$

Направлене на адміністрацію підприємства у вигляді позову (экоплатежу) за збиток ОС. Адміністрація підприємства відає організацією і реалізацією всіх необхідних для виконання заходів інженерно-технічних робіт, а також питаннями фінансування. Проведення заходів фінансується підприємством за рахунок власних засобів. У зв'язку з цим метою адміністрації протягом періоду часу Т є мінімізація сумарних витрат Z в цей період, пов'язаних з проведенням заходів (останнє виходить з еколого-економічної ефективності заходів і, таким чином, наявності економічного ефекту від їх виконання). Для формального представлення критерію Z розглянемо структуру витрат, пов'язаних з проведенням заходів в динаміці. Введемо функції часу qi(t), ci(t), що позначають відповідно миттєві експлуатаційні і капітальні витрати на i-м джерелі і що володіють наступними властивостями.

Функція ci(t) є довизначена на інтервалі [0, T] функція освоєння капіталовкладень c0, (формула 1.8).

$$C_i(t) = \begin{cases} 0, t \in [0, t_i^f - \tau_i[U]t_i^f, T] \\ c_i^0(t), t \in [t_i^f - \tau_i t_i^f] \end{cases} \tag{1.8}$$

Функція qi(t) відмінна від нуля і постійна після проведення заходу (формула 1.9).

$$q_i^{(t)} = q_i^0 U(t - t_i^f), t \in [0, T] \quad (1.9)$$

Тоді критерій Z можна представити в наступному вигляді (формули 1.10).

$$Z = \int_0^T \sum_i Z_i(t)dt$$
$$Z_i(t) = c_i(t) + q_i(t) + g_i^M(t) \quad (1.10)$$

При плануванні фінансування робіт по введенню заходів адміністрація повинна враховувати динаміку сумарних витрат підприємства. Формальне представлення динаміки сумарних витрат в спільному випадку вимагає обліку багато чисельних економічних аспектів функціонування підприємства і є самостійною проблемою, вирішення якої виходить за рамки даної статті. У зв'язку з цим враховуватимемо вплив динаміки сумарних витрат на об'єктах проведення заходів в спрощеному вигляді, задавши цей вплив у вигляді обмеження (формула 1.11).

$$\sum_i Z_i(t) \le A(t) \quad (1.11)$$

де A(t) — задана функція часу (можлива інтерпретація A як різниці між сумарними максимально можливими миттєвими витратами, не пов'язаними з проведенням заходів).

Дія адміністрації, що управляє, є (формула 1.12).

$$T^M = \{t_i^t; i = \overline{1, N}, t_i^f \in [0, T]\} \quad (1.12)$$

Мінімізація параметра Z (формула 1.13).

$$Z \to \min \quad (1.13)$$

Таким чином, система управління проведенням заходів є ієрархічною, а керовані величини (витрати підприємства в процесі виконання заходів, збиток ОС в одиницю часу) - динамічними.

Визначення раціональних стратегій управління в даній системі (так само як і інтерпретація поняття раціональної стратегії) можна здійснити декількома способами. По-перше, можливе вживання методів імітаційного моделювання. Здобуття кінцевих

результатів для деякої конкретної системи в цьому випадку пов'язане з необхідністю постановки і проведення імітаційних експериментів на ЕОМ, що практично не реалізовується в рамках існуючого статусу і наявних можливостей СЕМ. По-друге, завдання після ряду спрощень, заснованих на деяких припущеннях (як правило, виконуваних на практиці або виправданих з практичної точки зору) можна звести до завдання визначення раціональних стратегій в стратегічній ієрархічній грі двох осіб. Застосуємо останній підхід.

Передбачимо, що позов (управління СЕМ) пропорційний збитку ОС джерела, а константа пропорційності не залежить від часу (формула 1.14).

$$g_i^M = S_i g_i(t), S_i \in [0,1] \tag{1.14}$$

Дане припущення означає, що управління СЕМ тепер є (формула 1.15).

$$S^M = \{S_i; i = \overline{1,N}, S_i \in [0,T]\} \tag{1.15}$$

Замінимо обмеження (1.12) еквівалентною системою не залежних від t обмежень. Для здобуття такої системи використовуємо циклічність з кроком I оцінки економічних показників підприємства (на практиці типова квартальна циклічність). Тоді в умовах циклічності замість функції A необхідно використовувати вектор Ak (формула 1.16).

$$k = \overline{0,\frac{T}{h}} \tag{1.16}$$

З урахуванням сказаного система обмежень набуває вигляду (формула 1.17).

$$Z_k(T^M, S^M) \leq A, k = \overline{0,\frac{T}{h}}, \tag{1.17}$$

Використовуючи усереднювання капіталовкладень на інтервалі (формула 1.18)

$$t \in [0, \tau_i] \tag{1.18}$$

Визначимо функцію миттєвих кап затрат (формула 1.19).

$$c_i(t) = c_i^{\bar{0}}\left(U(t - t_i^f + \tau) - U(t - t_i^f)\right),$$
$$c^{\bar{0}} = K_{\frac{i}{\tau_i}} \qquad (1.19)$$

Тепер всі ступінчасті функції з (1.8,1.9) виражені через U. Тоді, використовуючи відповідну апроксимацію для U, можна отримати все Zk(Tu, Su), у явному вигляді.

Таким чином, отримана модель системи управління проведенням природоохоронних заходів у вигляді статичної дворівневої ієрархічної системи, цілі управління і дії, що управляють.

Оптимальні стратегії управління в розглянутій системі можуть бути отримані на основі використання теорії статичних ігор з ієрархічною структурою.

1.11. Методи розрахунку екологічних показників

Зазвичай в місті протягом одного дня відбирається 50 - 100 проб повітря на різні складові домішок. Для оцінки міри забруднення атмосфери міста в цілому використовуються різні узагальнені показники. Одним з найбільш простих показників забруднення служить нормована (безрозмірна) концентрація домішок (q), усереднена по всьому місту і по всіх термінах спостережень (формула 1.20).

$$\bar{q} = \frac{1}{N}\sum_{i=1}^{N}\frac{q_i}{\overline{q_i}} \qquad (1.20)$$

де qi - середня за день концентрація в і - тому пункті;

qi - середньосезонна концентрація в тому ж пункті;

N - число стаціонарних пунктів в місті.

Нормування на середньосезонну концентрацію дозволяє виключити вплив зміни спільній концентрації від року до року, що дає можливість використовувати для аналізу ряд спостережень за декілька років.

Як інший показник забруднення використовується коефіцієнт при першому членові розкладання концентрації домішки по природних ортогональних функціях (е.о.ф., формула 1.21).

$$a_1 = \sum_{i=1}^{N} \varphi_{L_1} * q_i \qquad (1.21)$$

де φ_{Li} - компоненти першої е.о.ф.;

qi - відхилення концентрації від її середнього значення в і - тому пункті.

Коефіцієнт (Хі характеризує одночасне зростання або зменшення концентрації домішки по всьому місту (цей показник можна використовувати лише в тих пунктах спостереження, в яких розрахунок виконується за допомогою ЕОМ).

Найбільшого поширення набув третій показник фонового забруднення міста параметр (формула 1.22).

$$\rho = \frac{m}{n} \qquad (1.22)$$

де n - спільне число спостереженні за концентрацією домішок в місті в течії одного дня на всіх стаціонарних пунктах, m-число спостережень _ за концентрацією, що перевищує середньосезонне значення більш ніж в 1,5 разу (qi>l,5 qj).

Параметр р тісно пов'язаний з коефіцієнтом сij, коефіцієнти кореляції г між ними складе 0,85 - 0,94, і нормованим середнім значенням q (г = 0,65 - 0,90). Середньосезонні значення qj в і - тому пункті визначаються для кожного року по середньомісячних значеннях, наприклад, для літа (формула 1.23).

$$q_i = \frac{q_{iv1} + q_{iv11} + q_{iv111}}{3} \qquad (1.23)$$

При оперативному розрахунку р слід орієнтуватися на середні значення qi за відповідний сезон попереднього року і попередній місяць даного року.

Для багатьох міст параметр р можна розраховувати по сукупності ряду домішок. Слід лише виключити ті специфічні домішки, які поступають в атмосферу з чітко виражених джерел. Існує досить тісна кореляційна зв'язок між параметром р , розрахованим для всіх домішок і середньодобових концентрацій q окремих домішок (СО, SO2, NO2). Коефіцієнти кореляції між р и q складають в Ленінграді і Читі 0,76 и 0.82 для S02, 0,70 , 0,84 для N02, 0,58 та 0,69 для СО відповідно.

Виділяють три рівні забруднення повітря в місті: високий (1 група) - р >0,35; підвищений (11 група) - 0,20 < р < 0,35 та знижений (111 - група) - р < 0,20.

Високий рівень забруднення (р > 0,35) формується за наступних метереологічних умов:

а) вночі або вранці даного дня дуже слабкий (до 1 м/с) вітер або безвітря. А напередодні спостерігалося підвищене значення р: р' > 0,3;

б) вдень (за даними спостереженнями о 15 годині) безвітря або дуже слабкий вітер, а напередодні р' > 0,15;

в) відносно висока температура повітря при слабкому (до 5 м/ с) вітрі в уранішні години даного дня, а напередодні р' > 0,30;

г) помірний (3-6 м/с) вітер і нестійка стратифікація вдень змінялися безвітрям увечері, а напередодні р' > 0,15;

д) напередодні р' > 0,4 , а в подальший день не очікується посилення вітру або випадання значних опадів;

е) при дуже слабкому (до 1 м/с) вітрі спостерігається туман або підведена інверсія.

Правдивість кожної з цих ознак (предикторів) складає 60 - 70 %. Знижений рівень забруднення наголошується при наступних метеорологічних умовах:

а) швидкість вітру (по флюгеру) перевищує 5-6 м/с;

б) помірний або сильний дощ;

в) напередодні в другій половині дня р' >0,15. Правдивість в цьому випадку складає 70 - 90 %.

Високий рівень забруднення спостерігається в центральних частинах стаціонарних антициклонів, в мало градієнтних баричних полях при адвекції тепла в тропосфері, в західних і північно-західних частинах малорухливих антициклонів, в теплих секторах циклонів при малих значеннях градієнта тиску. Знижене забруднення спостерігається при активній циклонній діяльності, при великих значеннях градієнта тиску.

Розпізнавання образів. Спосіб розпізнавання образів зводиться до визначення міри близькості конкретної обстановки до вказаних вище рівнів забруднення. З цією метою для кожного рівня забруднення визначаються середнє значення х і середнє квадратичне відхилення ох наступних предикторів: швидкості вітру на висоті 10 м-коду (U10) і 500 м-кодів (U500), різниці температур (ΔT = Т0 – Т500) повітря в земної поверхні Те на висоті 2 м-коду і параметра р' в попередній день. Потім розраховуються величини (формула 1.24).

$$\rho' = \sum_{k=1}^{5} \left[\frac{(x_k - \overline{x_{kj}})}{\sigma_{xkj}}\right]^2 \qquad (1.24)$$

Близькість даної конкретної метеообстановки до середнього достатку j -того рівня забруднення, j - 1, 11, 111. Тут $X_1 = u_{10}$, X2 = U500, X3 = ΔT, X4 = To, xs = p'; Xkj - середні значення величини Хк при j - тому рівні забруднення; σ_{xkj} — середньоквадратичне відхилення величини Хк при тому ж j - тому рівні забруднення.

Наприклад, середні значення і середньоквадратичні відхилення предикторів, отримані за даними спостережень взимку в Красноярську, представлені в таблиці 1.1.

У згоді з теорією, забруднення повітря тим значніше, чим менше швидкість вітру в земної поверхні і на висоті 500 м-коду, більше (по модулю) різниця ΔT, хоча зміни її украй малі, нижче приземна температура повітря То і вище рівень забруднення в попередній день p' (таблиця 1.1).

Таблиця 1.1 - Середні значення і середньоквадратичні відхилення метеопоказників

Рівень забруднення	Середні значення					Середньоквадратичні відхилення				
	U_{10} м/с	U_{500} м/с	ΔT °С	T_0 °С	p'	U_{10} м/с	U_{500} м/с	ΔT °С	T_0 °С	p'
1	1,44	6,1	-1,6	-25	0,31	2,4	3,4	4,0	8,2	0,10
11	1,85	8,7	-1,3	-20	0,17	2,3	4,6	4,3	9,7	0,10
111	6,25	13,3	-1,4	-15	0,15	5,2	4,4	3,5	11,9	0,08

На підставі фактичних даних про предикторах Хк по співвідношенню (1.26) розраховуємо $\rho^2 1, \rho^2 11, \rho^2 111$. Дану конкретну обстановку відносимо до того рівня, для якого величина p^2_j виявиться найменшою. Так в один з днів спостерігалося: U_{10}=0, U_{500} = 5 м/с, ΔT = -3°С, T_0 = - 22 °С, p' = 0,27. При тих значеннях X_{kj} и σ_{xkj}, які відображують в таблиці 1.1, в цьому випадку маємо $\rho^2 1$ = 0,89, $\rho^2 11$ = 2,87, $\rho^2 111$ = 7,84. Таким чином даний випадок можна віднести до високого рівня забруднення, оскільки p^2 1— найменше. Правдивість прогнозів, складених по методу розпізнавання образів, складає 73 - 75 %.

Засоби послідовної графічної регресії. Метод послідовної графічної регресії включає побудова кореляційних графіків по наявному ряду спостережень для визначення параметра p' по різних поєднаннях величин (предикторів), які роблять істотний вплив на рівні забруднення. У найбільш поширеному варіанті схеми прогнозу використовуються наступні пари предикторів: U10, ΔT , U500 и p'. На рисунку 1.4 наведений приклад побудови кореляційних графіків. На рисунку 1.5 а по осях координат відкладені швидкість вітру U10 на висоті 10 м-коду, тобто на рівні флюгера, і різниця ΔT температур повітря в земної поверхні T0 і на висоті 500 м T500, на рисунку 1.4 б - швидкість вітру U500 і значення (p') параметра p в попередній день. Будуються подібні графіки для кожного міста з використанням всіх наявних матеріалів спостережень за декілька років. З цією метою по виміряних значенняхΔT и U10 наносяться на графік (рисунку 1.4 а) крапки, біля яких вписується визначене за той же термін, що і дляΔT , U10, значення параметра p .

Після того, як на графік нанесена чимала кількість крапок, що охоплюють всі рівні забруднення, проводяться ізолінії параметра p (кратні 0,1) так, щоб між ізолініями, що проводяться p знаходилося найбільше число крапок з відповідними відмітками. Аналогічно будується графік по значеннях U500 и p'. Графіки ці будуються окремо не лише для кожного міста, але і за наявності великого ряду спостережень для кожного сезону року.

Можна відзначити деякі особливості графіків, подібних представленому на рисунку 1.4. Високі і підвищені рівні забруднення (p > 0,3 - 0,4) наголошуються в наступних випадках:

а) при приземному вітрі, близькому до штилю, і інверсійному достатку нижньої частини пограничного шару;

б) при помірній (2-3 м/с) швидкості вітру і при значному падінні температури повітря з висотою (ΔT > 3°C/100 м).

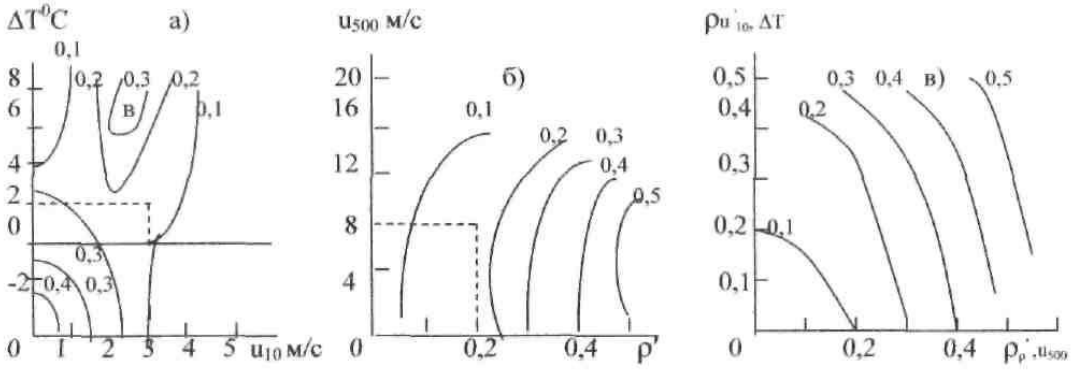


Рисунок 1.4 Графіки для передбачення фонового забруднення в місті.

а - залежність параметра р від різниці температур в шарі 0 - 500 м (ΔT) і від швидкості вітру на рівні флюгера (U10); б - залежність параметра р від швидкості вітру на рівні 500 м (U500) і значення р у попередній день (р'); в - прогностичні значення р, визначувані по чотирьом предикторам.

Наявність цього максимуму на рисунку 1.4 а) пояснюється забрудненням приземного шару повітря домішками, висотними джерелами, що викидаються (заввишки більш 100 - 200 м): при слабкому вітрі і інверсійній стратифікації (ослабленому турбулентному обміні) ці домішки зберігаються на висоті; при сильному вітрі турбулентний обмін розносить домішка на весь пограничний шар; і лише при помірних швидкостях вітру і відсутності інверсії температури викиди від висотних джерел, поширюючись вниз і вгору, створюють підвищені рівні забруднення повітря поблизу земної поверхні;

У високих рівнях забруднення в попередню добу, а також за ослабленої швидкості вітру на висоті 500 м. Після того, як графіки, подібні змальованим на рисунку 1.4, за результатами спостережень за декілька років побудовані, їх використовують для прогнозу рівнів забруднення. На відміну від діагнозу, при прогнозі як вхідні параметри ΔT , U10, U500 притягуються їх прогностичні значення, передбачені по якій-небудь з прийнятих методик (синоптичною, чисельною). Параметр р визначається за даними спостережень за поточний день і використовується як вхідний параметр на рисунку 1.4 б) при прогнозі рівня забруднення наступного дня. Остаточна думка про рівень забруднення р складається по рисунку 1.4 в), на якому по осях відкладені значення параметра р, визначені по рисунку 1.4 а) ($\rho_{U10}, \Delta T,$), а також рисунку 1.4 б) ($\rho_{\rho'}, \mathbf{U_{500}}$)•

1.12. Інформаційна модель системи екологічного моніторингу

При розробці проекту екологічного моніторингу необхідна наступна інформація:

а) джерела вступу забруднюючих речовин в природне довкілля - викиди забруднюючих речовин в атмосферу промисловими, енергетичними, транспортними та іншими що призводять до викиду в атмосферу небезпечних речовин і розливу рідких забруднюючих і небезпечних речовин і т.д.;

б) перенесення забруднюючих речовин - процеси атмосферного перенесення;процеси перенесення і міграції у водній середі;

в) процеси ландшафтно-геохімічного перерозподіли забруднюючих речовин - міграція забруднюючих речовин за ґрунтовим профілем до рівня ґрунтових вод; міграція забруднюючих речовин по ландшафто-геохімічному сполученню з врахуванням геохімічних бар'єрів і біохімічних круговоротів; біохімічний круговорот і т.д.;

г) дані про достаток антропогенних джерел забруднення - потужність джерела забруднення і місце розташування його, гідродинамічні умови вступу забруднення в довкілля.

Характер і механізм узагальнення інформації про екологічну обстановку при її русі по ієрархічних рівнях системи екологічного моніторингу визначаються за допомогою поняття інформаційного портрета екологічної обстановки. Останній є сукупністю графічно представлених просторово розподілених даних, що характеризують екологічну обстановку на певній території, спільно картооснови місцевості. На рисунку 1.5 відображено схему інформаційних потоків системи екологічного моніторингу.

Рисунок 1.5 – Схема інформаційного потоку системи екологічного моніторингу

1.13. Передача даних у системі екологічного моніторингу

Для вибору найбільш оптимального способу передачі даних системи екологічного моніторингу з точки зору таких показників як: якість, ціна, надійність треба проаналізувати наступне:

а) Типи можливих каналів зв’язку:

1) Фізичні пари (не ущільнена лінія зв'язку).

2) Мідні кабельні канали, симетричні кабелі, коаксіальні кабелі, вживані на них системи частотного і тимчасового ущільнення.

3) Супутникові канали, їх достоїнства і недоліки.

4) Радіорелейні канали.

5) Оптоволоконні канали, принципи передачі сигналів за допомогою світлового променя, характеристики оптичних волокон, типи оптичних волокон, перешкоди, характерні для оптичних ліній.

б) Принципи передачі цифрової інформації по аналоговому каналу:

1) Необхідність узгодження цифрового потоку з аналоговим каналом, призначення модемів.

2) Асинхронна цифрова передача, її особливості.

3) Синхронна цифрова передача, її особливості, формат синхронного потоку даних.

4) Необхідність системи синхронізації при синхронній передачі, помилки, пов'язані з порушенням синхронізації.

5) Передача цифрового потоку по аналоговому телефонному каналу, поняття фазової і амплітудної модуляції, амплітудні і фазові спотворення, швидкість модуляції.

6) Взаємодія комп'ютера з модемом, ООД і АКД.

в) Комутація цифрових каналів:

1) Цифрова і аналогова комутації, переваги цифрової комутації.

2) Тимчасова комутація.

3) Просторова комутація.

4) Побудова комбінованих просторово-часових комутаторів.

5) Різні варіанти мережевих топологій: зіркоподібна, ієрархічна, кільцева, порівняльна оцінка топологій.

г) Основні методи передачі даних:

1) Комутація каналів.

2) Комутація повідомлень.

3) Комутація пакетів.

д) Локальні обчислювальні мережі:

1) Характерні особливості локальних мереж в цілому.

2) Типи кабелів для побудови локальних мереж: коаксіальний кабель, витаючи пара, оптоволокно.

3) Принципи функціонування і особливості мереж Ethernet, поняття сегменту ЛВС, допустимі параметри сегментів, об'єднання різних сегментів.

4) Принципи функціонування і особливості мереж Token Ring.

5) Порівняльні характеристики різних типів ДВС.

6) Організація взаємодії між різними ділянками локальних мереж, рипітеры, мости, маршрутизатори.

7) Підключення локальної мережі до магістральної мережі.

е) Технологія роботи і додатка мережі Інтернет:

1) Особливості роботи мережі Інтернет на базі технології комутації, датаграмна служба, її достоїнства і недоліки.

2) Спільний опис технології TCP/IP: функції протоколу, технологія роботи мережі, елементи системи маршрутизації, стандарти в Інтернет.

3) Спільні наслідки з принципів роботи Інтернет: проблеми, пов'язані з продуктивністю комутаторів, нерівномірність затримок при передачі, скрута при наданні голосових і відеопослуг через мережу Інтернет.

4) Адресація в мережі Інтернет, структура адреси, типи адрес, буквені імена, DNS.

5) Спільна технологія входження в співтовариство Інтернет: виділення адресного простору, організація каналу зв'язку до провайдера, побудова власної топології в рамках адресного простору.

6) Основні застосування на мережі Інтернет: WWW, FTP, Gopher, IRC, електронна пошта, телеконференції.

7) Поняття Інтернет, призначення і принципи побудови, складові частини.

ж) Мережі X.25:

1) Структура протоколів мереж X.25.

2) Спільний опис технології X.25: сигналізація, захист від помилок, адресація.

3) Поняття віртуального з'єднання, опис служби віртуальних з'єднань, логічні канали, багатоканальна процедура.

4) Основні послуги мереж X.25: замкнуті групи абонентів, швидкий вибір, постійні віртуальні канали, клас пропускної спроможності, ідентифікатор користувача.

5) Методи асинхронного доступу в мережу X.25, Пади, їх призначення.

6) Взаєморозрахунки в мережах X.25, принципи оплати послуг мереж X.25.

7) Огляд мереж X.25: РОСПАК, IACHET, ІНФОТЕЛ, РОСПРІНТ, РОСНЕТ.

з) Технологія Frame Relay:

1) Спільне уявлення про технологію Frame Relay.

2) Корінні відзнаки технології Frame Relay від технологій X.25 і TCP/IP.

3) Технологія функціонування системи Frame Relay.

4) Принципи організації призначеного для користувача доступу до мережі Frame Relay, параметри якості в Frame Relay, визначення параметрів якості при підключенні абонента.

5) Місце технології Frame Relay на ринку послуг.

и) Мережі ISDN:

1) Принципові відзнаки мережі ISDN від звичайної телефонної мережі.

2) Нові можливості мережі ISDN в порівнянні з телефонною мережею.

3) Спільна структура ISDN.

4) Послуги, що надаються мережею ISDN.

5) Структура абонентського стику ISDN, його складові частини, апаратні засоби.

6) Типи каналів, що надаються абонентові на вході в мережу.

7) Організація системи передачі даних через мережу ISDN, підсистема комутації пакетів, пристрої, необхідні для підключення терміналів до підсистеми комутації пакетів, способи організації такої підсистеми.

8) Опис процедури встановлення з'єднань через мережу ISDN.

к) Технологія ATM:

1) Потреби операторів зв'язку в створенні єдиної інтегральної мережі.

2) Прогрес технологій комутації.

3) Вимоги до систем зв'язку для передачі цифрової інформації.

4) Основні призначені для користувача характеристики систем передачі даних: затримка передачі, достовірність переданої інформації.

5) Концептуальні основи технології ATM: відсутність теоретичної межі в нарощуванні швидкості роботи мережі, відсутність систем захисту від помилок, система синхронізації, структура пакетів ATM.

6) Основні достоїнства і недоліки технології ATM: єдина мережа, гарантована якість послуг для різних видів сервісу, вимоги, що пред'являються мережею до абонента, неповноцінне статистичне ущільнення.

7) Віртуальні з'єднання в ATM: віртуальні канали, віртуальні дороги.

8) Контроль якості в мережі ATM: параметри встановлюваного з'єднання, параметри якості сервісу.

9) Типи рівнів адаптації АТМ, їх призначення, класи якості сервісу для різних рівнів адаптації.

10) Режим LAN Emulation і MPOA - додаткові послуги мережі АТМ.

11) Основи абонентської сигналізації в АТМ, встановлення з'єднань, типів з'єднань.

12) Технології підключення до опорної мережі АТМ локальних мереж, окремих робочих станцій.

1.14. **Технічне завдання на розробку**

На рисунку 1.6 відображено блок - схема системи міського екологічного моніторингу.

а) Гідрометслужба.

Щомісячно надає інформацію про:

1) Кількість стихійних звалищ (шт.).

2) Кількість вивезеного сміття (кг).

3) Сума коштів, яка виділялась для вивозу сміття зі стихійних звалищ (грн.) .

Два рази на рік надає інформацію про:

1) Кількість висаджених дерев і чагарників (шт.).

2) Суму коштів, яка витрачена на посадку дерев і чагарників (грн.).

3) Кількість суботників (шт.).

4) Кількість присутніх (працюючих) на суботниках (осіб).

б) Схід ДРГП (Східне державне регіональне геологічне підприємство).

Надає інформацію про результати хімічного аналізу ґрунтів на вміст фенолів та інших хімічних речовин. Періодичність надання інформації визначається календарним планом договору.

в) Лісомисливське господарство.

Щоквартал надає інформацію про:

1) Кількість висаджених дерев (шт.).

2) Суму коштів, яка виділялась на посадку (грн.).

3) Кількість пожеж (шт.).

4) Суму збитків від пожеж (грн.).

5) Кількість людей, що прийняли участь у посадці дерев (осіб).

г) Комплексна інспекція водних ресурсів.

Два рази на рік проводить спільні перевірки і надає інформацію про:

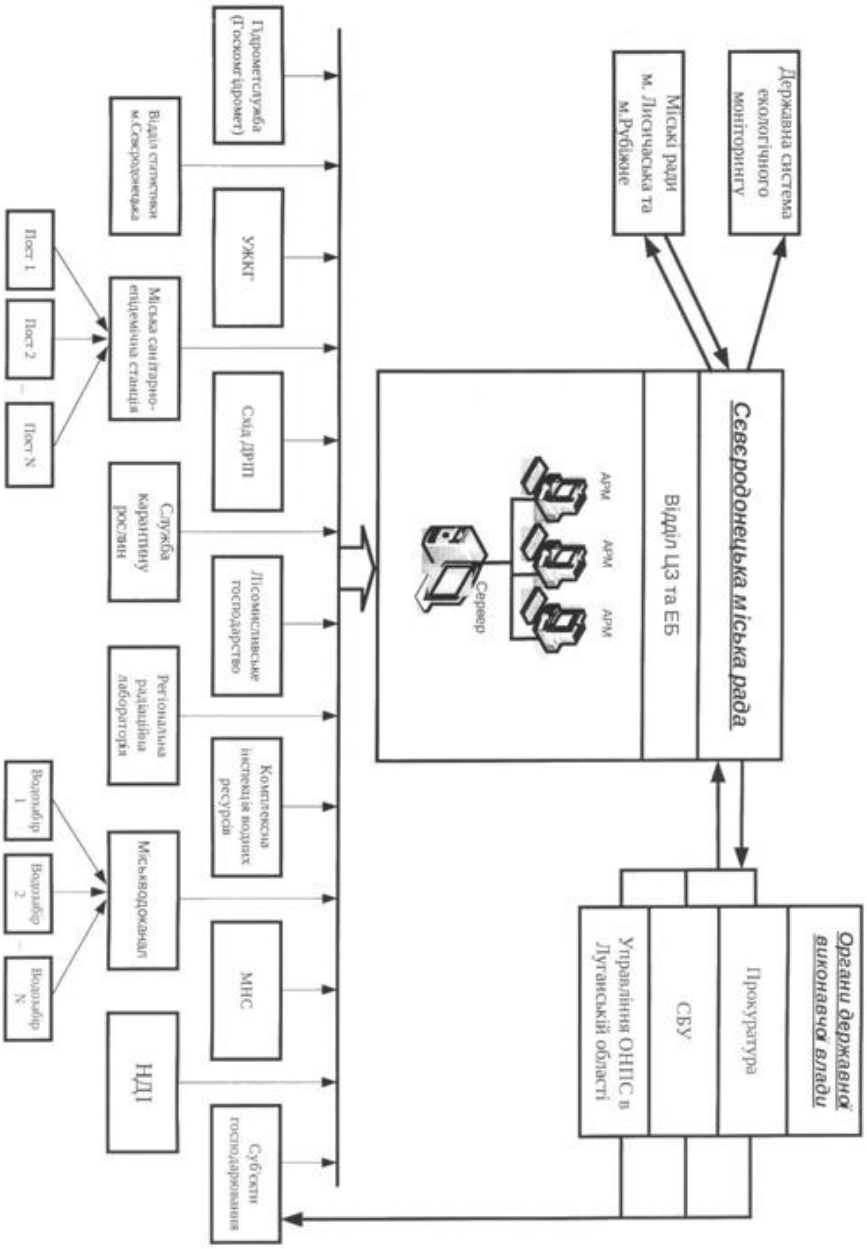


Рисунок 1.6 - Блок - схема системи екологічного моніторингу

1) Кількість виявлених порушень Водного Кодексу (шт.).
2) Кількість накладених штрафів (шт.).
3) Суму штрафів (грн.).
4) Загальні дані щодо забруднення поверхневих вод.

д) МНС (Міністерство надзвичайних ситуацій).

Щомісячно надає інформацію про:

1) Кількість надзвичайних ситуацій природного та техногенного характеру (шт.).

2) Кількість постраждалих (осіб).

3) Кількість загиблих (осіб).

4) Суму нанесених збитків (грн.).

е) Суб'єкти господарювання.

У разі потреби надають інформацію про:

1) Ліміти на утворення відходів.

2) Кількість викинутих у повітря забруднюючих речовин (т/рік).

3) Суму збору за забруднення довкілля (грн.).

4) Кількість джерел викидів (шт.).

5) Кількість одиниць пересувних джерел (автомобілів) (шт.).

6) Об'єми скиду ($м^3$/рік).

7) Кількість забруднюючих речовин у скиді (т/рік).

ж) Відділ статистики М.Сєвєродонецька.

з) Міська санітарно-епідемічна станція.

1) Щокварталу надає інформацію про:

2) Стан атмосферного повітря.

3) Стан водойомів (ріки Сєвєрський Донець та ріки Борова).

и) Служба карантину рослин.

Щокварталу надає інформацію про:

1) Рослини, на які введено карантин в Луганській області.

2) Рослини, на які введено карантин в М.Сєвєродонецьку.

3) Кількість виявлених випадків виявлення карантинних рослин (шт.).

4) Площу заражених територій міста ($м^3$).

5) Кількість накладених штрафів (шт.).

6) Суму штрафів (грн.).

к) Регіональна радіаційна лабораторія.

Щокварталу надає інформацію про:

1) Кількість відібраних проб (шт.).

2) Кількість проб з перевищенням допустимого рівня потужності поглиненої дози гама-випромінювання (мкр/) (шт.).

3) Кількість проб з перевищенням еквівалентної рівноважної об'ємної активності радону (бк/м^3).

л) Міськводоканал.

Один раз на рік надає інформацію про:

1) Кількість водозаборів.

2) Потужність водозаборів.

3) Якість води.

Призначення та галузь застосування.У дипломному проекті потрібно дослідити і розробити програмне забезпечення сервера регіональної системи екологічного моніторингу м. Сєвєродонецьк.

Призначення програмного забезпечення сервера регіональної системи екологічного моніторингу:

а) збір оперативної інформації про стан екології в місті і відображення її в зручному вигляді;

б) підтримка статистичної, математичної, спеціалізованої обробки, а також підсистеми ухвалення рішень.

Мета розробки даної системи:

а) створення розрахованого на багато користувачів програмного забезпечення, що здійснює збір інформації, обробку, що дозволяє отримати об'єктивну оцінку про екологічний стан міста і регіону в цілому у будь-який час;

б) аналіз моніторингової інформації і генерація звітів.

У вимоги замовника до системи входять пункти, що свідчать про те, що система працюватиме в розрахованій на багато користувачів середі, орієнтованій на роботу в публічній мережі Інтернет. Система, повинна функціонувати з чітким розмежуванням прав читання, доповнення, коректування і видалення елементів бази, а також заповнення і коректування окремих реквізитів елементів, різними користувачами або групами користувачів; і про необхідність забезпечення захисту від несанкціонованого і одночасного доступу до даних в базі даних.

Програмне забезпечення сервера регіональної системи екологічного моніторингу, що розробляється, повинне попередити виключення ситуацій одночасного доступу до даних (зміна) засобами шпигунський програм і вірусів.

Вимоги до функціональних характеристик. Підсистема синхронізації повинна відповідати наступним вимогам:

а) забезпечення безпечних методів відправки і здобуття моніторингової інформації ручним введенням і автоматизованим;

б) незалежність від числа постачальників моніторинговій інформації і вигляду її представлення, а також від часу оновлення;

в) розумні вимоги до ресурсів, що дозволяють всій системі функціонувати в реальному режимі часу навіть при інтенсивному потоці заявок до підсистеми;

г) забезпечення необхідної частоти перевірки на наявність оновлень моніторингової інформації з аналізом результату і генерацією звіту;

д) незалежність від СУБД і структури бази даних.

Вимоги до надійності.Основним показником надійності програмного продукту є середнє число прогонів програми до настання помилки. Воно має бути не нижче чим 1000 разів. Підтвердження показників надійності має бути забезпечене на стадії стабілізації програмного забезпечення сервера.

Постановка завдання до магістерської роботи

Програмне забезпечення сервера регіональної системи екологічного моніторингу повинне відповідати наступним вимогам:

а) стежити, щоб моніторингова інформація була коректно і вчасно отримана від заданого постачальника інформації;

б) стежити за правами доступу до бази даних користувачів сервера;

в) обробляти і аналізувати наявну інформацію в базі даних і вчасно повідомляти про виникнення виняткових ситуацій;

г) блокувати небезпечних користувачів, шпигунські програми, віруси, що намагаються дістати доступ до бази даних;

д) перевіряти аутентифікацію кожного користувача і чітко розмежовувати його права доступу в системі;

е) перевіряти сервери на аварійні завершення;

ж) забезпечивати адміністративний режим;

з) підтримувати автоматичну підсистеми ухвалення рішень на основі наявної моніторингової інформації з генерацією звіту, що містить прогноз і результати обробки, аналізу.

Результатом роботи повинен стати готовий до використання працездатний тестовий сервер регіональної системи екологічного моніторингу, що забезпечує реалізацію вимог завдання.

РОЗДІЛ 2. ВИБІР ВАРІАНТІВ ВИКОНАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СЕРВЕРУ СИСТЕМИ РЕГІОНАЛЬНОГО ЕКОЛОГІЧНОГО МОНІТОРИНГУ ДЛЯ ПРОГНОЗУВАННЯ ВИТРАТ ВІД ЗАБРУДНЕННЯ АТМОСФЕРНОГО ПОВІТРЯ

2.1 Варіанти рішення поставленої задачі

Існує декілька варіантів створення такого роду програмного забезпечення:

а) Розробка програмного забезпечення на основі технологій ASP.NET, IIS, C#,xml,sql2000,ajax,css,uml.

Один з найбільш відповідних варіантів рішення поставленої задачі. Проте вимагає довгого часу розробки оскільки дана “зв'язка” технологій розрахована на великі довгі проекти. Виходячи з поставленого завдання можливо реалізація останньою в перспективі;

б) Розробка програмного забезпечення на основі технологій RUBY,SQL,HTML,CSS,JAVAScript,UML.

Альтернативний спосіб рішення поставленої задачі проте виникають трудності із за непоширеності мови RUBY і повного відсутності навчальних посібників російською мовою. Одним з недоліків який зробить істотний вплив -продуктивність. Що при реалізації великого проекту буде помітніше.

в) Розробка програмного забезпечення на основі технологій Python, SQL,HTML,CSS, JAVAScript,UML. аналогічний спосіб рішення поставленої задачі описаний вище з тими ж труднощами і недоліками.

г) Розробка програмного забезпечення на основі технологій PERL,XML,HTML,MYSQL,AHACHE,AJAX, JAVAScript,CSS,UML.

Один з найбільш відповідних варіантів рішення поставленої задачі, який може бути узятий, враховуючи переваги вхідних в ”зв'язці” технологій і мови Perl. Проте не є найбільш відповідним варіантом рішення поставленої задачі, із за обов'язкової наявності інтерпретатора мови PERL.

д) Розробка програмного забезпечення на основі технологій PHP,XML,HTML,MYSQL,AHACHE,AJAX, JAVAScript,CSS,UML.

Найбільш прийнятний варіант рішення поставленої задачі виходячи з вимог проекту, а також достоїнств і недоліків вхідних в реалізацію технологій. Мова PHP найбільш кращим чином личить для реалізації відносно невеликих, швидких по продуктивності проектів в коротких проміжок часу. Проте має недолік пов'язаний із

зайвою простотою реалізації типів і класів, що при побудові великих проектів виявить трудності в більшому написанні коду, чим з ”зв'язкою” запропонованою в пункті 1.

2.2 Автоматична система екологічного моніторингу

Автоматичну систему екологічного моніторингу, можна представити як мережевий комплекс, об'єднуючий вимірювальні пристрої і контролери пунктів моніторингу, робочі станції центру моніторингу між собою, а також з рівнем управління регіоном .

На рисунку 2.1 відображено структурна схема автоматичної системи екологічного моніторингу.

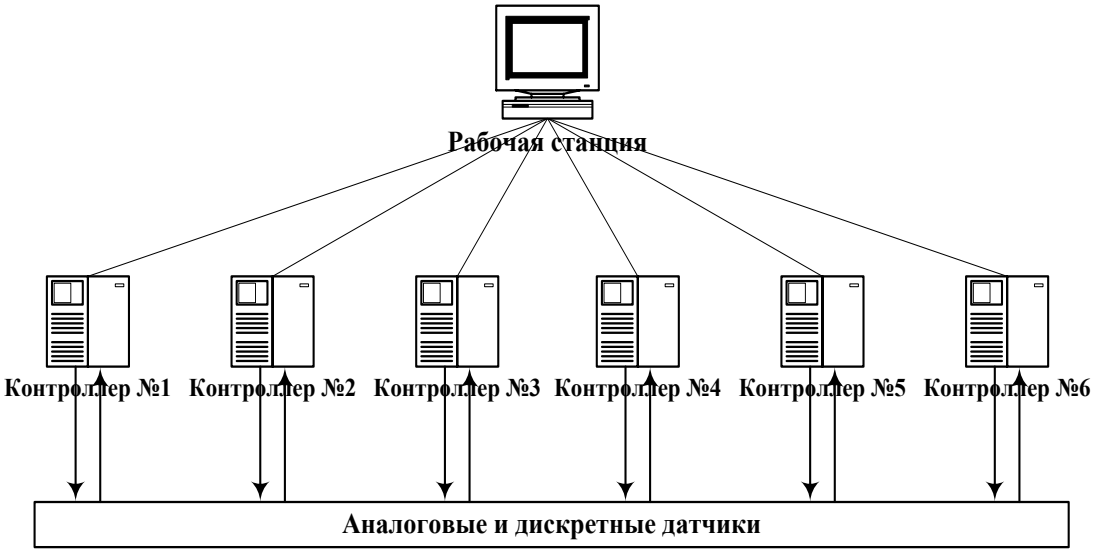


Рисунок 2.1 –Автоматична система екологічного моніторингу

Робочі станції займаються представленням екологічної інформації, її архівацією і аналізом. На рисунку 2.2 відображено рівні керування системи екологічного моніторингу.

Рисунок 2.2 - Рівні керування системи екологічного моніторингу

РОЗДІЛ 3. ДОСЛІДЖЕННЯ МЕТОДІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ВИТРАТ ВІД ЗАБРУДНЕННЯ АТМОСФЕРНОГО ПОВІТРЯ ПІДПРИЄМСТВАМИ ПРОМИСЛОВОГО РЕГІОНУ

3.1 Дослідження методів розробки програмного забезпечення серверу регіональної системи екологічного моніторингу

3.1.1 PHP

Історія PHP починається з 1995 року, коли Расмус Лердорф (Rasmus Lerdorf) створив просте застосування на мові Perl, що аналізує відвідин користувачами його резюме на веб-сайті. Потім, коли цим застосуванням вже користувалися декілька чоловік, а число тих, що бажають отримати його постійно збільшувалося, Лердорф назвав своє творіння Personal Home Page Tools версія 1 і виставив для вільного скачування. З цієї миті почався небувалий зліт популярності PHP.

Як це завжди буває, терміново було потрібно доопрацювання і доповнення. Для їх реалізації Расмус створює нову версію пакету, тепер уже написану на C. Отриманий таким чином інструменту набуває робоча назва PHP/FI (Personal Home Page / Forms Interpreter - Персональна Домашня сторінка/Інтерпретатор Форм), надалі він також буде відомий під назвою PHP 2. Ця версія вже більшою мірою схожа на сьогоднішній PHP. Вона мала синтаксис і спосіб іменування змінних в стилі мови Perl, автоматичну інтерпретацію форм, інтеграцію з базами даних (в основному з mSQL) і можливість вбудовування PHP операторів в html-код сторінки. При цьому все працювало дуже швидко, оскільки PHP компілювався до веб-сервера-сервера Apache. До 1997 року PHP використовувався вже на 50,000 доменах (не більше 1% всіх веб-серверов).

У тому ж 1997 року до проекту PHP підключилися Зів Сураськи (Zeev Suraski) і Енді Гутманс (Andi Gutmans). Будучи студентами одного з Ізраїльських університетів, вони намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи вихідний код PHP 2, Зів і Енді прийшли до виводу про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягом декількох місяців вони блискуче впоралися з цим завданням, тим більше приємно, що дана робота була зарахована ним як учбове навантаження університету.

Закінчивши роботу Зів і Енді зв'язалися з Расмусом, який сприйняв всі зміни PHP "на ура". З цієї миті з'являється PHP Group - група однодумців, що працюють над розвитком технології PHP. Отриманий продукт спільної діяльності побачило світло в 1998 році під назвою PHP 3.

При цьому головною особливість PHP 3 була можливість розширення ядра, що залучило до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість працювати з величезним кількість баз даних, протоколів, підтримувати велике число API. До кінця 1998 кількість користувачів PHP перевалила за 100000, а PHP був вже встановлений на не менше чим 10% серверах Інтернету. У пресі було опубліковано більше 20 статей по темі PHP і вийшли 2 книги.

Відразу ж після виходу PHP 3, Енді Гутманс і Зів Сураськи почали переробку ядра PHP. В першу чергу належало вирішити проблему підвищення продуктивності. Новий движок, названий Zend Engine (від імен творців: Zeev і Andi), успішно справлявся з цим завданням і був реалізований в 1999 році. Основний ідей його використання була можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.

PHP 4, що працює на цьому движку вийшов в 2000 році. На додаток до поліпшення продуктивності, PHP 4 мав нові можливості по підтримці сесій, буферизацію виводу, безпечні способи обробки інформації, що вводиться користувачем, і нові мовні конструкції. З виходом 4 версії PHP почав використовуватися вже на більш ніж 20% доменів Інтернету.

За час з 2000 по 2004 рік продовжувалися активні роботи по поліпшенню 4 версії, але майже відразу PHP Group приступила до продумування можливостей нової версії. В першу чергу було вирішено підсилити об'єктні можливості мови, що дозволяло використовувати його для реалізації досить крупних проектів. Роботи із створення версії 5 велися тривалий час, в них брала участь рекордна кількість фахівців, з яких хотілося б особливо відзначити Стерлінга Хьюза (Sterling Hughes і Маркуса Бергера (Marcus Boerger).

І ось, нарешті, в липні 2004 року, виходить офіційний реліз PHP 5. В першу чергу, як і планувалося, переробці піддався весь механізм роботи з об'єктами. І якщо в попередніх версіях об'єктно-орієнтоване програмування на PHP було можливе в мінімальній мірі, із-за чого і використовувалося на практиці не часто, то PHP 5 володіє прекрасним потенціалом реалізації об'єктного програмування. Окрім цього, PHP

збагатився рядом коштовних розширень для роботи з XML, різними джерелами даних, генерації графіки і ін.

Недоліки:

а) Інтерпретований - безпека написаному на ньому коду вкрай погана.

б) Досить незручна система типізації.

в) Відносно довго виконуються функції та процедури інтерпретатора PHP.

г) Відсутня JIT компіляція.

д) Дуже проста і мало ефективна модель ООП.

е) Помилки дизайну мови.

ж) Відсутність послідовної ідеології мови.

з) Відсутність послідовної ідеології стандартної бібліотеки.

и) Дивацтва реалізації, викликані помилками в дизайні мови.

Переваги:

а) Дуже зручний для розробки простих і не довгострокових проектів.

б) Простий у навчанні.

в) Невимогливий до програмного забезпечення серверу.

г) Має можливість роботи як інтерпретатор та як модуль у HTTP сервері.

д) Має досить простий код та інструменти роботи з базами даних у порівнянні з аналогічними системами.

3.1.2 Ruby

Ruby - одна з наймолодших мов програмування. Його творець Юкихиро Мацумото (Yukihiro Matsumoto, також відомий під псевдонімом Matz), професійний японський програміст, розповідає: «Я почав розробку Ruby 24 лютого 1993 року. Перша hello world програма на Ruby заробила літом того ж року, а альфа-версія була готова до грудня 1994.» Назва мови відбувається від імені коштовного каменя рубіна (по аналогії з іншою широко поширеним мовою програмування Perl: pearl -перли).

От як характеризує Ruby його автор: «Це потужна і динамічна об'єктно-орієнтована мова з відкритими исходниками. Ruby працює на багатьох платформах, включаючи Linux і інші реалізації Unix, MS-DOS, Windows 9x/2000/NT, BEOS і MACOS.»

Головна мета Ruby - ефективність розробки програм, і користувачі знайдуть, що програмування на ньому ефективно і навіть забавно.

У Японії Ruby став популярним з моменту появи першої загальнодоступної версії в 1995 році, проте наявність документації лише на японській мові стримувала його подальше поширення. Лише у 1997 році з'явився опис Ruby англійською мовою, а в 1998 році відкрився форум ruby-talk. З цієї миті Ruby почав свій хід по всьому світу. За останній рік з'явилися три англомовні книги, присвячені йому [2-5], хоча на російську мову, на жаль, жодна з них ще не переведена. Зараз Ruby входить в більшість дистрибутивів ОС Linux, доступний він і користувачам інших операційних систем.

Переваги:

а) має простий синтаксис;

б) підтримує обробку виключень;

в) дозволяє перевизначити операторів;

г) є чисто об'єктно-орієнтованою мовою (complete, full, pure object oriented language) в якій, на відміну від Java або Perl, все - об'єкти;

д) дозволяє працювати з цілими числами довільної величини;

е) не вимагає оголошення змінних;

ж) використовує префікси (@, $, @@) для завдання зоні видимості (scope) змінних;

з) підтримує багато потокове програмування. Ruby є динамічною мовою. На відміну від статичних мов, подібних C++ або Java, методи і змінні в Ruby можуть бути додані або перевизначені під час виконання програми. Це дозволяє, наприклад, обійтися без директив умовної компіляції #ifdef, необхідних для мови C. Тут виявляється схожість Ruby з такими мовами, як Lisp і Smalltalk;

и) Ruby - універсальна і гнучка мова. З її допомогою можна однаково витончено реалізувати як те, що традиційно робиться за допомогою інтерпретатора Kornshell, так і те, що пишеться зазвичай на C.

Недоліки:

а) Ruby - мова, що інтерпретується;

б) досить повільна у роботі.

3.1.3 Python

Мова Python дозволяє швидко створювати як прототипи програмних систем, так і самі програмні системи, допомагає в інтеграції програмного забезпечення для вирішення виробничих завдань. Python має багату стандартну бібліотеку і велику кількість модулів

розширення практично для усіх нужд галузі інформаційних технологій. Завдяки ясному синтаксису вивчення мови не складає великої проблеми. Написані програми виходять структурованими формою, і в них легко прослідити логіку роботи. На прикладі мови Python розглядаються такі важливі поняття як: об'єктно-орієнтоване програмування, функціональне програмування, подієво-керовані програми (GUI- додатку), формати представлення даних (Unicode, XML і тому подібне). Можливість діалогового режиму роботи інтерпретатора Python дозволяє істотно скоротити час вивчення самої мови і перейти до вирішення завдань у відповідних наочних областях. Python вільно доступний для багатьох платформ, а написані на нім програми зазвичай переносимо між платформами без змін. Ця обставина дозволяє застосовувати для вивчення мови будь-яку наявну апаратну платформу. Python є мовою програмування, що інтерпретується. Вона надзвичайно проста і містить невелике число ключових слів, в той же час дуже гнучка і виразна. Це мова більш високого рівня ніж Pascal, C++ і, природно C, що досягається, в основному, за рахунок вбудованих високорівневих структур даних (списки, словники).

Переваги:

а) Безперечною перевагою є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах. Першою такою мовою була C, проте її типи даних на різних машинах могли займати різну кількість пам'яті і це служило деякою перешкодою при написанні дійсно переносимої програми. Python же таким недоліком не володіє. Наступна важлива межа - розширюваність мови, цьому надається велике значення і, як пише сам автор, мова була задумана саме як розширювана. Це означає, що є можливість вдосконалення мови всіма зацікавленими програмістами. Інтерпретатор написаний на C і вихідний код доступний для будь-яких маніпуляцій. У разі потреби, можна вставити його в свою програму і використовувати як вбудовану оболонку. Або ж, написавши на C свої доповнення до Python і скомпілювавши програму, отримати "розширений" інтерпретатор з новими можливостями.

б) Наявність великого числа модулів, що підключаються до програми, забезпечують різні додаткові можливості. Такі модулі пишуться на C і на самому Python і можуть бути розроблені всіма досить кваліфікованими програмістами. Як приклад можна привести наступні модулі:

1) Numerical Python - розширені математичні можливості, такі як маніпуляції з цілими векторами і матрицями;

2) Tkinter - побудова додатків з використанням графічного призначеного для користувача інтерфейсу (GUI);

3) OPENGL - використання обширної бібліотеки графічного моделювання двух- і тривимірних об'єктів Open Graphics Library фірми Silicon Graphics Inc. Даний стандарт підтримується, у тому числі, в таких поширених операційних системах як Microsoft Windows 95 OSR 2, 98 і Windows NT 4.0.

Єдиним недоліком, є порівняно невисока швидкість виконання Python-програми.

3.1.4 JAVA

Мова Java зародилася як частина проекту створення передового програмного забезпечення (ПЗ) для різних побутових приладів. Реалізація проекту була почата на мові C++, але незабаром виникли ряд проблем, найкращим засобом боротьби з якими була зміна самого інструменту - мови програмування. Стало очевидним, що необхідна незалежна мова програмування, що дозволяє створювати програми, які не доводилося б компілювати окремо для кожної архітектури і можна було б використовувати на різних процесорах під різними операційними системами.

У 1990 році розробник ПЗ компанії Sun Microsystems Патрік Нотон (Patrick Naughton) зрозумів, що йому набридло підтримувати сотні різних інтерфейсів програм, використовуваних в компанії, і повідомив виконавчого директора Sun Microsystems і свого друга Ськотту Макнілі (Scott McNealy) про свій намір перейти працювати в компанію NEXT. Макнілі, у свою чергу, попросив Нотона скласти список причин своєї незадоволеності і висунути таке вирішення проблем, неначебто він був Богом і міг виконати все, що завгодно.

Нотон, хоча і не розраховував на те, що хтось зверне увагу на його лист, все ж виклав свої претензії, нещадно розкритикувавши недоліки Sun Microsystems, зокрема, архітектуру, що розробляється в той момент, ПЗ NEWS. До здивування Нотона, його лист здобув успіх: він був розісланий всім провідним інженерам Sun Microsystems, які не забарилися відгукнутися і висловити гарячу підтримку своєму колезі і схвалення його поглядів на ситуацію в Sun Microsystems. Звернення викликало схвалення і у вищого керівництва компанії, а саме, у Біла Джоя (Bill Joy), засновника Sun Microsystems, і Джеймса Гослінга (James Gosling), начальника Нотона.

Того дня, коли Нотон повинен був піти з компанії, було прийнято рішення про створення команди провідних розробників з тим, щоб вони робили що завгодно, але створили щось незвичайне.

Команда з шести чоловік, з кодовою назвою Green, пішла в самовільне вигнання, занурившись в дослідження побутових пристроїв, таких як Nintendo Game Boys, пристроїв дистанційного керування. Команда Green намагалася знайти засоби, за допомогою яких можна було б встановити взаємодію між цими пристроями. Незабаром стало ясно, що такі електроприлади, як відеомагнітофони, програвачі лазерних дисків, стереосистеми - всі вони були реалізовані на різних процесорах. Це означало, що якщо виробник захоче додати телевізору або відеомагнітофону додаткові функції або характеристики, він буде затиснутий в рамках засобів, зашитих в апаратне забезпечення. Ця проблема, у поєднанні з обмеженістю пам'яті мікросхем цих пристроїв, висунула новий підхід до програмування ПЗ, який повинен був стати ведучим на ринку побутової електроніки.

Команда приступила до розробки нової об'єктно-орієнтованої мови програмування, яка була названа Oak (дуб), на честь дерева, що зростало під вікном Гослінга.

Незабаром компанія Sun Microsystems перетворила команду Green в компанію First Person. Нова компанія володіла цікавою концепцією, але не могла знайти їй відповідного вживання. Після ряду невдач несподівано ситуація для компанії різко змінилася: був анонсований Mosaic - так народився World Wide Web, з якого почався бурхливий розвиток Internet.

Нотон запропонував використовувати Oak в створенні Internet- додатків. Так Oak став самостійним продуктом, незабаром був написаний Oak-компилятор і Oak-браузер "WebRunner". У 1995 році компанія Sun Microsystems прийняла рішення оголосити про новий продукт, перейменувавши його в Java (єдине розумне пояснення назві - любов програмістів до кави). Коли Java опинилася в руках Internet, стало необхідним запускати Java-аплети - невеликі програми, що завантажуються через Internet. WebRunner був перейменований в HotJava і компанія Netscape встала на підтримку Java-продукту.

Переваги:

а) безпека;

б) об'єктна орієнтованість;

в) надійність;

г) інтерактивність;

д) незалежність від архітектури ЕОМ;

е) інтерпретація плюс висока продуктивність;

ж) простота вивчення.

Одним з головних недоліків мови Java традиційно вважається невисока швидкість роботи програм в порівнянні з додатками на мові C++.

3.1.5 PERL

Perl була розроблена Ларрі Уоллом (Larry Wall) в 1986 році, коли він був системним адміністратором одного проекту UNIX, пов'язаного із створенням багаторівневої безпечної мережі, що об'єднувала декілька комп'ютерів, рознесених на великі відстані. Робота була виконана, але було потрібно створення звітів на основі великого числа файлів з багато чисельними перехресними засланнями між ними.

Спочатку Ларрі передбачав використовувати для цих цілей фільтр awk, але виявилось, що останній не міг управляти відкриттям і закриттям великого числа файлів на основі тієї, що міститься в них же самих інформації про розташування файлів. Його першою думкою було написати спеціальну системну утиліту, яка зможе вирішити поставлене завдання, але пригадавши, що до цього йому вже довелося написати декілька утиліт для вирішення завдань, що не "беруться" стандартними засобами UNIX, він прийняв кардинальне рішення - розробити мову програмування, яка поєднувала б в собі можливості обробки текстових файлів (sed), генерації звітів (awk), вирішення системних завдань (shell) і низькорівневе програмування, доступне на мові C. Результатом цього рішення і з'явилася мова Perl, інтерпретатор для якого був написаний на C.

За твердженням самого Ларрі Уолла при створенні мови Perl їм рухала лінь - не у прямому розумінні, а в сенсі того, що для вирішення завдання, що стояло перед ним, слід було б написати велику кількість програм на різних мовах, що входять до складу інструментальних засобів UNIX, а це досить утомливе заняття.

Нова мова програмування поєднувала в собі можливості системного адміністрування і обробки файлів - два основні завдання, що вирішуються зазвичай при програмуванні в системі UNIX. Причому слід зазначити, що мова Perl з'явилася з практичних міркувань, а не через бажання створити ще один "красивий" засіб для роботи в UNIX, тому-то вона і набула широкого поширення серед системних адміністраторів, коли Ларрі Уолл надав її широкому кругу користувачів. З появою мови Perl з'явилася можливість вирішувати завдання за допомогою одного інструменту, і не витрачати час на вивчення декількох мов середи програмування UNIX.

Переваги:

а) Perl - кросплатформена мова програмування, програми написані на Perl працюють на UNIX, Windows, Macintosh, VMS і інших платформах.

б) Perl має вбудовані потужні механізми роботи з текстовими даними, регулярні вирази - невиключна частина мови.

в) Можна писати програми як за допомогою процедур, так і об'єктів.

г) Інтеграція з різними базами даних за допомогою єдиного інтерфейсу (DBI).

д) Підтримує Unicode (Підтримка Unicode).

е) Дозволяє використовувати зовнішні бібліотеки, написані на інших мовах, за допомогою XS або SWIG.

ж) Величезна кількість готових модулів для вирішення найрізноманітніших завдань (Модулі Perl).

Недоліком є те, що Perl - скриптова мова (що інтерпретується), що дуже зручно в WWW, але зовсім не зручно для стандартного користувача.

3.1.6 HTML

Початок історії HTML слід віднести до далекого 1986 року, коли Міжнародна організація по стандартизації (ISO) прийняла стандарт ISO-8879, "Standard Generalized Markup Language (SGML)". Стандарт цей присвячений опису SGML - узагальненої метамови, що дозволяє будувати системи логічної, структурної розмітки будь-яких різновидів текстів. Слово "структурна" означає, що коди, що управляють, вносяться до тексту при такій розмітці, не несуть жодної інформації про зовнішній вигляд документа, а лише вказують кордони і супідрядність його складових частин - тобто задають його логічну структуру.

Творці SGML прагнули максимально абстрагуватися від проблем представлення електронного тексту в різних програмах, на різних комп'ютерних платформах і пристроях виводу. Так, якщо за допомогою SGML розмічається документ, що містить заголовки, ідеологія мови забороняє вказувати, що такий-то заголовок повинен набиратися, скажімо, шрифтом Times напівжирного зображення кегля 12 пунктів. SGML у такому разі вимагає обмежитися вказівкою на рівень заголовка і його місце в ієрархічній структурі документа.

Завдяки таким обмеженням розмічений текст зможе без зусиль інтерпретувати будь-яка програма, що працює з будь-яким пристроєм виводу. Наприклад, при роботі в графічному інтерфейсі заголовок може дійсно виводитися напівжирним шрифтом підвищеного кегля; програма, що використовує текстовий інтерфейс, виділить його порожнім рядком зверху і знизу і, можливо, підвищеною яскравістю символів; синтезатор мови, що читає документ вголос, зможе відзначити заголовок паузою і зміною інтонації.

Можна сказати, що SGML-разметка оголює нематеріальну "душу" тексту, для якої згодом будь-яка програма-інтерпретатор зможе підібрати відповідне до випадку "тіло".

Проте абстрактність SGML цим не вичерпується. SGML є не готовою системою розмітки тексту, а лише зручна метамова, що дозволяє будувати такі системи для конкретних обставин. Життя багатообразне і непередбачуване: сьогодні вам потрібно виділяти в текстах заголовки, а завтра, можливо, знадобиться розмічати підписи в листах, математичні формули або імена дійових осіб в п'єсі. Стандарт SGML визначає лише синтаксис запису елементів розмітки - тегов - і їх атрибутів, а також правила визначення нових тегов і вказівки структурних стосунків між ними. Для практичної ж розмітки документів потрібний додаток SGML - набір визначених відповідно до стандарту тегов, що є, по суті, формальним описом структури документа.

Мова SGML - це типове дітище академічної науки, витончена іграшка теоретиків. Її створення не було викликане насущною практичною необхідністю. Принципи, на яких будується ця мова цікаві; поза сумнівом, ідеологія SGML зробила вплив на багато комп'ютерних розробок. Проте сам по собі SGML не набув скільки-небудь помітного поширення - до тих пір, поки в 1991 р. співробітники Європейського інституту фізики часток , зайняті створенням системи передачі гіпертекстової інформації через Internet, не вибрали SGML як основу для нової мови розмітки гіпертекстових документів. Ця мова - найвідоміше з додатків SGML - була названий HTML (HyperText Markup Language, "мова розмітки гіпертексту").

Спочатку HTML, як і належить SGML-додатку, розділяла всі особливості ідеології SGML. Вся розмітка була чисто логічною, і лише в описовій частині стандарту, супроводжуючій формальне визначення тегов, можна було прочитати що-небудь начеб "в графічних браузерах дія цього тега може передаватися курсивним зображенням".

А першим (і довгий час єдиним) графічним браузером в ті далекі часи була програма Mosaic, розроблена, як і власне WWW, в науковій установі - Національному центрі суперкомп'ютерних додатків США (National Center for Supercomputer Applications, NCSA).Отже немає нічого дивного в тому, що в це "золоте століття" жодних протиріч між офіційними стандартами і їх реалізацією в браузерах ще не існувало. HTML неквапливо розвивалась, залишаючись в рамках парадигми структурної розмітки, і в квітні 1994 р. почалася підготовка специфікації наступної версії мови - 2.0. Цим займався утворений в тому ж році Консорціум W3 (W3 Consortium, скорочено W3C;),що успадкував від CERN верховну владу і авторитет в світі WWW.

Зараз консорціум, що має статус "міжнародної некомерційної організації", об'єднує понад 150 організації-членів, у тому числі фірми Netscape, Microsoft і безліч інших. Проте в 1994-1995 рр. його членами були майже виключно університети і наукові установи. Настільки "академічний" склад W3C позначався як на самих документах, що публікуються консорціумом, так і на процедурі (і особливо на термінах) їх прийняття.

Досить сказати, що остаточний варіант HTML 2.0, єдиним серйозним удосконаленням в якому був механізм бланків (форм) для відсилання інформації з комп'ютера користувача на сервер, був офіційно затверджений лише у вересні 1995 р., коли в W3C вже повним ходом йшло обговорення HTML 3 (або, як його називали спочатку, "HTML+").

Переваги:

а) простота;

б) недоліки;

в) мізерність і статичність сторінок.

3.1.7 CSS

Каскадні таблиці стилів або CSS (від англійського Cascading Style Sheets) є наслідком подальшого розвитку HTML і дають нам можливість перейти на наступний рівень представлення інформації. Таблиці стилів дозволяють розділити смисловий вміст сторінки і його оформлення.

3.1.8 MySQL

Винахідником MYSQL є Михайло Віденіус aka Monty з шведської компанії TCX. У 1979 році він розробив засіб управління базами даних, який називався UNIREG. Надалі UNIREG була розширена для підтримки великих баз даних і була переписана на декількох мовах. У 1994 році компанія TCX почала розробляти додатки для www з використанням UNIREG. Проте у зв'язку з великими накладними витратами UNIREG не могла успішно використовуватися для динамічної генерації Web-страниц. Тому Віденіус вирішив зв'язатися з автором mSQL, Хьюзом, щоб запропонувати йому підключити mSQL до обробника B+ ISAM в UNIREG. Проте Хьюз успішно просунувся на шляху до mSQL 2, і компанія вирішила створити сервер баз даних під свої потреби.

У TCX узяли за основу UNIREG і використовували утиліти сторонніх розробників для mSQL, написали API для своєї системи, який спочатку сильно збігався з API для mSQL. Проте це дозволяло будь-якому користувачеві mSQL, що бажає перейти на сервер баз даних TCX, внести до свого коду незначні зміни. Вихідний код нової бази даних був повністю оригінальних. Таким чином, в травні 1995 року в компанії була база даних MYSQL 1.0 що повністю задовольняє потребам компанії.

Що стосується назви, то Віденіус говорить про це так: «До кінця не ясно, звідки йде назва MYSQL. У TCX базовий каталог, а також значне число бібліотек і утиліт протягом десятка років мали префікс “my„. В той же час мою дочку (яка на декілька років молодше) теж звуть My. Тому залишається таємною, яке з двох джерел дало назву MYSQL».

MYSQL перенесена на багато ОС UNIX, під Win32 і OS/2 з моменту випуску в Інтернет і є платформою баз даних, що швидко розвивається, завдяки безлічі програмістів, зацікавлених в її розвитку.

Переваги:

а) безкоштовність;

б) відкритий код;

в) простота установки;

г) простота налаштування;

д) простота вивчення;

е) простота використання;

ж) зрозумілі API для багатьох мов програмування;

з) поширеність.

3.1.9 Технологічне програмування

Techno FBD призначений для інженерів-технологів, які вирішують завдання управління технологічним процесом. Важко придумати наочніший засіб програмування контурів управління і регулювання. Програма на Techno FBD є схемою, що складається з набору функціональних блоків, зв'язаних між собою через входи і виходи. У TRACE MODE® 6 включено більше 150 типових функціональних блоків, що реалізовують широкий набір функцій, - від простих логічних операцій до готового адаптивного регулювальника. Фільтрація, ПІД, ПДД, модальне, нечітке, позиційне регулювання, статистичні, тригонометричні, а також блоки управління клапаном, засувкою, мотором - все це реалізовано у вигляді стандартних FBD-блоку TRACE MODE® 6.

Techno LD сподобається інженерам, звиклим до складання схем релейної логіки. Зовні редактор LD дуже схожий на редактор FBD, лише замість функціональних блоків користувачеві пропонується використовувати "контакти" і "котушки". За бажання в схему Techno LD можна включати і вкраплення із звичайних блоків FBD, хоча ця можливість TRACE MODE® 6 дещо виходить за рамки стандарту.

Techno ST орієнтована перш за все на програмістів, вонан є мовою програмування високого рівня, схожа на Паскаль. У ній підтримуються масиви (у тому числі багатовимірні), контроль перетворення типів, присутні такі конструкції як DO-WHILE, REPEAT-UNTIL, FOR-TO-DO, IF-THEN-ELSE, CASE-OF та інші інтуїтивно зрозумілі будь-якому програмістові оператори. Заголовок програми будується автоматично за списком аргументів, що дозволяє економити час на оформленні програми. Службові слова, мітки, коментарі і числові константи виділяються кольором, як це прийнято в сучасній середі розробки, налаштування кольорів доступні користувачеві.

Techno IL- це проста мова мнемонічних інструкцій, що зовні нагадує асемблер. Ця мова була включена в стандарт для програмування контролерів, що володіють низькою обчислювальною потужністю. Програми IL легко транслюються в машинні коди будь-якого процесора, що дозволяє створювати дуже швидкі програми. Проте, на сьогоднішній день проблема продуктивності давно вирішена, і реально Techno IL не має жодних переваг перед ST або FBD, тим більше, коли йдеться про програмуванні операторської станції. Проте, ця мова була включена в TRACE MODE® 6 для підтримки застарілого устаткування. Сам по собі цей факт зайвий раз підкреслює перевагу стандарту МЕК 6-1131/3 перед хай досконалішими, але локальнішими засобами автоматизації.

Techno SFC - це потужний засіб структуризації складних алгоритмів. По суті SFC не є самостійною мовою. У перекладі з англійського абревіатуру SFC можна перевести як "схема функціональної послідовності". Зовні програма на Techno SFC схожа на блок-схему алгоритму, на якій відображують окремі програмні блоки (кроки), переходи між ними і умови, по яких виконуються ці переходи. Кожен програмний блок, як і кожна умова переходу - це підпрограма на будь-якій з мов стандарту МЕК 6-1131/3. Ця мова дуже зручна для програмування стадійних (batch) процесів, систем дозування і бізнес-додатків. Techno SFC може бути легко використаний як інженерами, так і бізнес-аналітиками

3.2 Дослідження технологій розробки програмного забезпечення серверу регіональної системи екологічного моніторингу

3.2.1 Apache

Популярний веб-сервер, використовуваний на більшості серверів в інтернет; включає великий набір розширень. Одним з найбільш успішних продуктів OpenSource є Web-сервер Apache. Цей сервер з'явився в квітні 1995 року, вже через рік він був найбільш популярним ПЗ для підтримки WWW- серверів з "ринкової долі" близько 35%. Половина ринку була освоєна на початок 1998 року, починаючи з цього часу, доля Apache більш ніж удвічі перевищує долю найближчого конкурента - Microsoft IIS, і вагається в межах 55-65%. На сьогоднішній день Apache підтримує функціональність приблизно 19 млн. WWW - серверів.

Такий безперечний успіх провокує на уважний аналіз - повторити успіх Apache було б приємно будь-якій групі розробників OpenSource-проекту.

Причини успіху Apache можна розділити на дві групи: технологічні причини, пов'язані з технічними перевагами над конкурентами, і нетехнологічні.

Технологічні причини успіху Apache:

- Технологічне лідерство. У початковий період своєї історії Apache був одним з технологічних лідерів на ринку - продуктивність була досить велика, потреби в ресурсах-малі. При цьому програма володіла можливістю легкого розширення шляхом додавання модулів, причому реалізовано це було багато краще, ніж у конкурентів. Можливість легкого розширення зумовила поява великої кількості похідних від Apache серверів - Stronghold, Apache/SSL, mod_perl - потреб Web - розробників, що задовольняють більшість, по сьогоднішній день.

- Технологічний консерватизм. Автори популярних програм швидко виявляються заваленими побажаннями користувачів. Якщо їм слідувати, то програми перевантажуються функціональністю, потрібною лише малому числу клієнтів, складність кодів зростає одночасно з числом проблем і так далі. Авторам Apache удалося зберегти необхідний баланс в цій області. ПЗ, що розробляється ними, має репутацію стабільного і передбаченого.

- Відвертість процесу розробки. Процес розробки Apache відкритий для спостереження і коментування і тому передбачений. Це дозволяє випускати додаткові модулі до нових версій практично одночасно з їх виходом.

Нетехнологічні причини:

- “Демократична” розробка. У проекті Apache реалізована унікальна схема розробки - по кожній зміні проводиться голосування, причому "істотні" зміни можуть бути зупинені правом вето будь-якого члена групи розробників, а неістотні - повинні набрати більше голосів "за", чи "проти". Така схема дозволяє блокувати сумнівні технологічні рішення, підтримуючи технологічний консерватизм. Очевидно, що така схема не може бути застосована в довільному проекті оскільки вимагає розумності від всіх розробників групи, що досяжно далеко не завжди.

- Підтримка користувачів. Не дивлячись на величезну призначену для користувача базу і некомерційний статус, підтримка Apache була і залишається дуже хорошою за якістю - повідомлення про проблеми аналізуються протягом 1-2 днів.

- Ліцензування. Істотною причиною успіху Apache є дійсно вільне ліцензування. Apache License, на відміну від найбільш поширеною в середі OpenSource GNU GPL, не нав'язує вільне поширення похідних робіт, а вимагає лише збереження права на ім'я - вказівки, що похідний проект використовує код, розроблений Apache Group. При такій схемі ліцензування комерційні компанії більш охоче вкладають свої ресурси в розвиток продукту, прикладом може служити участь IBM в розробці Apache 2.0 і перенесенні Apache на платформу Windows.

Всі перераховані причини успіху Apache є істотними, відсутність будь-якою з них помітно погіршило б продукт в очах частини, або всіх користувачів.

Перевагами Apache вважаються надійність і гнучкість конфігурації. Він дозволяє підключати зовнішні модулі для надання даних, використовувати СУБД для аутентифікації користувачів, модифікувати повідомлення про помилки і так далі Підтримує IPv6

Недолік -відсутність зручного стандартного інтерфейсу для адміністратора.

3.2.2 MYSQL

У 1995 році Девід Оксмарк, що працює в компанії Detron HB і є бізнес-партнером фірми, почав «тиск» на TCX з тим, щоб вона почала поширювати СУБД MYSQL через Інтернет. Крім того, Девід взяв участь в роботі над документацією. Версія 3.11.1 СУБД MYSQL була випущена в 1996 році у вигляді бінарного дистрибутива для роботи під управлінням ОС Linux і Solaris. Сьогодні MYSQL працює на багатьох платформах і доступна як в двійкових кодах, так і у вихідних текстах.

Одна з причин популярності MYSQL серед користувачів PHP полягає в тому, що підтримка цього сервера включається в постачання PHP. Завдяки хорошим характеристикам і великому набору стандартних інтерфейсних функцій, дуже простих у використанні, MYSQL став найпопулярнішим засобом для роботи з базами даних в PHP.

Ліцензійна політика MYSQL відрізняється більшою гнучкістю порівняно з іншими серверами баз даних. По суті, MYSQL поширюється безкоштовно за винятком тих випадків, коли маєте намір її продавати або продавати послуги, що створюються з її допомогою.

MYSQL володіє відмінною переносимістю і може, з тим же успіхом, використовуватися на комерційних операційних системах, таких як Solaris, Irix або Windows, і на будь-якій апаратурі аж до потужних серверів. Більш того, так само як і її «дорожчі суперники», вона дозволяє обробляти великі бази даних, що містять мільйони записів.

Переваги:

а) швидкість;

б) стійкість;

в) легкість у використанні.

Недоліки:

У MYSQL відсутні:

а) підтримка вкладених запитів, типа SELECT * FROM table1 WHERE id IN (SELECT id FROM table2). Затверджується, що така можливість буде у версії 3.23;

б) не реалізована підтримка транзакцій. На зміну цьому пропонується використовувати LOCK/UNLOCK TABLE;

в) немає підтримки зовнішніх (foreign) ключів;

г) немає підтримки тригерів і процедур, що зберігаються;

д) немає підтримки представлень (VIEW). У версії 3.23 планується можливість створювати представлення.

3.2.3 PHP

Якщо для роботи PHP використовується модуль веб-сервера apache mod_php.

Переваги:

а) найвища швидкість роботи скрипту, в порівнянні з іншими методами;

б) простота роботи, сервер сам обробляє скрипт;

в) спільний конфігураційний файл для всіх скриптів (php.ini);

г) можливість завдання змінних конфігурації PHP в конфігураційному файлі web- серверу або засобами файлу .htaccess.

Недоліки:

а) Всі скрипти запускаються з правами, з яким працює web- сервер, тим самим якщо є необхідність запису в директорію - права доступу необхідно дати на неї всім.

б) В разі запуску сторонніх додатків скриптами (наприклад, поштова розсилка), немає можливості ідентифікувати користувача, який запустив процес.

в) Зайве навантаження на web-сервер, Apache зайнятий обробкою скриптів може повільно віддавати інші статичні дані.

г) Помилки в скрипті можуть привести до непрацездатності всього web-сервера.

Якщо PHP як CGI використовується, тоді маємо запуск PHP-скрипту через його передачу на виконання безпосередньо інтерпретатору PHP.

Переваги:

а) всі скрипти виконуються з правами користувача - власника www-домену;

б) можливість індивідуального налаштування PHP для кожного користувача;

в) менша витрата оперативної пам'яті в порівнянні з модулем Apache;

г) помилки в скрипті не приводять до падіння веб-сервера.

Недолік - проблеми з авторизацією засобами PHP (засобами команди Header) унаслідок того, що не передаються деякі змінні сервера php-скрипту.

Якщо маємо PHP як FASTCGI, при цьому використовується модуль Apache mod_fastcgi, тоді скрипти передаються його засобами на вхід інтерпретатора PHP.

Перевага – за рахунок кешування деяких проміжних даних скрипт не інтерпретується кожного разу при виконанні і досягається вища швидкість в порівнянні з PHP.

Недолік - зайвий процес користувача (php-cgi) знаходиться в пам'яті після першого звернення до процесу.

3.2.4 IIS

Це набір серверів для декількох служб Інтернету від компанії майкрософт. IIS поширюється з операційними системами сімейства Windows NT.

Дозволяє розміщувати в Інтернеті сайти Всвітньої павутини. IIS підтримує протоколи HTTP, HTTPS, FTP, POP3, SMTP.

За даними компанії Netcraft на жовтень 2008 року більше 62 з половиною мільйонів сайтів обслуговуються IIS. Основним компонентом IIS є служба WWW (звана також W3SVC), яка надає клієнтам доступ до сайтів Всвітньої павутини по протоколах HTTP і, якщо налагоджено HTTPS один сервер IIS може обслуговувати декілька сайтів (IIS 6.0 і вище). Кожен сайт має наступні атрибути: IP- сайту; TCP- на якому служба WWW чекає підключень до даного сайту; Заголовок вузла (Ім'я рідної магістралі) - значення заголовка, вказуюче зазвичай DNS -ім'я сайту.

Таким чином, наприклад, один сервер з одним IP- може обслуговувати на одному TCP- декілька сайтів. Для цього необхідно створити декілька DNS записів і розрізняти сайти по заголовках вузла.

Для кожного сайту вказується домашній каталог - каталог у файловій системі сервера, відповідний «Корню» сайту.

Реалізація для IIS. IIS підтримує декілька різних технологій створення ASP.NET - розроблена компанією майкрософт технологія; для IIS це - основний на сьогоднішній день засіб створення і. IIS 6.0 поставляється разом з операційними системами, в які також спочатку входить .NET структура так, що підтримка ASP.NET неначе вже вбудована в IIS 6.0; для раніших версій необхідно окремо завантажити і встановити .NET.

КОБРА - передуюча ASP.NET технологія створення динамічних веб-сторінок на основі сценаріїв. Входить в постачання IIS починаючи з версії 3.0.

CGI - стандартна міжплатформена низькорівнева технологія створення.

ISAPI - низькорівнева технологія, аналогічна інтерфейсу модулів Апач, надаючи повний доступ до всіх можливостей IIS, можливість розробки веб-додатку в машинному коді і можливість перевизначення частини функцій IIS і додавання до нього функцій, як пов'язаних з генерацією контенту, так і не пов'язаних з цим. Підсистема виконання скрипту КОБРА і підсистема ASP.NET, виконані як модулі ISAPI.

Перший ступінь інтеграції - включення в одні сторінки тексту з інших сторінок. Строго кажучи, немає, оскільки IIS підтримує лише обмежений набір можливостей і без того мало функціонального першого ступіню інтеграції. Зокрема, IIS5 підтримує лише статичне включення.

За допомогою CGI додатки для IIS можуть розроблятися на основі практично будь-яких, у тому числі сторонніх, інструментів, що допускають запис в стандартний потік

виводу і читання змінних середи (Perl,C++ і навіть засобами інтерпретатора командного рядка Cmd.exe).

Технологія ISAPI дозволяє, з одного боку, створювати спеціальні додатки для IIS, що вимагають особливо тісної взаємодії з механізмом сервера з іншого боку є зручною платформою для організації ефективної взаємодії IIS з іншими технологіями розробки, - наприклад, PHP і Perl.

3.2.5 ASP.NET

Першою технологією, що дозволила створювати динамічні веб-сайти став інтерфейс CGI, який дозволяв звичайним програмам формувати вміст веб-сторінки і повертати користувачеві результат у вигляді HTML-сторінки. Цей спосіб створення динамічних веб-сайтів популярний до цих пір. Найпоширенішими мовами програмування в цій області є Perl, Python і Delphi.

Використання CGI має ряд недоліків: складність відділення програмного коду від HTML-відображення, відсутність вбудованих засобів для створення призначеного для користувача інтерфейсу і як наслідок - висока вартість розробки веб-сайту. При створенні веб-сайту доводиться або все писати самому, або використовувати готове рішення, яке досить важко пристосувати до конкретних умов.

Наступним поколінням в розробці динамічних веб-сайтів став інтерфейс ISAPI. З точки зору розробки він мало чим відрізняється від CGI, проте вирішено багато проблем з продуктивністю і масштабованістю.

Гіпертекстові препроцесори стали новим рівнем еволюції серверних технологій. Основна їх відзнака - можливість вставляти програмний код прямо в HTML-страницу. Це дозволило значно спростити процес розробки. Найпопулярнішим гіпертекстовим препроцесором є PHP. Другий по популярності ASP від Microsoft, який значно поступається PHP в спільному випадку, проте має деякі переваги при роботі з технологіями Microsoft. Недолік препроцесорів в тому, що вони використовують скриптові мови, які мають мало можливостей і невисоку продуктивність.

Найсучасніший підхід до створення веб-додатку- це використання технологій JSP і ASP.NET. При цьому ASP.NET є новішою. В JSP є одна перевага – кросплатформеність (може працювати і на Windows і на Unix серверах). Втім, є кросплатформена реалізація технології ASP.NET під назвою Mono. Проте, в основному вибір технології зводитися до вибору в'язки: JSP+Linux або ASP.NET+Windows. І не дивлячись на те, що Linux є

безкоштовною операційною системою, все більше компаній і розробників вибирають саме ASP.NET+Windows.

ASP.NET є частиною Microsoft .NET Framework, її можна викачати на сайті Microsoft безкоштовно. Знадобитися викачувати ASP.NET лише, якщо треба встановити свій веб-сервер. При цьому звичайно доведеться купити Windows.

Вартість хорошого платного хостингу в Україні майже однакова для Linux і для Windows. Windows-хостинг - parking.ru, all4hosting.ru, 1gb.ru, capitalhost.ru. Безкоштовний тестовий - webmatrixhosting.ru.

Для ASP.NET існують безкоштовні засоби розробки (Visual WebDeveloper Express, WebMatrix, SharpDevelop).

Переваги

а) об'єктно-орієнтований підхід;

б) підтримка візуальних компонентів, що інкапсулюють не лише виведення HTML, але і Javascript і навіть AJAX. Окрім вбудованих, існують компоненти сторонніх виробників. Є можливість успадковувати свій компонент від чужого і покращувати його, адаптуючи під свої потреби;

в) вбудовані візуальні компоненти для редагування і відображення даних, навігації, авторизації і так далі;

г) вбудовані засоби кешування, моніторингу, що дозволяють підвищити продуктивність і масштабованість додатків;

д) повноцінна компільована мова програмування (C#, VB.NET, Delphi, J#);

е) зручна середа розробки і відладки додатків;

ж) шаблони дизайну, призначені для користувача елементи управління, модель codebehind - дозволяють використовувати багато разів один і той же код;

з) вбудована авторизація і аутентифікація;

и) вбудована підтримка веб-сервисів.

3.2.6 AJAX

AJAX (від англ. Asynchronous Javascript and XML - «асинхронний JavaScript і XML») - це підхід до побудови інтерактивних призначених для користувача інтерфейсів веб-додатку, що полягає в «фоновому» обміні даними браузера з веб-сервером. В результаті при оновленні даних веб-сторінка не перезавантажується повністю, і веб-сервера-додатка стають швидшими і зручнішими.

Технологія AJAX - це не самостійна технологія, а концепція використання декількох суміжних технологій. AJAX базується на двох основних принципах: використання технології динамічного звернення до сервера «на льоту», без перенавантаження всієї сторінки повністю. Вперше термін AJAX був публічно використаний 18 лютого 2005 року в статті Джесси Джеймса Гарретта (Jesse James Garrett) «Новий підхід до веб-додакту». Гарретт придумав термін, коли йому довелося якось назвати новий набір технологій, пропонований їм клієнтові.

Проте в тій або іншій формі багато технологій були доступні і використовувалися набагато раніше, наприклад в підході «Remote Scripting», запропонованим компанією Microsoft в 1998 році, або з використанням HTML елементу IFRAME, що з'явився в Internet Explorer 3 в 1996 році. AJAX став особливо популярний після використання його компанією Google в сервісах Gmail, Google Maps і Google Suggest.

Переваги:

а) Використання AJAX дозволяє значно скоротити трафік при роботі з веб-сервером-додатком завдяки тому, що часто замість завантаження всієї сторінки досить завантажити лише невелику частину, що змінилася.

б) Зменшення навантаження на сервер.

в) Прискорення реакції інтерфейсу.

г) Оскільки потрібно завантажити частину, що лише змінилася, то користувач бачить результат своїх дій швидше.

Недоліки:

а) Інтеграція із стандартними інструментами браузера.

б) Динамічно створювані сторінки не реєструються браузером в історії відвідин сторінок, тому не працює кнопка «Назад», що надає користувачам можливість повернутися до переглянутих раніше сторінок.

в) Інший недолік зміни вмісту сторінки при постійному URL полягає в неможливості збереження закладки на бажаний матеріал. Частково вирішити ці проблеми можна за допомогою динамічної зміни ідентифікатора фрагмента (частини URL після #), що дозволяють багато браузерів.

г) Динамічно завантажуваний вміст недоступний пошукачам Пошукові машини не можуть виконувати JavaScript, тому розробники повинні поклопотатися про альтернативні способи доступу до вмісту сайту.

д) Старі методи обліку статистики сайтів стають неактуальними.

е) Багато сервісів статистики ведуть облік переглядів нових сторінок сайту, для сайтів сторінки яких широко використовують AJAX, така статистика втрачає актуальність.

3.2.7 UML

UML (Unified Modeling Language), який став стандартом де-факто" в області розробки програмного забезпечення і застосовується для вирішення завдань інших наочних областей, наприклад, завдань бізнес-моделювання.

У основі UML лежить декілька об'єктно-орієнтованих методів, кожен з яких спочатку був орієнтований на підтримку окремих етапів об'єктно-орієнтованого аналізу і проектування (ООАП). Метод Граді Золить (Grady Booch), умовна назва Booch (Booch'91, Booch Lite, Booch'93) - вважався найбільш ефективним на етапах проектування і розробки програмних систем. Метод Джеймса Румбаха (James Rumbaugh), Object Modeling Technique (OMT, пізніше OMT-2) - оптимальний для аналізу процесів обробки даних в інформаційних системах. Метод Айвара Джекобсона (Ivar Jacobson), Object-Oriented Software Engineering (OOSE) – містив засоби представлення варіантів використання, що мають істотне значення на етапі аналізу вимог в процесі проектування бізнес-додатків.

Історія розвитку UML датується 1994 р., коли почалася інтеграція/унифікація вищезгаданих методів. Проект уніфікованого методу (Unified Method) версії 0.8 був опублікований в жовтні 1995 р.

Всі питання розробки і супроводу мови UML сконцентровані в рамках консорціуму OMG (Object Management Group). Хоча OMG був створений з метою розробки пропозицій по стандартизації об'єктних і компонентних технологій CORBA, мова UML придбала статус другого стратегічного напряму в роботі консорціуму. У листопаді 1997 р. OMG оголосив UML стандартною мовою об'єктно-орієнтованого моделювання і перейняв на себе обов'язки по її подальшому розвитку. Група фахівців забезпечує публікацію описів подальших версій мови UML і запитів пропозицій RFP (Request For Proposals) по її стандартизації. Статус мови UML визначений як відкритий для всіх пропозицій по доопрацюванню і удосконаленню.

3.2.8 SCADA

SCADA - система диспетчерського контролю і збору даних.

Основні завдання, вирішувані SCADA:

а) обмін даними з ПЗО (пристрої зв'язку з об'єктом, тобто з промисловими контролерами і платами вводу/виводу) в реальному часі через драйвери;

б) обробка інформації в реальному часі;

в) відображення інформації на екрані монітора в зрозумілій для людини формі;

г) ведення бази даних реального часу з технологічною інформацією;

д) аварійна сигналізація і управління тривожними повідомленнями;

е) підготовка і генерування звітів про хід технологічного процесу;

ж) здійснення мережевої взаємодії між SCADA ПК;

з) забезпечення зв'язку із зовнішніми додатками (СУБД, електронні таблиці, текстові процесори і т. д.). У системі управління підприємством такими додатками найчастіше є додатки, відношувані до рівня MES;

и) SCADA- дозволяють розробляти АСУ ТП в розподіленій архітектурі.

Інколи SCADA комплектуються додатковим ПЗ для програмування промислових контролерів. Такі SCADA називаються інтегрованими і до них додають термін SoftLogIn.

Термін SCADA має двояке тлумачення. Найширше поширено розуміння SCADA - як програмного комплексу, що забезпечує виконання вказаних функцій, також інструментальних засобів для розробки цього програмного забезпечення. Проте, часто під SCADA- мають на увазі комплекс.

Термін SCADA еволюціонував разом з розвитком технологій автоматизації і управління технологічними процесами. У 80-і роки під SCADA - частіше розуміли комплекси збору даних реального часу.

3.3 Дослідження C# та PHP

Технологія розробки ASP.NET реалізована з базовою мовою програмування C#. Отже можна стверджувати, що швидкість роботи будь-якого проекту, розробленого на ASP.NET, буде залежити від швидкості компілювання та виконання коду мови C#.

Технологія розробки PHP реалізована з використанням мови програмування PHP. Отже, для того щоб відповісти на питання , яка технологія працює швидше за іншу треба дослідти роботу кожної з них.

Для цього розробимо програму тестування для ASP.NET та PHP.

Програми тестування швидкості роботи для технології ASP.NET та PHP відображено у додатку А.

У таблиці 3.1, 3.2 відображено результати тестування обох технологій, відповідно.

Таблиця 3.1 - Результати дослідження мови C#

Назва операції	Час всього тесту, мС	Час однієї ітерації, мС	Кількість ітерацій за одну секунду	Потреба в пам'яті кБ
Empty loop (checked)	77,3663	0,0000039	258 510 631	0
Empty loop (unchecked)	50,6916	0,0000025	394 542 388	0
Light loop (checked)	75,899	0,0000038	263 507 972	0
Light loop (unchecked)	75,758	0,0000038	263 998 687	0
Reflection method call (A)	150,1856	0,0750928	13 317	32,8
Reflection method call (B)	113,066	0,0565330	17 689	73,7
Virtual method call	57,357	0,0000287	34 869 321	8,2
Normal method call	117,994	0,0000059	169 500 077	8,2
Static method call	76,9989	0,0000038	259 743 996	0
String append (A)	250,5852	0,0250585	39 907	379
String append (B)	127,7419	0,0638710	15 657	398,6
String builder (A)	160,7238	0,0001607	6 221 855	3145,8
String builder (B)	93,2476	0,0004662	2 144 827	6291,5
Class construction	53,7755	0,0002689	3 719 162	4805,7
Class constr. & destr.	71,6926	0,0003585	2 789 687	0
Struct construction	134,4791	0,0000672	14 872 199	32000
Struct constr. & destr.	136,5715	0,0000683	14 644 339	0
Integers (checked)	113,9136	0,0000570	17 557 160	0
Integers (unchecked)	113,5672	0,0000568	17 610 714	0
Floats (checked)	86,0012	0,0004300	2 325 550	0
Floats (unchecked)	84,5987	0,0004230	2 364 101	0

Таблиця 3.2 - Результати дослідження язику PHP

Назва операції	Час всього тесту, мС	Час однієї ітерації, мС	Кількість ітерацій за одну секунду
Light loop	408,3409	0,0020417	489787
Virtual method call	326,565	0,0163283	61244
Normal method call	327,827	0,0163913	61008
Function call	122,221	0,0061110	163638
String append (A)	329,671	0,0016484	606665
String append (B)	370,111	0,0018506	540378
Class construction	227,898	0,0113949	87759
Class constr. & destr.	300,503	0,0150252	66555
Ints	406,113	0,0020306	492474
Floats	288,3281	0,0028833	346827

Середовище дослідження:

а) Microsoft (R) Visual C# .NET Compiler version 7.00.9466;

б) Microsoft (R) .NET Framework version 1.0.3705;

в) Compiler options: "Release" default;

г) PHP 4.2.3 з ZendOptimizer.

Результати приведені для консольних версій програм: Для C# - Release- версія консольного застосування, для PHP - скрипт виконувався командою, php -C -q -c "C:\WINDOWS" SpeedTest.php.

Окрім цього тестувалися ASP.NET & PHP as Apache2 module - версії.

Результати для них відрізнялися в гіршу сторону на 3-7%.

Для ASP.NET- тестувалися 2 версії:

а) ASP.NET .aspx- сторінка;

б) ASP.NET Web Application.

Для обох версій були однакові (різниця <1%) результати.

Спільні зауваження по C# - тестам:

Checked & Unchecked - версії:

Checked: весь тест виконувався в режимі checked {...} (перевірки на переповнювання діапазону значень для арифметичних і інших операцій для всіх value-типів (int, long, double, etc...) включені).

Unchecked: весь тест виконувався в режимі unchecked {...} (перевірки на переповнювання діапазону значень для арифметичних і ін. операцій для всіх value-типов (int, long, double, etc...) вимкнені). Не вказано - значить, unchecked. Ні у одному тесті реально не відбувається переповнювання діапазону.

У тести завжди входить час на виконання інструкцій циклу:

for (i=0; i<XXX; i++) {...}

Тестів досить багато - C# тестувався детальніше. Тести XXX method call за 1 ітерацію (pass) роблять 3 виклики методів крім того, виконується складання (по суті, в тест входить час LightLoop(unchecked)).

Зауваження per-test:

Empty-loop: час виконання однієї ітерації циклу: for (i=0; i<XXX; i++) {...}

Light-loop: час виконання однієї ітерації циклу:

for (i=0; i<XXX; i++) {j+=1; j+=2; j+=3;}

Reflection method call (A):

Одна ітерація включає трикратне (для різних методів):

а) здобуття інформації про тип об'єкту;

б) здобуття інформації про метод об'єкту по його імені;

в) виклик методу по Invoke;

г) "накопичення" результату виклику.

Тест для PHP відсутній, оскільки будь-який виклик методу там йде "по імені".

Reflection method call (B):

Одна ітерація включає трикратне (для різних методів):

а) виклик методу по Invoke;

б) "накопичення" результату виклику.

Тест для PHP відсутній, оскільки будь-який виклик методу там йде "по імені".

Virtual method call:

Одна ітерація включає 3 виклики 1 віртуального методу (переобтяжений) для різних об'єктів і "накопичення" результату виклику.

C# обганяє PHP в ~ 500 разів.

Normal method call:

Одна ітерація включає 3 виклики 1 звичайного методу для різних об'єктів і "накопичення" результату виклику.

Схоже, що Native-code був сильно оптимізований, оскільки швидкість ~ 4,7 такту процесора на 1 ітерацію

Для PHP цей спосіб виклику нічим не відрізняється від попереднього.

C# обганяє PHP в ~ 3000 разів.

Static method call:

Одна ітерація включає 3 виклики 1 звичайного методу для різних об'єктів і "накопичення" результату виклику.

Мабуть, Native-code був оптимізований і виклик static-методов був замінений inline-кодом, оскільки швидкість ~ 2 такти процесора на 1 ітерацію (що збігається з Light-loop (unchecked)).

У PHP нічого відповідного не знайшлося (немає статичних методів).

Function call:

Одна ітерація включає 3 виклики 1 різних функцій і "накопичення" результату виклику. У C# функції відсутні, аналог - статичні методи. Швидкість дуже низька по порівнянню навіть із звичайними викликами методів в C# (у ~1000 раз).

String append (A):

Одна ітерація - додавання 1 символу до рядка:

S += "1";

У C# рядки - константи, тому реально при таких операціях просто змінюється заслання (тобто створюється новий об'єкт, в нього копіюється старий рядок + модифікації, і повертається вказання на нього). Результат - повільна робота таких ось конструкцій. Взагалі, в цій ситуації необхідно використовувати інший клас.NET Framework - StringBuilder.

String append (B):

Одна ітерація - додавання 10 символів до рядка:

S += "1234567890";

String builder (A):

Одна ітерація - додавання 1 символу до рядка в StringBuilder:

S.Append("1").

Швидкість дуже висока (у 10 разів швидше PHP).

String builder (B):

Одна ітерація - додавання 10 символів до рядка в StringBuilder:

S.Append("1234567890").

Швидкість досить висока (у 4 рази швидше PHP).

Class construction

Одна ітерація - створення простого об'єкту:

A[i]= new INNERA();

C# виграє в 40 разів.

Class construction & descruction:

Одна ітерація - створення і руйнування (1 раз в кінці роботи) простого об'єкту:

A[i]= new INNERA();

C#: Руйнування - видалення заслання на масив і виклик GC.Collect()

PHP: Руйнування - видалення заслання на масив

C# виграє в 40 разів.

Struct construction:

Створення масиву з об'єктів-структур:

A = new INNERA[Size];

Лише для C#

Struct construction & descruction

Створення масиву з об'єктів-структур, потім руйнування і збірка сміття:

A = new INNERA[Size];

Лише для C#

Ints:

Цілочисельні обчислення. Одна ітерація - виконання:

A += (A*i)%10;

C# виграє в 35 разів.

Floats

Дійсні обчислення. Одна ітерація - виконання:

A += A+(i/A*i);

C# виграє в 6,8 разів.

3.3.1 Висновки

У таблиці 3.3 відображено порівняльна характеристика часу виконання C# та PHP.

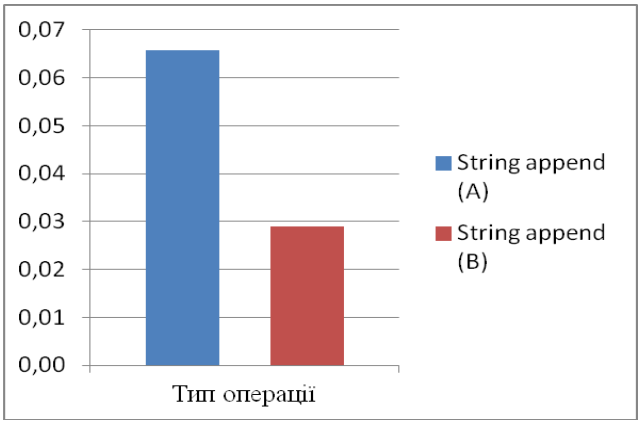
C# на більшості тестів показує більш, ніж 10-кратний виграш в продуктивності. При цьому в C# присутні всі можливості PHP (і навіть набагато більші).

Але для середніх і дрібних завдань різниця у продуктивності повністю знівелює протоколом HTTP веб-сервером.

Таблиця 3.3 - Порівняльна характеристика часу виконання C# та PHP

Назва операції	C#	PHP	t2/t1
	Час виконання, мС t1	Час виконання,мС t2	
Empty loop	0,0000039	0,0009041	231,8205128
Light loop	0,0000038	0,0020417	537,2894737
Virtual method call	0,0000287	0,0163283	568,9303136
Normal method call	0,0000059	0,0163913	2778,1864407
String append (A)	0,0250585	0,0016484	0,0657821
String append (B)	0,0638710	0,0018506	0,0289740
Class construction	0,0002689	0,0113949	42,3759762
Class constr. & destr.	0,0003585	0,0150252	41,9112971
Floats	0,0004300	0,0028833	6,7053488

На рисунку 3.1 відображено у скільки разів C# швидше за PHP у порівнянні з різними операціями.



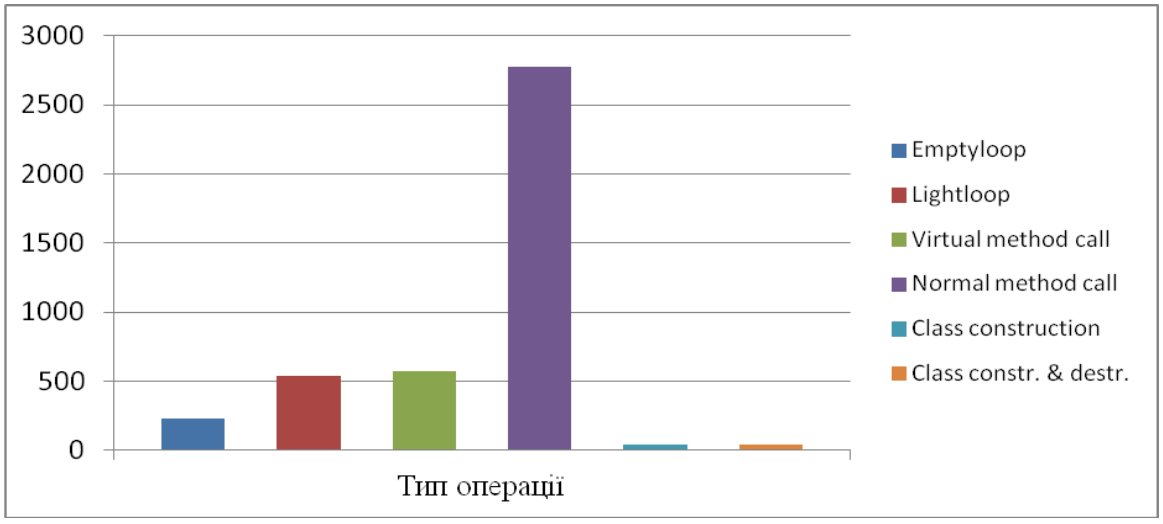


Рисунок 3.1 – У скільки разів швидкість роботи C# вища за PHP

РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СЕРВЕРУ РЕГІОНАЛЬНОЇ СИСТЕМИ ЕКОЛОГІЧНОГО МОНІТОРИНГУ ДЛЯ ПРОНОЗУВАННЯ ВИТРАТ ВІД ЗАБРУДНЕННЯ АТМОСФЕРНОГО ПОВІТРЯ

4.1 Проектування програмного забезпечення серверу системи екологічного моніторингу за допомогою мови UML

Як було сказано раніше система регіонального екологічного моніторингу знаходиться в розрахованій на багато користувачів середі Інтернет із спільним доступом.

Назвемо користувачів зовнішніми діями або акторами, а сервер системою. Тоді зовнішні дії на систему можна класифікувати як показано на рисунку 4.1.

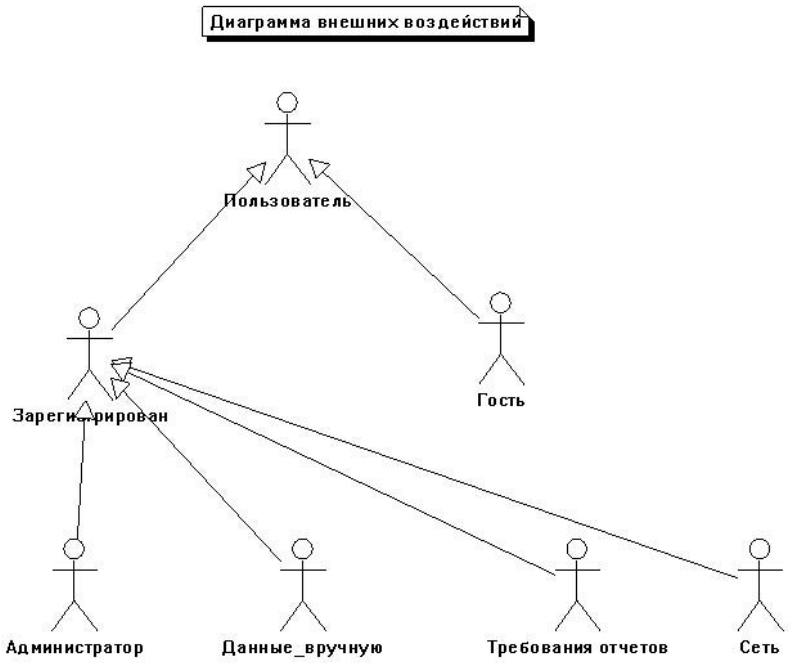


Рисунок 4.1 - Діаграма акторів

Кожна зовнішня дія на систему викликатиме “у відповідь реакцію” системи відповідно до того, до якого вигляду ця дія відноситься. Для відображення взаємодії системи із зовнішніми діями скористаємося діаграмами варіантів використання мови UML (рис. 4.2, 4.3).

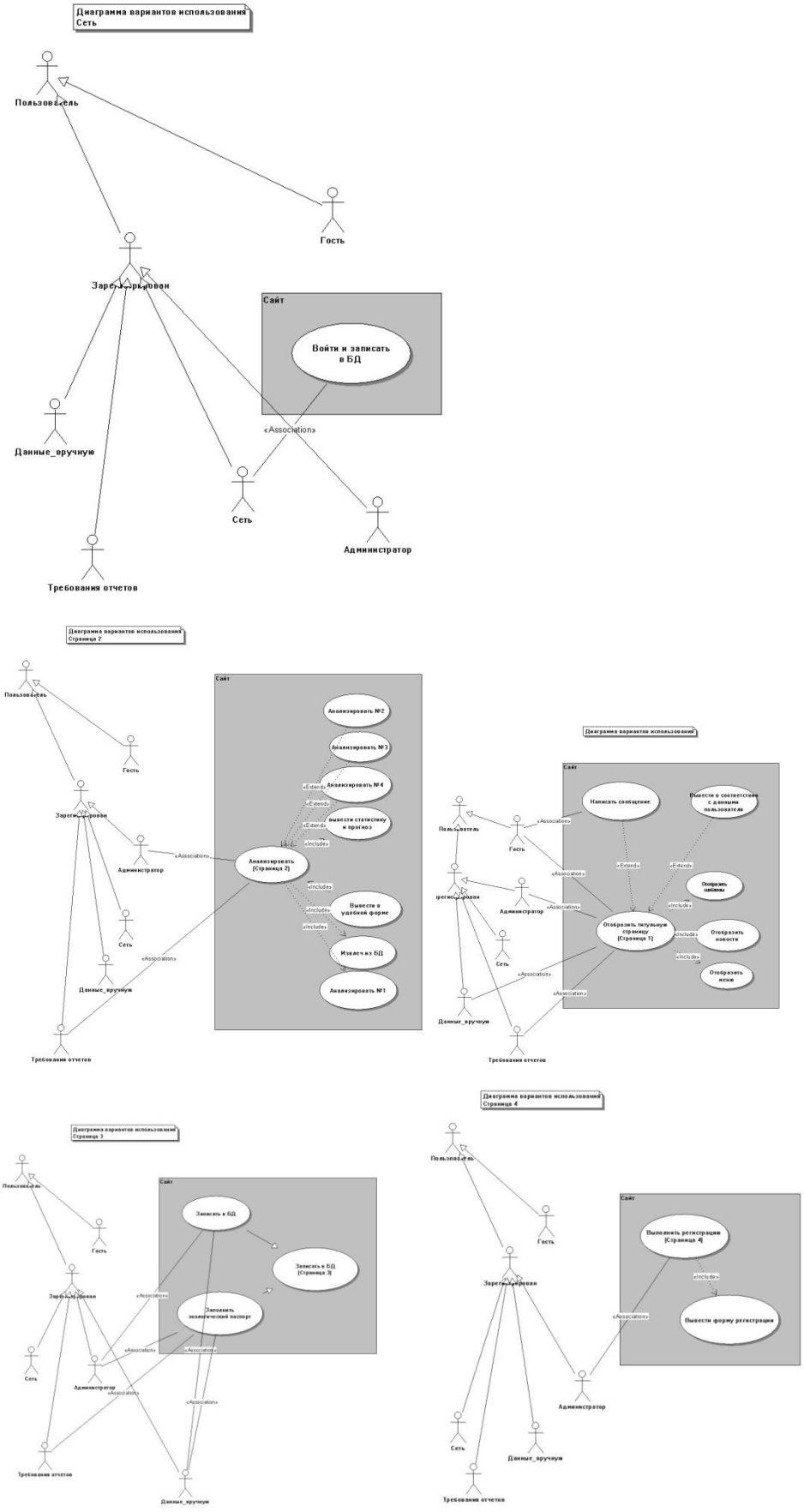


Рисунок 4.2 - Діаграма варіанту використання

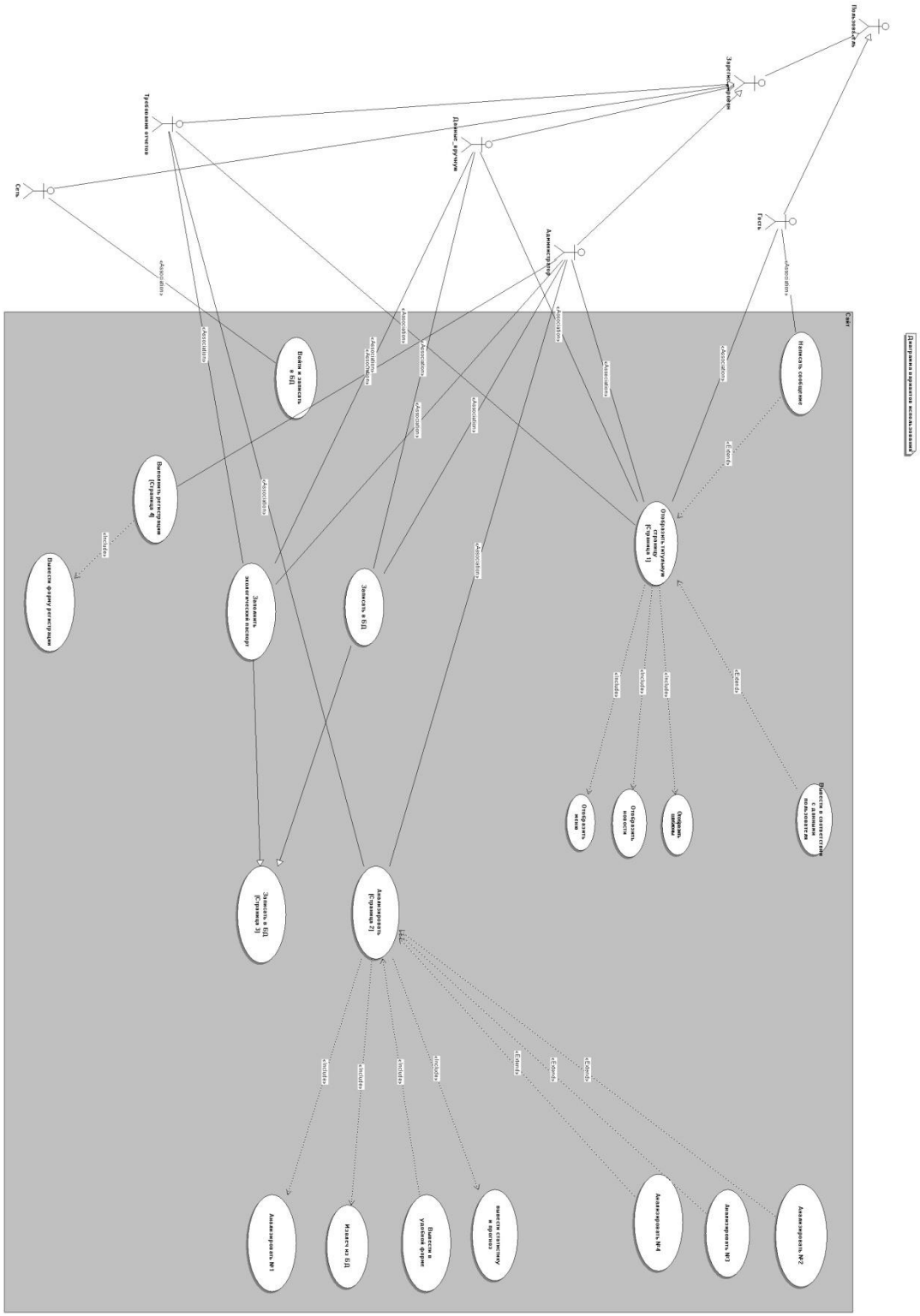


Рисунок 4.3 - Діаграма варіанту використання загального вигляду

На діаграмі варіантів використання (рис.4.2,4.3) відображено, що кожен актор має обмежені права доступу до ресурсів системи. Наприклад, актор-дані уручну може окрім входу на титульну сторінку, записувати дані в базу даних і заповнювати екологічний

паспорт, а вимога звітів -заповнювати або коректувати екологічний паспорт і аналізувати дані. Актор-мережа має лише доступ до бази даних.

Окрім прав доступу також видно структуру компонентів, що активізуються акторами системи. Наприклад, титульна сторінка включає шаблон, новини, меню, а також можливість додаткового підключення, виведення даних відповідно до даних користувача.

На рисунку 4.4 відображено діаграму компонентів програмного забезпечення, де відображено, що основу роботи складають чотири перенавантажуванні сторінки index.php,logic.php,workBD.php,registration.php. Окремо відображує модулі, що підключаються, при роботі сервера. Сторінка logic.php підключає спільний модуль статистичного аналізу analization.php і набір спеціалізованих модулів analizatio1.php,analization2.php, кожен з яких відповідає за обробку спеціалізованої моніторингової інформації, що відноситься до певного типу носія.

Діаграма компонентів дає спільне уявлення про майбутню структуру програмного забезпечення і можливість виділити основні розділи і блоки, які стануть основою розробки.

На рисунку 4.5 відображено діаграму стану для користувача, що намагається увійти до системи без використання форми реєстрації.

На рисунку 4.6 відображено діаграму архітектури, що відображує взємодію технологій, на стороні клієнта та серверу при ручному занесенні моніторингової інформації.

На рисунку 4.7 відображено діаграму розгортання, що відображує майбутнє розміщення програмного забезпечення та структуру, в якій знаходитиметься вихідний код.

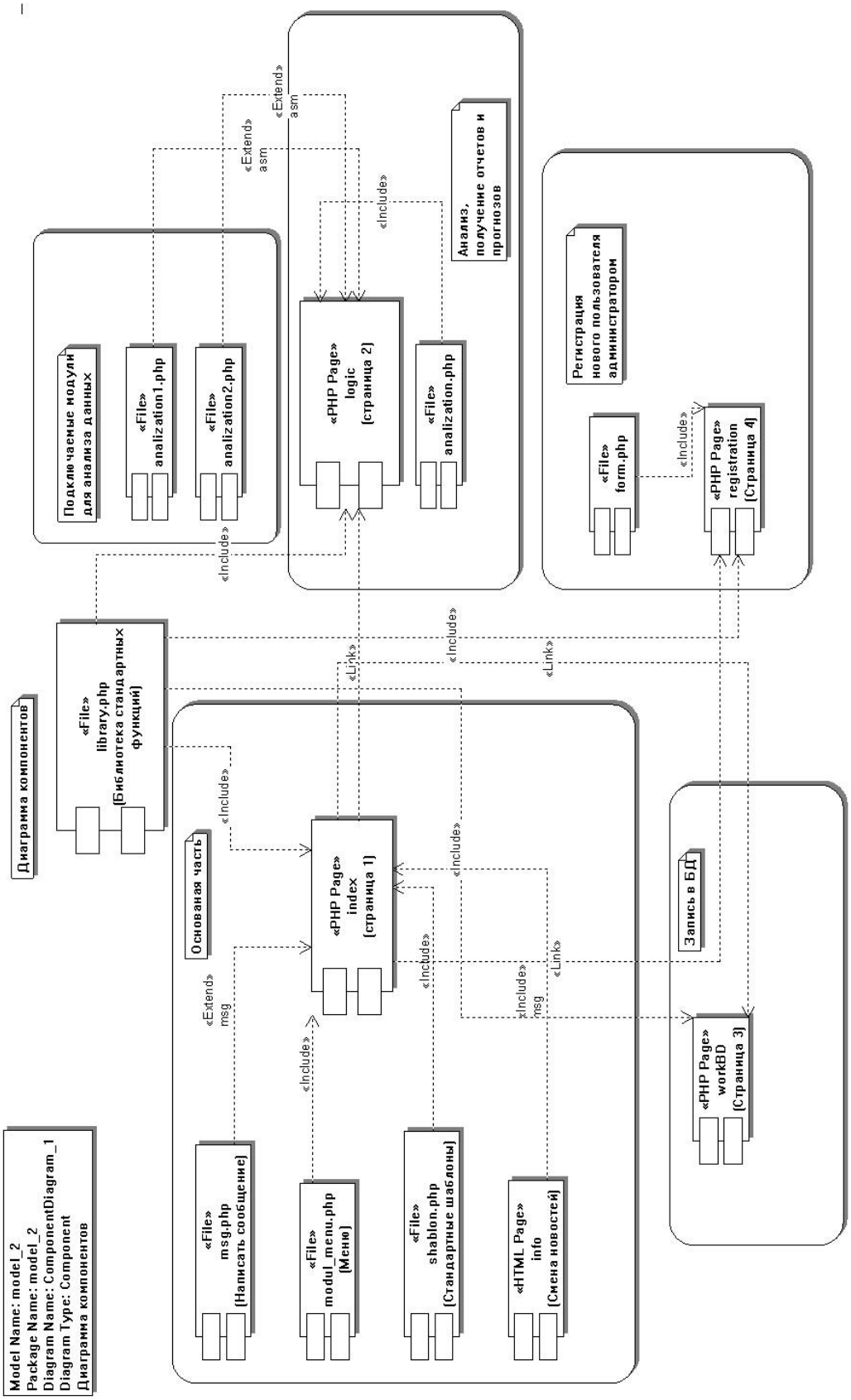


Рисунок 4.4 - Діаграма компонентів

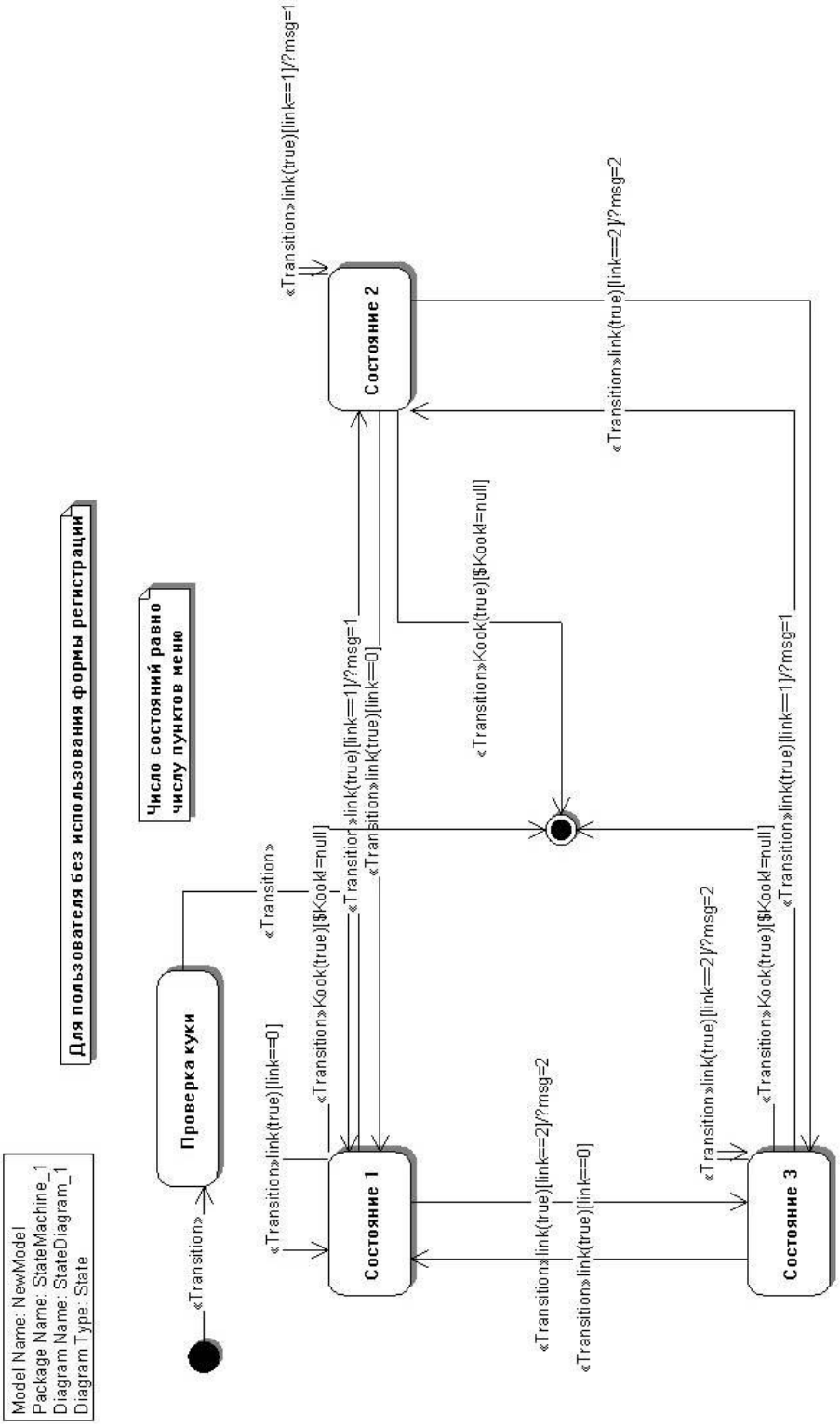


Рисунок 4.5 - Діаграма стану

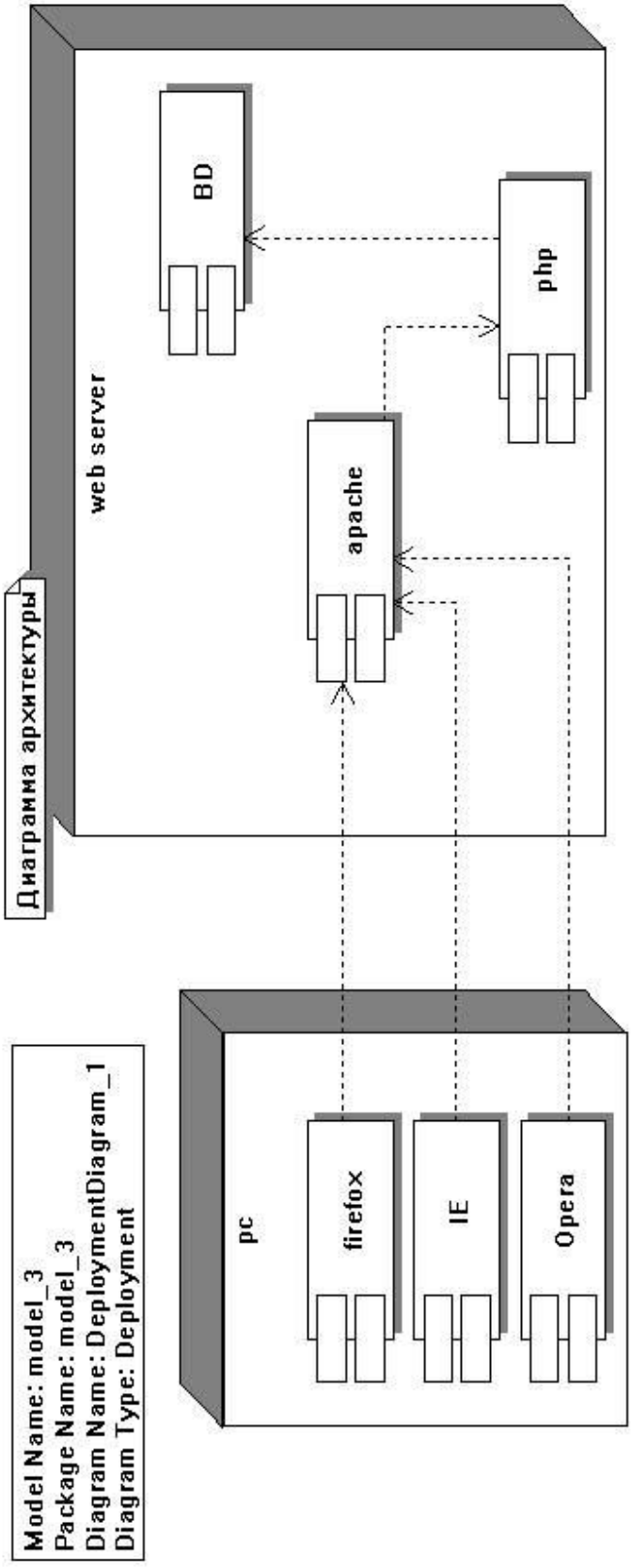


Рисунок 4.6 - Діаграма архітектури

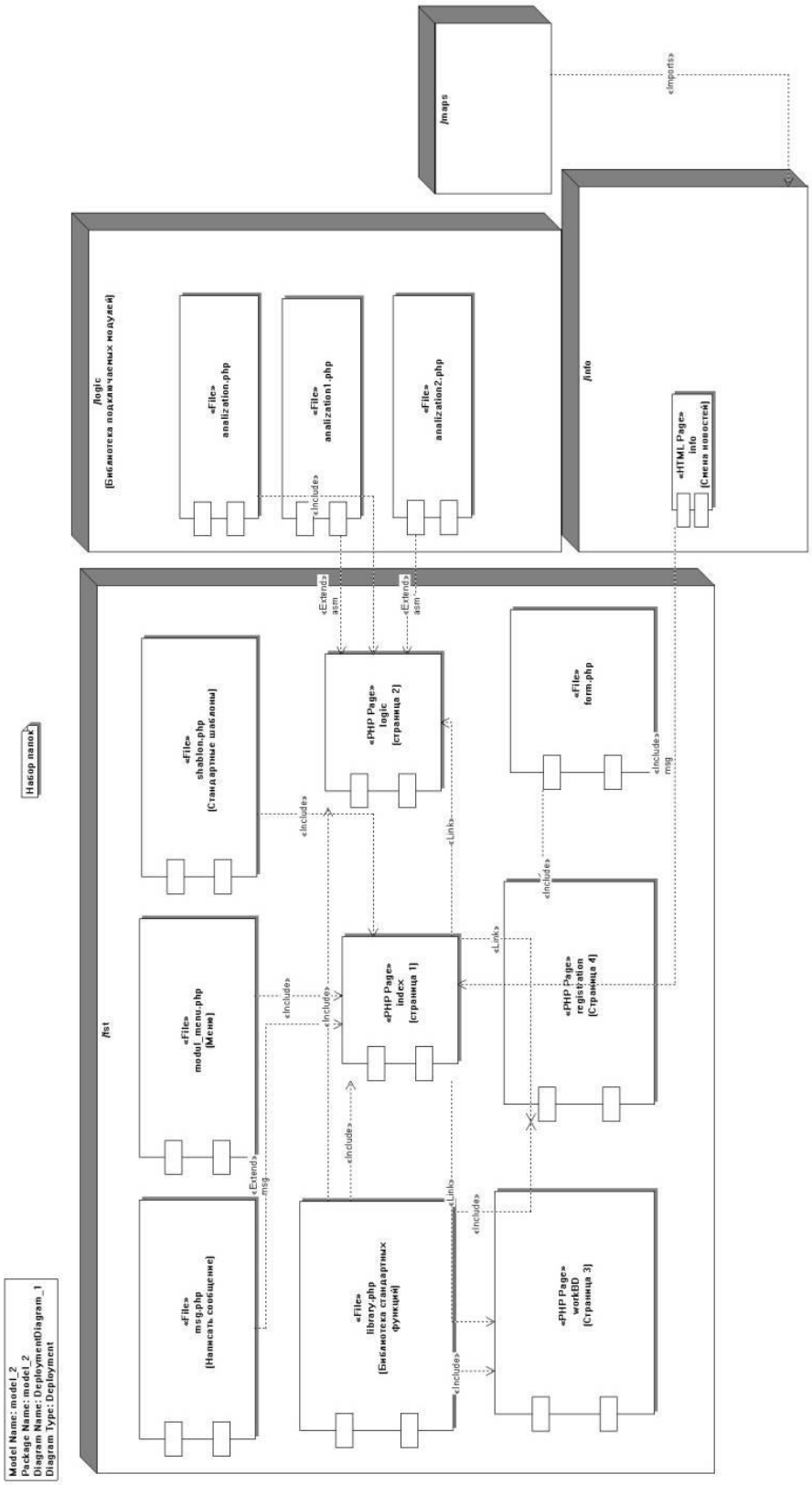


Рисунок 4.7 - Діаграма розгортання

4.2 Розробка бази даних

У загальному вигляді базу даних для зберігання інформації про носія інформації, екологічний паспорт, моніторингову інформацію, полів memo, можна представити у формі двох таблиць (рисунок 4.8). У одні поля заноситься назва параметра або характеристики, в інших їх пояснення або значення.

Рисунок 4.8 - База даних зберігання інформації у загальному вигляді

Переваги - відносна простота зберігання інформації і реалізації.

Недоліки - вся інформація зібрана в одну таблицю. При приватному оновленні і довгому зберіганні кількість записів в ній буде настільки великою, що час вибірки може бути істотним.

Усунути даний недолік можливо, якщо кожному постачальникові інформації виділити свою базу даних. Таким чином, згідно завдання, маємо 13 постачальників (носіїв) інформації, отже 13 баз даних по кожному. При цьому забезпечується можливість наявності декількох носіїв заданого типа інформації, оновлюючих виділену ним базу даних моніторингової інформації. Отримані 13 баз даних назвемо спеціалізованим базами даних.

Проте необхідно забезпечити можливість реєстрації та доступу до ресурсів всіх останніх користувачів, які не є постачальниками (носіями) моніторинговій інформації, і користувачів, що мають необмежений доступ до всіх ресурсів (адміністраторам), -

створюється ще дві бази даних - база даних користувачів сайту, база даних адміністратора.

На рисунку 4.9 відображено структуру такої бази даних.

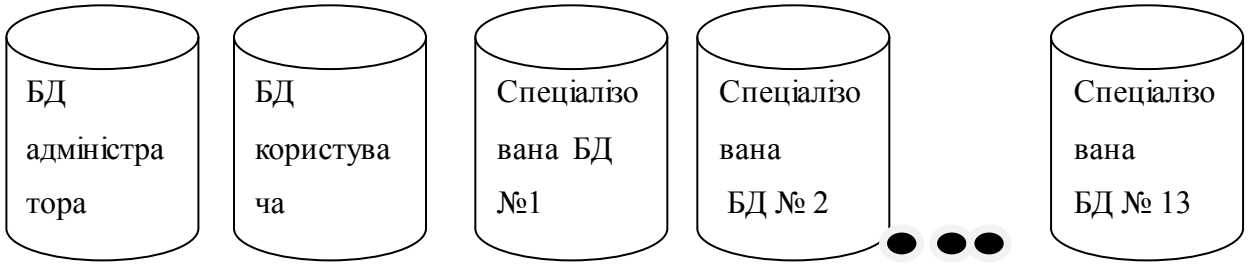


Рисунок 4.9 - Структура бази даних програмного забезпечення серверу регіональної системи екологічного моніторингу

Переваги:

а) відносна простота реалізації, менший об'єм записів для зберігання в одній таблиці, швидша вибірка, підвищення надійності інформації, що зберігається;

б) вищий рівень захищеності від злому;

в) чіткіше розділення прав доступу користувачів;

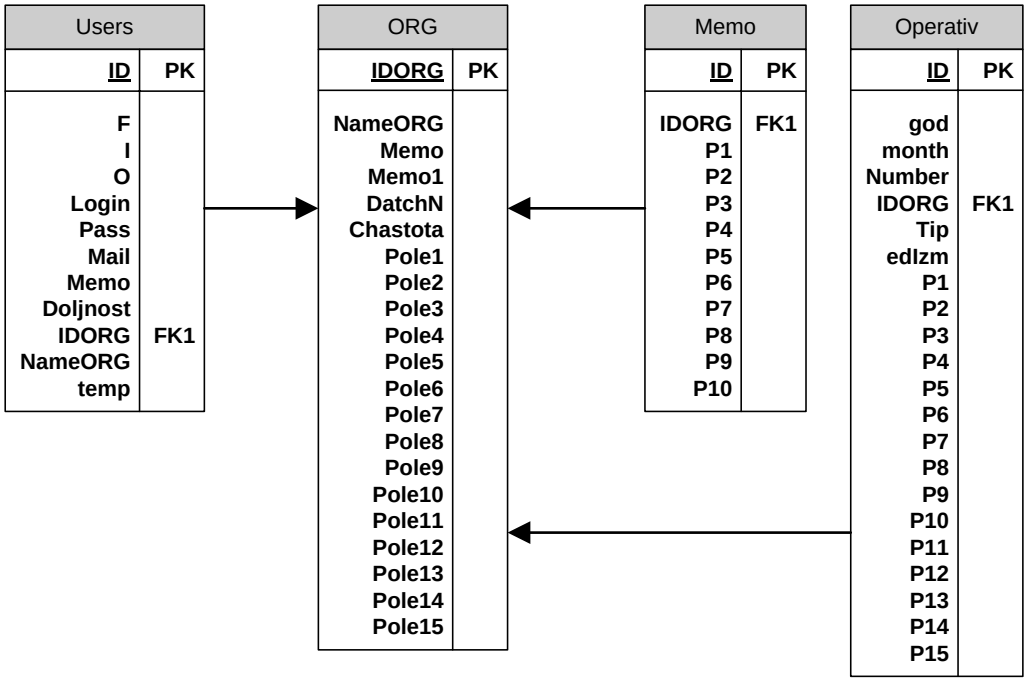
г) передбачається можливість декільком користувачам мати доступ до занесення оперативної інформації, а також можливість наявності декількох постачальників.

Недолік - при витяганні всієї моніторингової інформації для аналізу і обробки доводиться здійснювати окремо звернення до кожної бази даних.

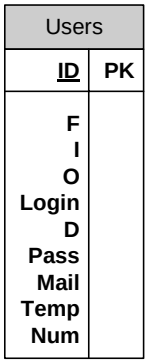
Структура спеціалізованої бази даних у загальному вигляді відображена на рисунку 4.10.

Рисунок 4.10 - Структура спеціалізованої бази даних у загальному вигляді

Структура спеціалізованої бази даних відображена на рисунку 4.11.



Структура БД пользователей сайта



Структура БД администратора

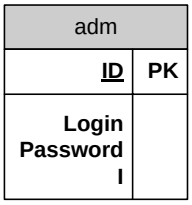


Рисунок 4.11 - Структура спеціалізованої бази даних

Типи полів даних, що використано у базах даних, відображено на рисунках 4.12,4.13.

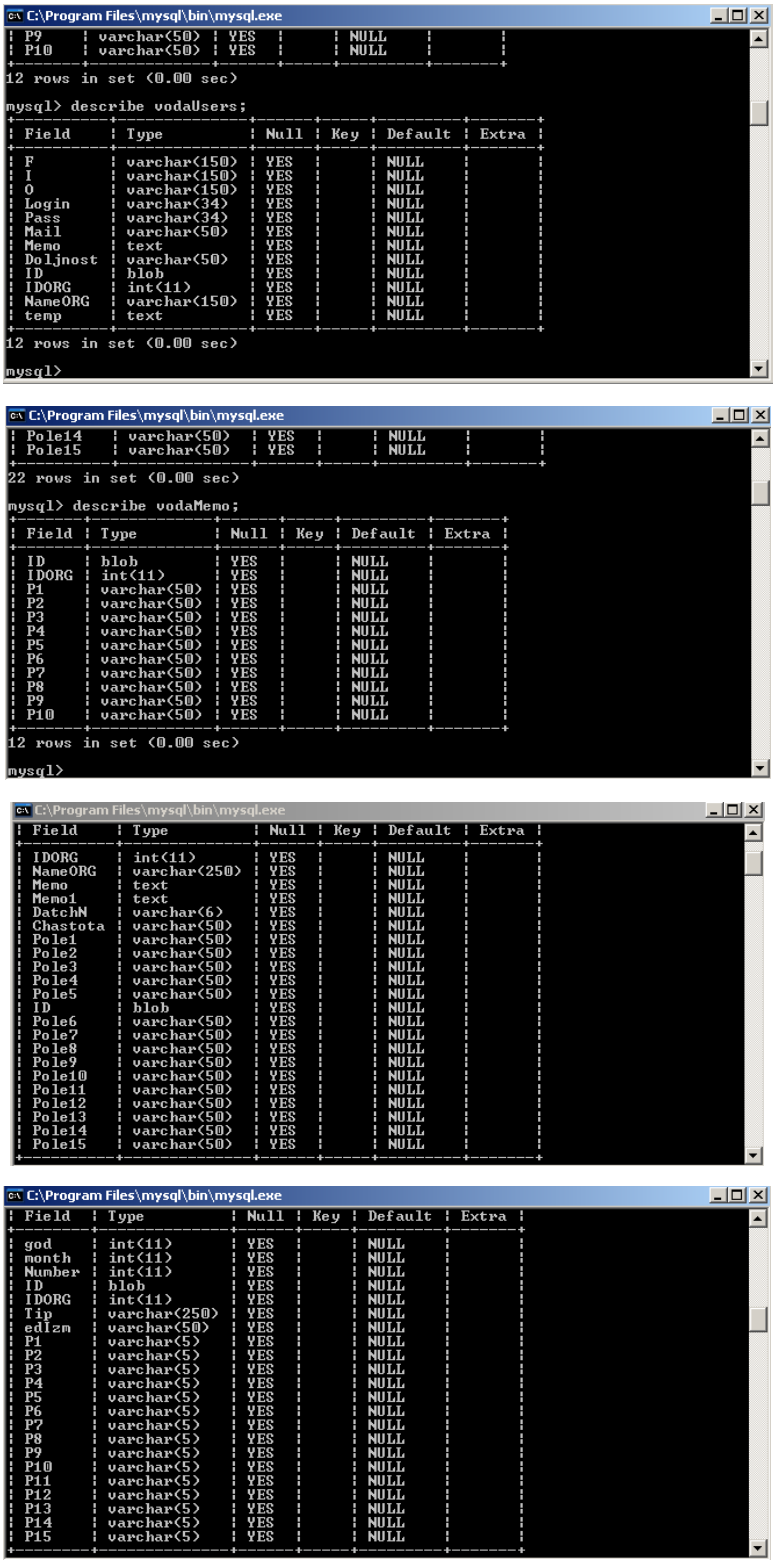


Рисунок 4.12 – Типи полів даних спеціалізованої бази даних

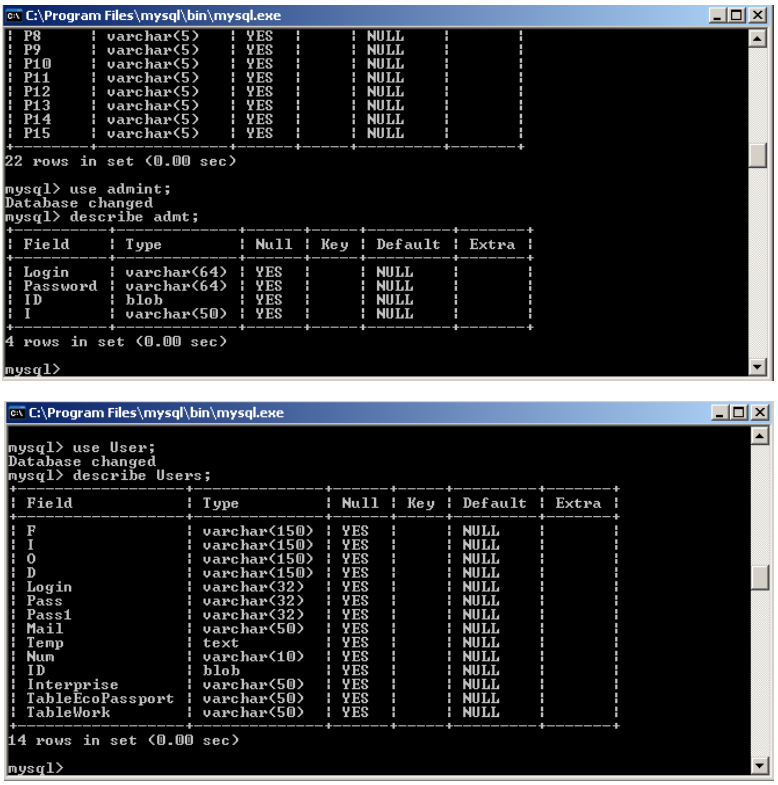


Рисунок 4.13 – Типи полів даних загальної бази даних

4.3 Розробка програмного забезпечення на основі технологій ASP.NET, IIS, C#,XML,SQL2000,AJAX,CSS,UML

Спочатку виводяться моніторингові дані за квартал про характеристики повітря м. Сєвєродонецьк. Після цього проводиться автоматичний розрахунок і будується графік у вигляді стовпчастих діаграм за вихідними даними.

Інформація відображується на сторінці Default.aspx.

Клас _Default : System.Web.UI.Page спадкоємець від класу Page. Здійснює вивід і розрахунок інформації.

Методи класу _Default:

- public void OUTTB:

а) Кількість аргументів: 1.

б) Мова: C#,XML.

в) Призначення: Виведення таблиць вихідних даних.

г) Опис: Double[] INF21 - динамічний масив вихідних даних для виводу.

- private void Stat:

а) Кількість аргументів: 0.

б) Мова: C#,XML.

в) Призначення: Виведення таблиці розрахункових даних.

г) Опис: Виводить в окрему таблицю розрахункові дані отримані згідно з вихідними даними моніторингової інформиації за 1 квартал.

- private void OutGraph:

а) Кількість аргументів: 0.

б) Мова: C#,XML.

в) Призначення: будує графік у вигляді стовпчастих діаграм.

г) Опис: графік будується виходячи з розрахункових даних моніторингової інформації.

Події класу _Default:

- protected void Page_Load. Виникає після відправки даних користувачеві у вигляді рядка коди XML,HTML,javaSCRIPT,C#.
- protected void REZ_Click. Виникає після події Page_Load при натисненні на кнопку ”Розрахувати” у формі.

Лістинг програми відображено у додатку Б.

На рисунку 4.14,4.15,4.16 відображено зовнішній вигляд розробленого програмного забезпечення.

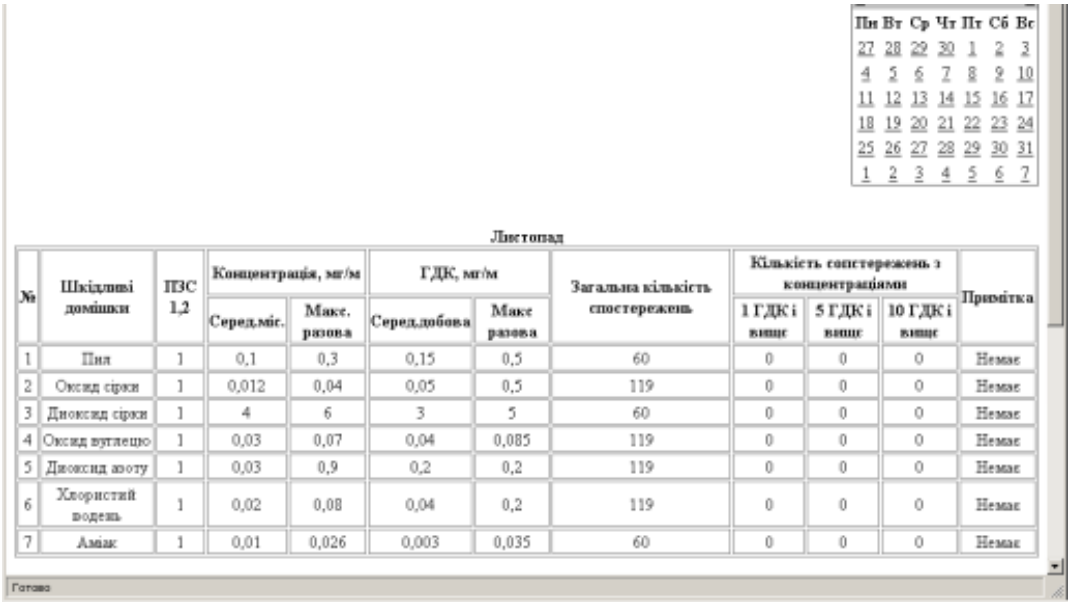
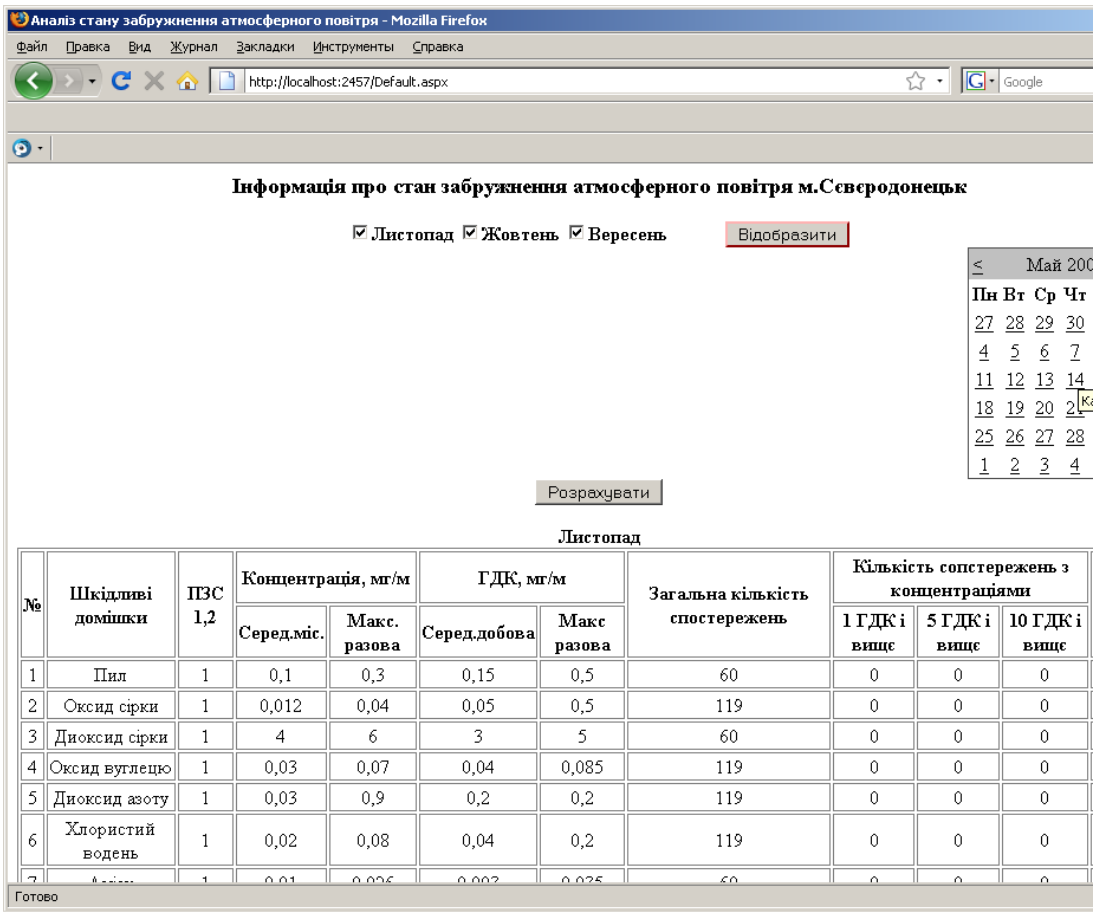


Рисунок 4.14 –Загальний вигляд



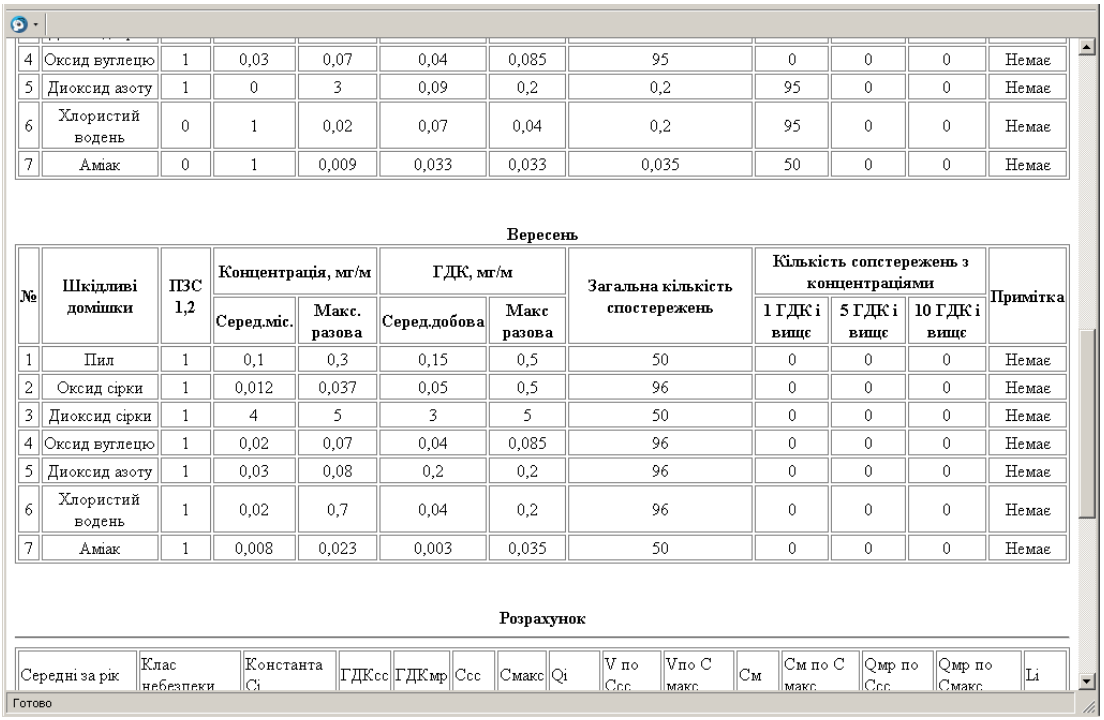
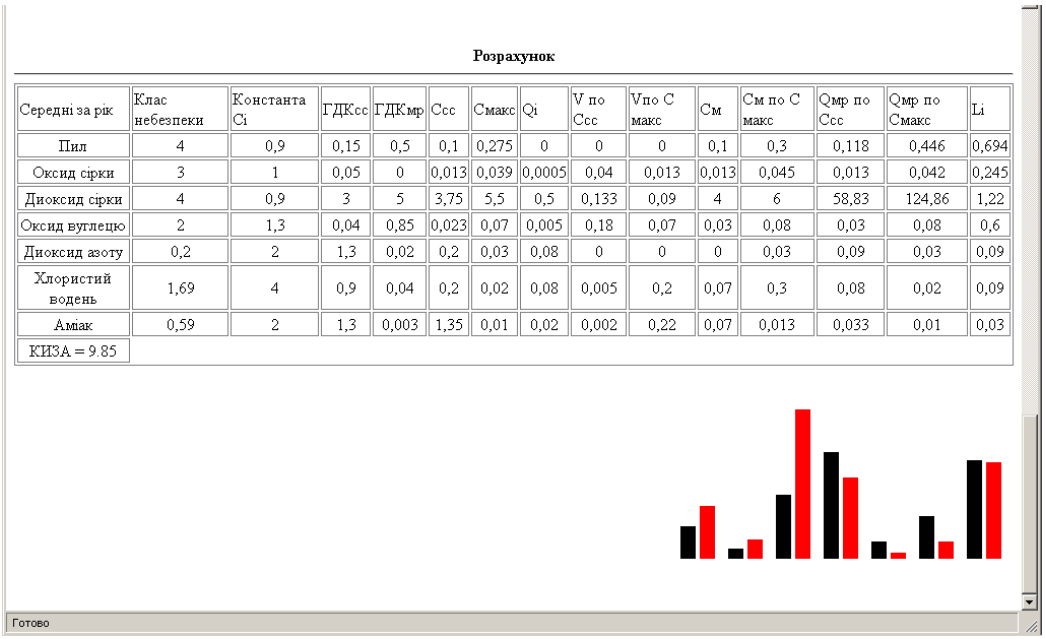


Рисунок 4.15 – Вихідні дані для розрахунку за квартал



Розрахунок

Середні за рік	Клас небезпеки	Константа Ci	ГДКсс	ГДКмр	Ссс	Смакс	Qi	V по Ссс	Vпо С макс	См	См по С макс	Qмр по Ссс	Qмр по Смакс	Li
Пил	4	0,9	0,15	0,5	0,1	0,275	0	0	0	0,1	0,3	0,118	0,446	0,694
Оксид сірки	3	1	0,05	0	0,013	0,039	0,0005	0,04	0,013	0,013	0,045	0,013	0,042	0,245
Диоксид сірки	4	0,9	3	5	3,75	5,5	0,5	0,133	0,09	4	6	58,83	124,86	1,22
Оксид вуглецю	2	1,3	0,04	0,85	0,023	0,07	0,005	0,18	0,07	0,03	0,08	0,03	0,08	0,6
Диоксид азоту	0,2	2	1,3	0,02	0,2	0,03	0,08	0	0	0	0,03	0,09	0,03	0,09
Хлористий водень	1,69	4	0,9	0,04	0,2	0,02	0,08	0,005	0,2	0,07	0,3	0,08	0,02	0,09
Аміак	0,59	2	1,3	0,003	1,35	0,01	0,02	0,002	0,22	0,07	0,013	0,033	0,01	0,03
КИЗА = 9.85														

Готово

Рисунок 4.16 – Розраховані дані та стовбчата діаграма

4.4 Розробка програмного забезпечення на основі технологій PHP,XML, HTML, MYSQL, AHACHE, AJAX, JAVAScript, CSS,UML

Основні сторінки:

а) index.php - відображення титульної сторінки сайту, а також стрічки новин і інформації для спільного доступу;

б) admin.php - реєстрація користувачів, що мають необмежений доступ до ресурсів системи;

в) income.php - відображення вікна вітання користувача, що знов увійшов до системи;

г) logic.php - розрахунок і відображення моніторингової інформації для обробки;

д) registered.php - вікно інформації про реєстрацію нового користувача;

е) registratiuonNewUser.php - реєстрація користувача, який є носієм інформації;

ж) workBD.php - занесення моніторингової інформації носієм информции уручну.

Основні модулі:

а) anakization.php - спільний і статистичний аналіз моніторингової інформації;

б) analization1.php-analization13.php - спеціалізовані модулі, що підключаються, для обробки моніторингової інформиації заданого типа (вода, повітря);

в) stor-stor7 - модулі новини вказаного розділу (вода, повітря, законодавчі акти, конференції та ін.);

г) trace.htm,map.htm - карти міста і доріг міста;

д) autorization,autorization1-autorization14 - модулі перевірки наявності даних про реєстрацію користувача в кожній базі даних з витяганням і розділенням прав доступу залежно від його приналежності до вказаної базі даних. Один користувач може бути водночас зареєстрований лише в одній базі даних;

е) library.php - бібліотека стандартних функцій;

ж) shablon.php - оформлення і дизайн сайту;

з) wz_jsgraphics.js - модуль на мові javaScript для малювання графічних примітивів.

Опис функцій:

- Leave_Data_Delete:

а) Кількість аргументів: від 8.

б) Мова: javaScript.

в) Призначення: додавання, видалення, пропуск аргументів в потоці виведення Url.

г) Опис: 0- режиму 0 - виконання і перехід на отриманий URL. 1- формування URL без переходу. 1-режим: 0- читати з URL 1- читати із заданого URL з наступного php. 2 або більше - читати URL з 2 аргументу. 2- URL з Java або ім'ям об'єкту контейнера, де знаходиться. 3- закладки 0 ~ -нема 4- режиму 0,1,2,3,4. 0,1- додати змінну. 0- видалити. 1- залишити. 2,3 -не додавати змінну. 5- значення змінної, що виводиться. 6 –ім’я змінної, що виводиться. 7 - список змінних.

- CREATEBUTTON:

а) Кількість аргументів: 11.

б) Мова: Html,javaScript,php.

в) Призначення: формування і виведення кнопки.

г) Опис: $i1,$i2,$i3 - URL або серверна адреса піктограми умовчання, наведення, натиснення миші на кнопці. $NAME,$NameBut - Ім'я параметра, що передається методом GET, і його значення. Відключення '~'. $namebutton - унікальний індекс створюваної кнопки. $hreff - значення цього параметра залежить від вибраного типа створюваної кнопки. Якщо створюється тип 3- вказується ім'я підпрограми обробки,

в останньому випадку URL адреса абсолютного переходу. Відключення '~'. $Raf- закладка, вказується без '#'. Відключення '~'. $title- спливаюча підказка. Відключення '~'. $MODE_URL - тип створюваної кнопки може набувати 0-3 значення: при натисненні перехід на вказану адресу URL по вмісту $hreff. Перехід на поточну адресу сторінки з додаванням параметра, що передається. Перехід на поточну адресу сторінки з видаленням параметра, що передається.

- SQLWork:

а) Число аргументів: 5.

б) Мова: php.

в) Призначення: формування запитів до БД.

г) Опис:$s-host,$s1-login,$s2-password,$s3-database,$s4-SQL query.

- Клас inf:

а) Число аргументів: 10.

б) Мова: Html,javaScript,php.

в) Призначення: Виведення блоку інформації заданого числа знаків і стилю.

Опис: $a2- розмір шрифту рядка заголовка. $a3- залежить від значення $a6 у спільному випадку число символів, що відображуються, в тексті. $a4- колір рядка заголовка. Відключення 'NOcolor'. $a5- - залежить від значення $a6 у спільному випадку число символів в заголовку. $a6- режим відображення, може приймати одне з значень (0- виводиться лише заголовок як гіперпосилання. 1-відображення частини символів вказаних вище, останні доповнюються '.' 2 і будь-яке другое- виводиться лише заголовок. 3-відображення всієї інформації. $a7- розрив рядків. Відключення 'NOT'. $a8- інформація, що виводиться. $a9- заголовок. $a10- залежить від значення $a6 у спільному випадку гіперпосилання для $a6. Без закладки. $a11- вставка закладки відключення '~'.)

Приклад створення:

```
$INF= new inf();
$INF-> Init($a2,$a3,$a4,$a5,$a6,$a7,$a8,$a9,$a10,$a11);
$INF->show();
```

- DrowGraphLine:

а) Кількість аргументів: 12.

б) Мова: Html,php,javaScript.

в) Призначення: будує графік за вказаними даними.

г) Опис: $a1,$a2,$a3,$a4 - координати. $a5- 0 -вивести початок створення графіка, 1- завершити створення графіка, 2- вивести графік і завершити. $a6- колір лінії.

$a7 - товщина. $a9 - ID об'єкту, в якому виводиться графік (HTML). $a10,$a11 - відносне позиціювання. $a12 - ID графіку.

- Graph($MAS,$L,$MAS1):

а) Кількість аргументів: 3.

б) Мова: HTML, XML, php.

в) Призначення: будує графік у вигляді стовпчастих діаграм.

г) Опис: $MAS - двовимірний масив даних. $L- підпис.$MAS1 - підписи даних.

- OutRecords:

а) Кількість аргументів: 0.

б) Мова: HTML, php.

в) Призначення: виводить таблицю записів після вибірки.

г) Опис: після вибірки коли заповнені глобальні масиви даних $SUM_R; $MAX_R; $MIN_R; $DATAREQUEST; виводить вміст таблиці Memo вибраної бази даних.

- OutTableForGraph:

а) Кількість аргументів: 0.

б) Мова: HTML, php.

в) Призначення: виводить форму-фільтр для побудови графіка у вигляді ламаної лінії.

г) Опис: виконується одночасно з фільтром вибірки записів з бази даних моніторингової інформації. Працює спільно з функцією GetGraph($First,$Number).

- GetGraph:

а) Кількість аргументів: 2.

б) Мова: HTML, javaScript, php.

в) Призначення: Виводить графік у вигляді ламаної лінії.

г) Опис: $First - з якого аргументу в базі даних моніторингової інформації побудувати графік за даними вибірки. $Number- по який аргумент в базі даних моніторингової інформації побудувати графік за даними вибірки.

- SELRecords:

а) Кількість аргументів: 0.

б) Мова: HTML, javaScript, php.

в) Призначення: Вивід даних з бази даних.

г) Опис: Форма-фільтр вибірки даних з бази даних.

Лістинг програми відображено у додатку Б.

Загальна характеристика програмного забезпечення:

- Версія БД - mySql 1.4.
- База даних - vodaT,Users, AdminT.
- Ім'я таблиці в БД - vodaUser,vodaOrg,vodaMemo,vodaOperativ,User,admT.
- Типів полів БД - integer,varchar,blob,text.
- Доступ до БД - localhost,root,password.
- Версія сервера http - apache 2.0.
- Середа розробки - VisualStudio2008, MSDN2008, VisualUML 5.3 Plus.

На рисунку 4.17-4.25 відображено зовнішній вигляд розробленого програмного забезпечення.

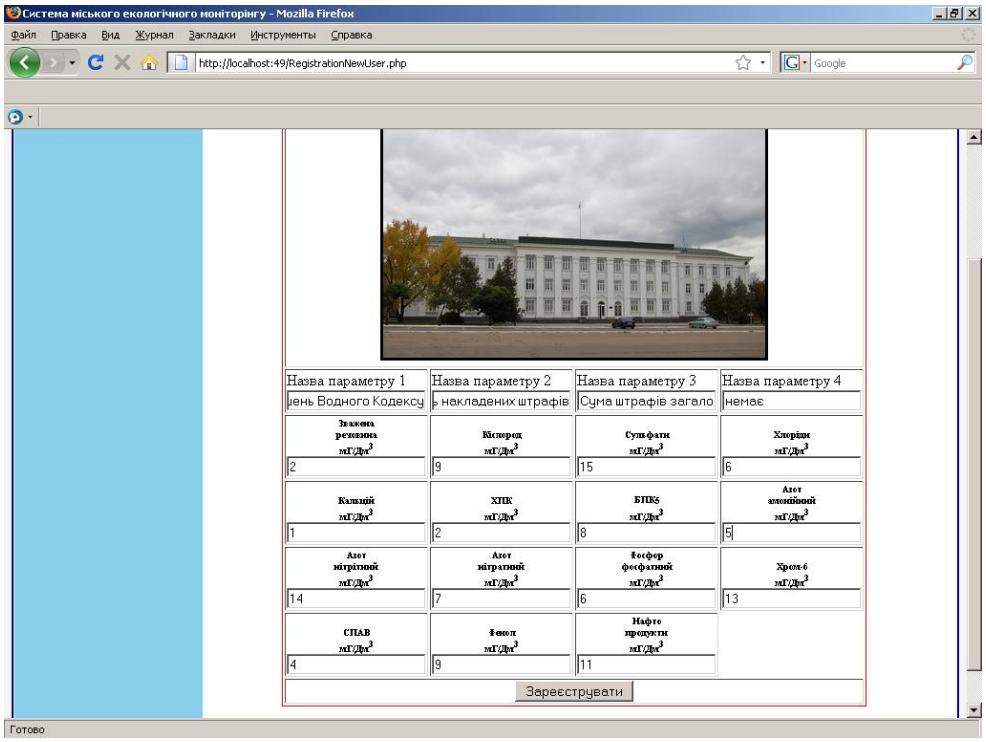


Рисунок 4.17 – Заповнення форми інформації щодо реєстрації користувача спеціалізованої бази даних

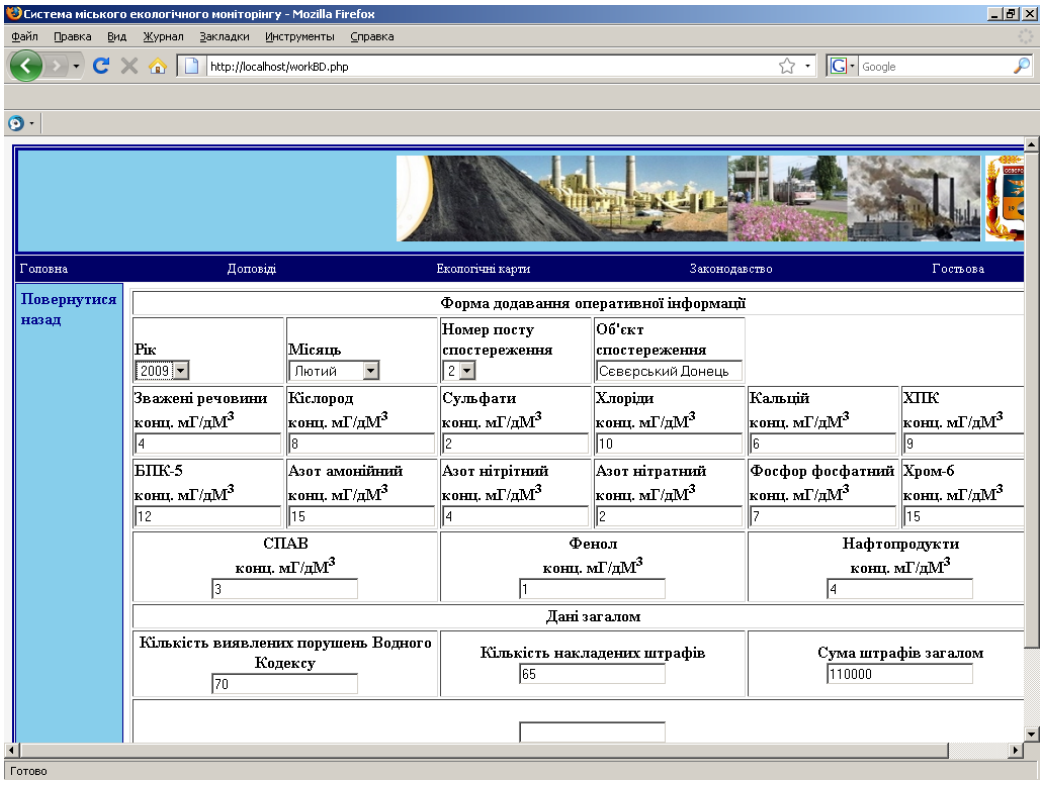


Рисунок 4.18 –Заповнення бази даних моніторинговою інформацією ручним методом та форма реєстрації користувача загальної бази даних (користувача)

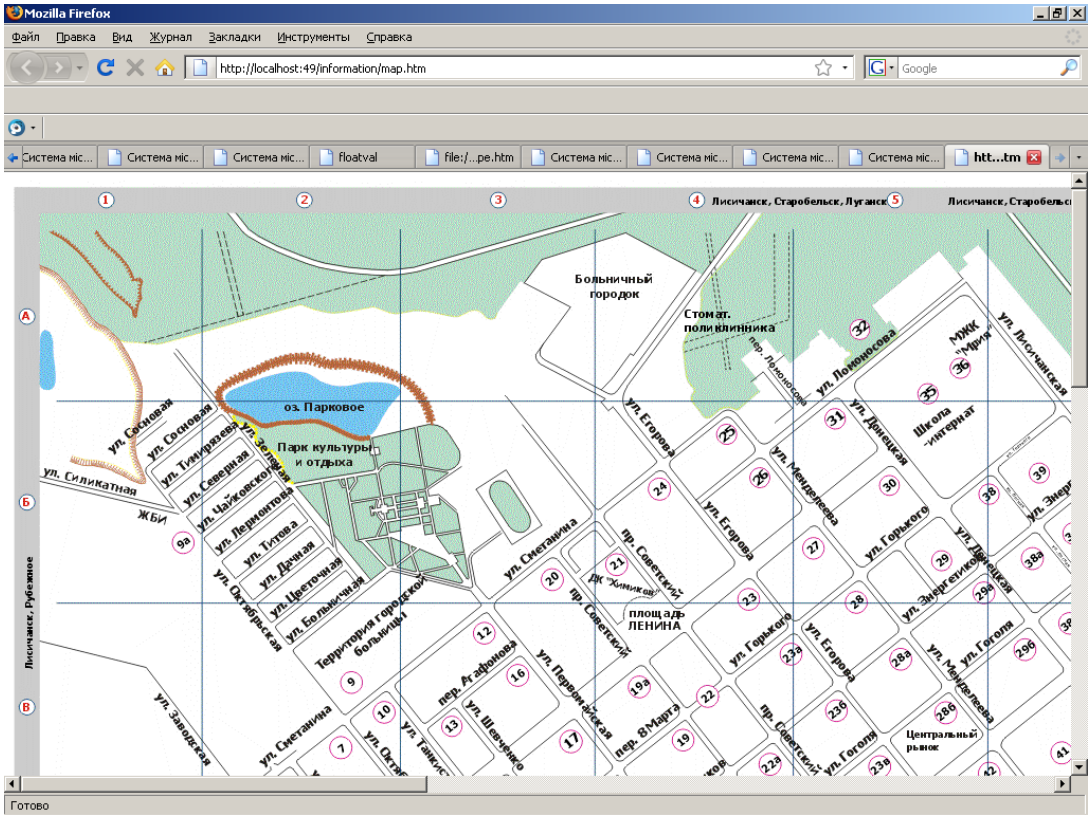


Рисунок 4.19 – Карта міста

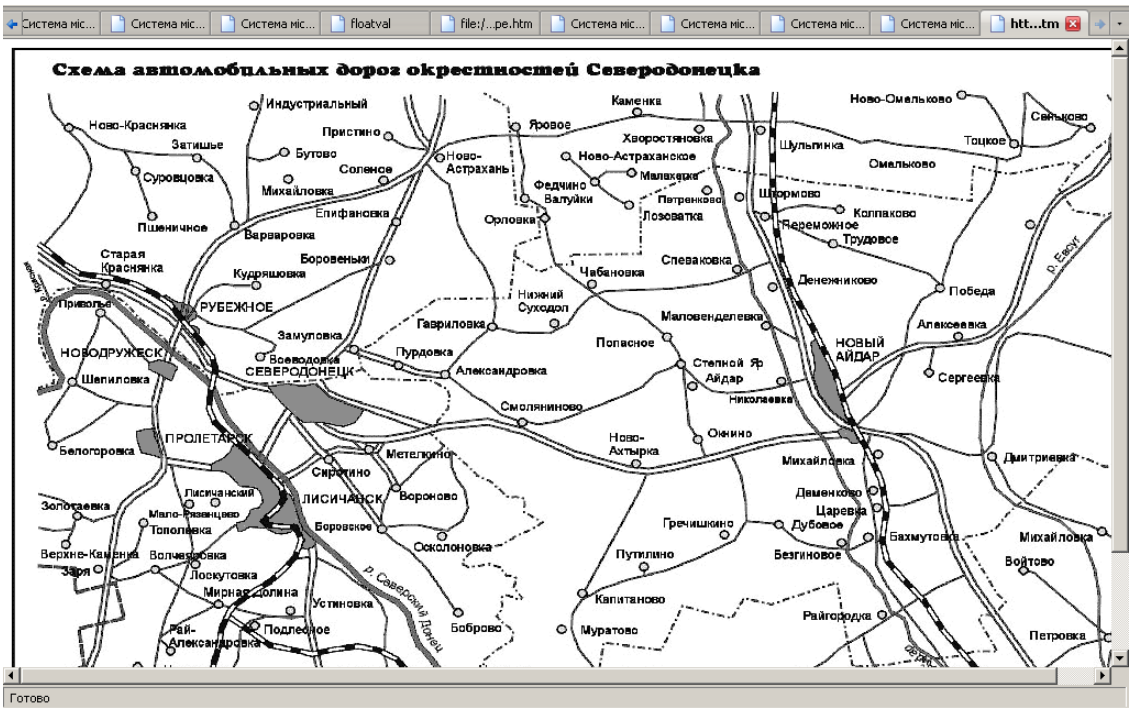


Рисунок 4.20 – Карта доріг міста

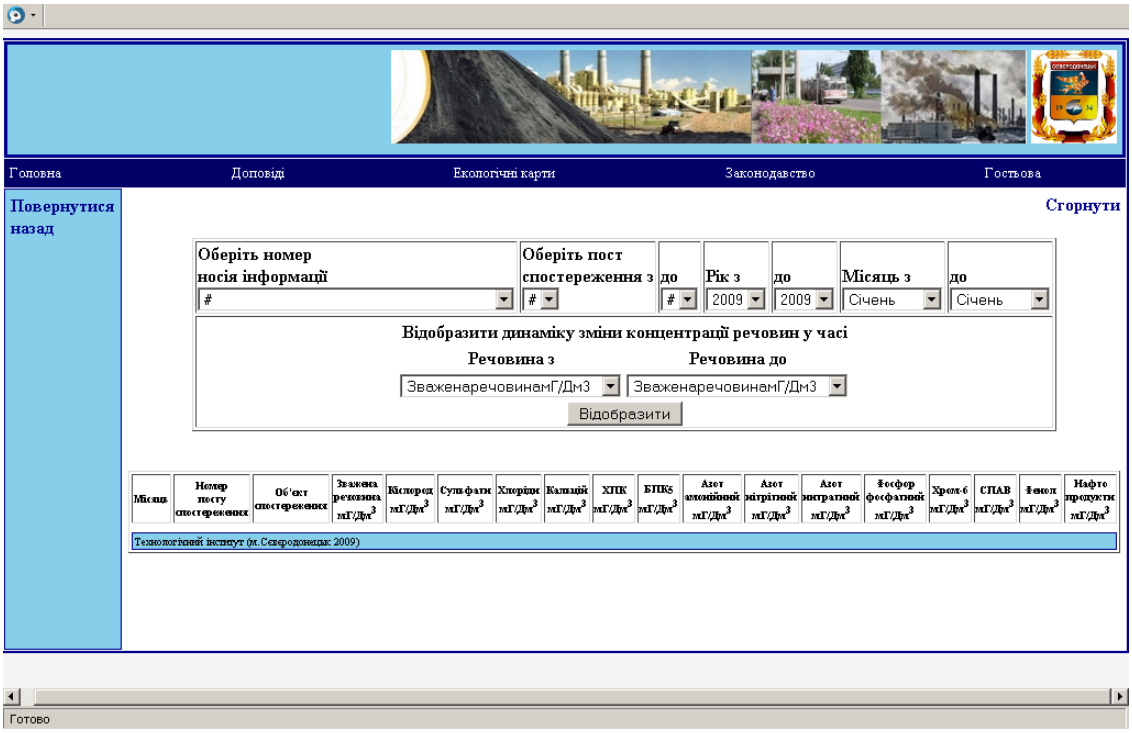
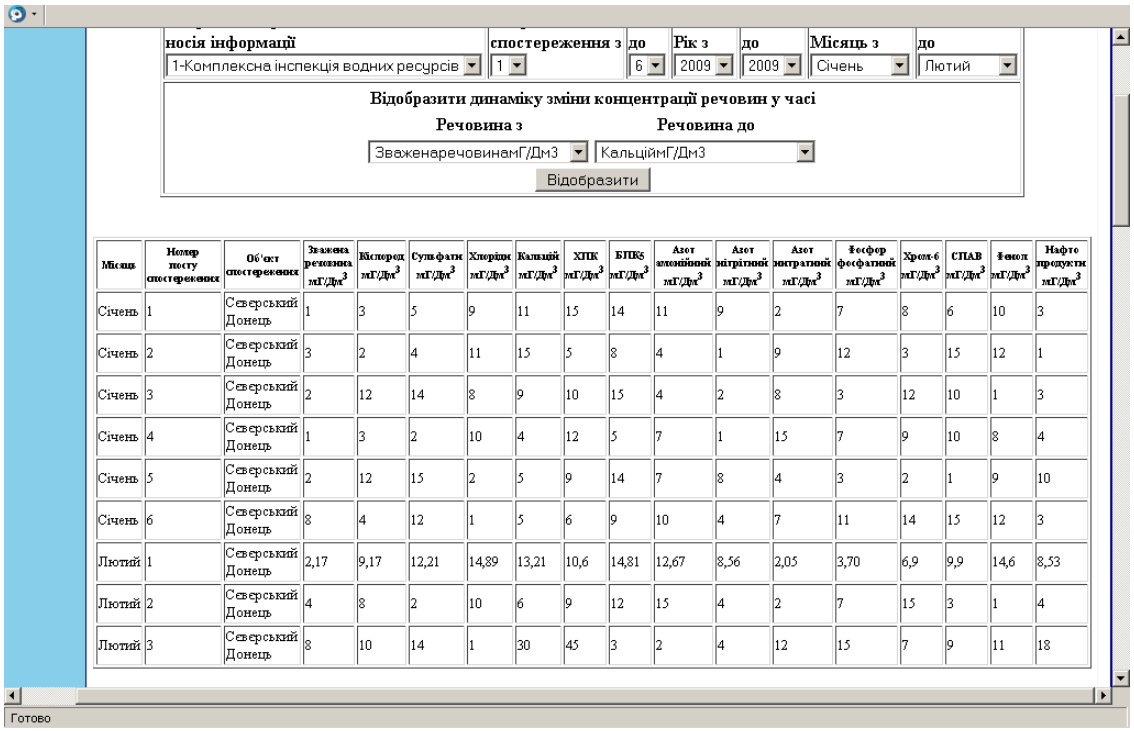


Рисунок 4.21 – Карта моніторингу та фільтр вибору моніторингової інформації



Місяць	Номер посту спостереження	Об'єкт спостереження	Важкі речовини мГ/Дм^3	Кисень мГ/Дм^3	Сульфати мГ/Дм^3	Хлориди мГ/Дм^3	Кальцій мГ/Дм^3	ХПК мГ/Дм^3	БПК5 мГ/Дм^3	Азот амонійний мГ/Дм^3	Азот нітрітний мГ/Дм^3	Азот нітратний мГ/Дм^3	Фосфор фосфатний мГ/Дм^3	Хром-6 мГ/Дм^3	СПАВ мГ/Дм^3	Фенол мГ/Дм^3	Нафтопродукти мГ/Дм^3
Січень	1	Сіверський Донець	1	3	5	9	11	15	14	11	9	2	7	8	6	10	3
Січень	2	Сіверський Донець	3	2	4	11	15	5	8	4	1	9	12	3	15	12	1
Січень	3	Сіверський Донець	2	12	14	8	9	10	15	4	2	8	3	12	10	1	3
Січень	4	Сіверський Донець	1	3	2	10	4	12	5	7	1	15	7	9	10	8	4
Січень	5	Сіверський Донець	2	12	15	2	5	9	14	7	8	4	3	2	1	9	10
Січень	6	Сіверський Донець	8	4	12	1	5	6	9	10	4	7	11	14	15	12	3
Лютий	1	Сіверський Донець	2,17	9,17	12,21	14,89	13,21	10,6	14,81	12,67	8,56	2,05	3,70	6,9	9,9	14,6	8,53
Лютий	2	Сіверський Донець	4	8	2	10	6	9	12	15	4	2	7	15	3	1	4
Лютий	3	Сіверський Донець	8	10	14	1	30	45	3	2	4	12	15	7	9	11	18

Рисунок 4.22 – Обрані дані та початок обробки та аналізу останніх

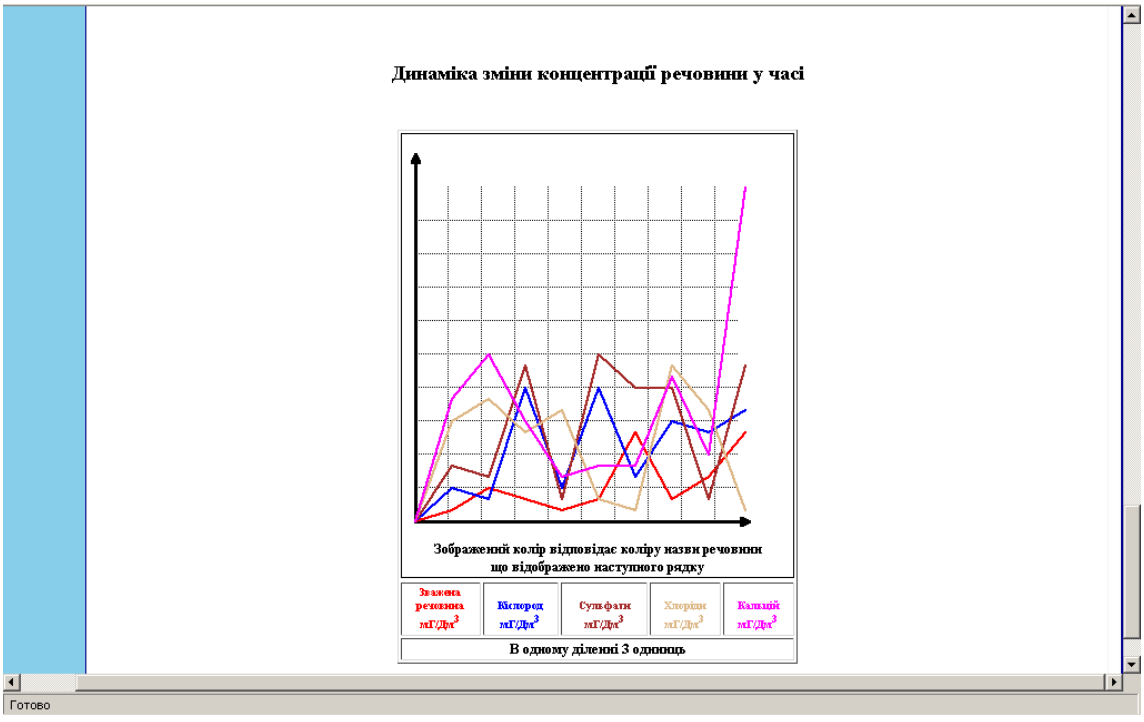


Рисунок 4.23 – Побудова графіку зміни концентрації речовини у часі

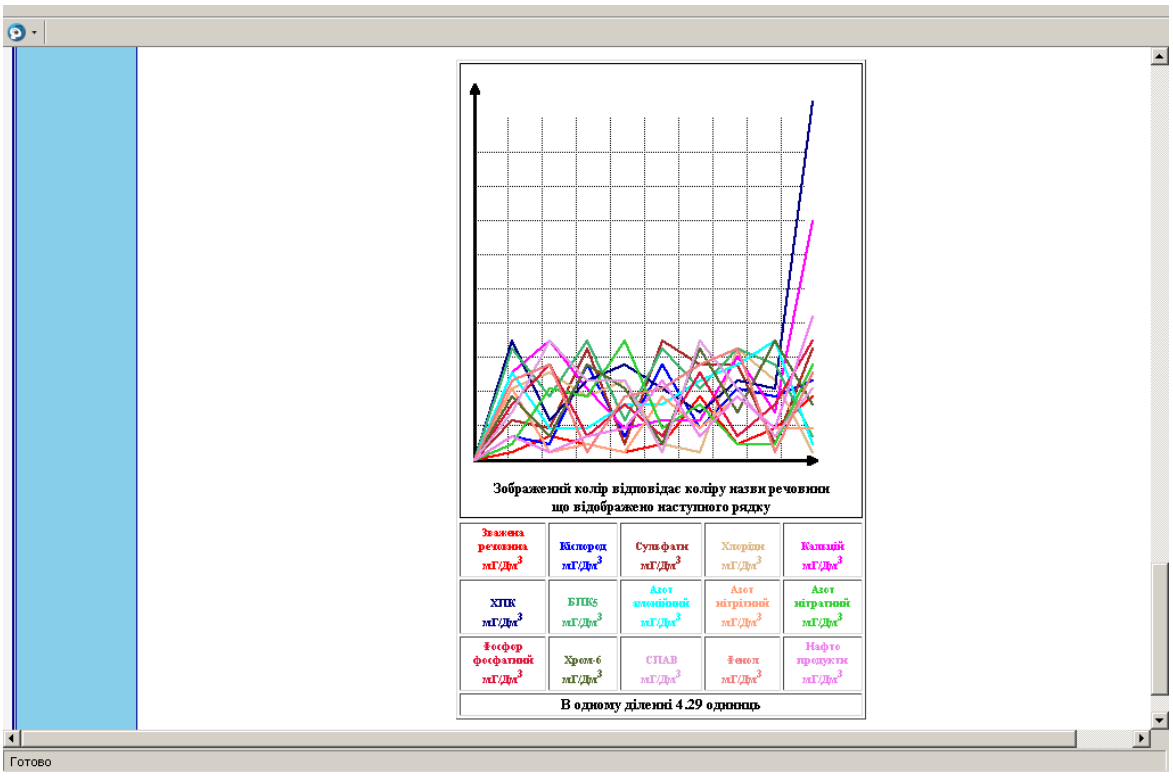


Рисунок 4.24 – Побудова графіку зміни концентрації речовини у часі

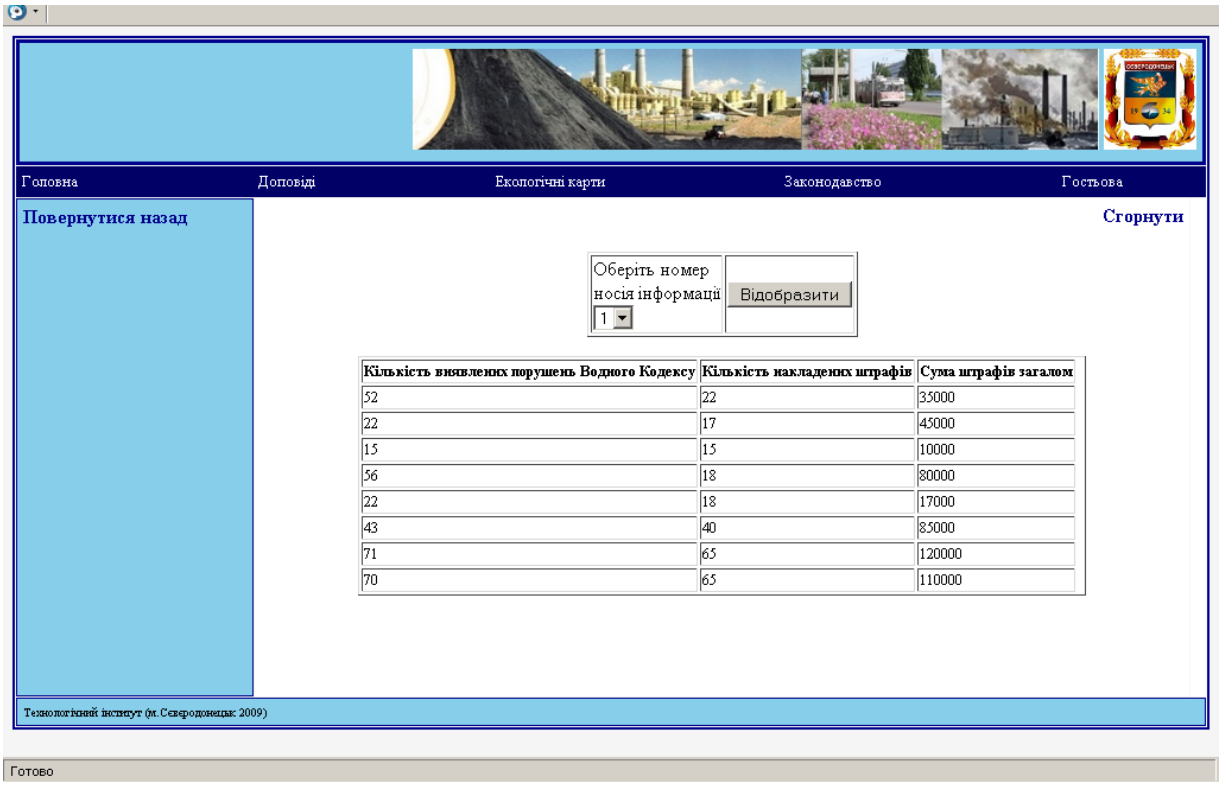


Рисунок 4.25 –Відображення даних загалом

4.5.Діаграми діяльності та блок схеми програмного забезпечення

На рисунку 4.26-4.31 відображено діаграми діяльності та блок схеми основних функцій та сторінок програмного забезпечення.

Рисунок 4.26 – Діаграма діяльності відображення контенту сторінки index1.php

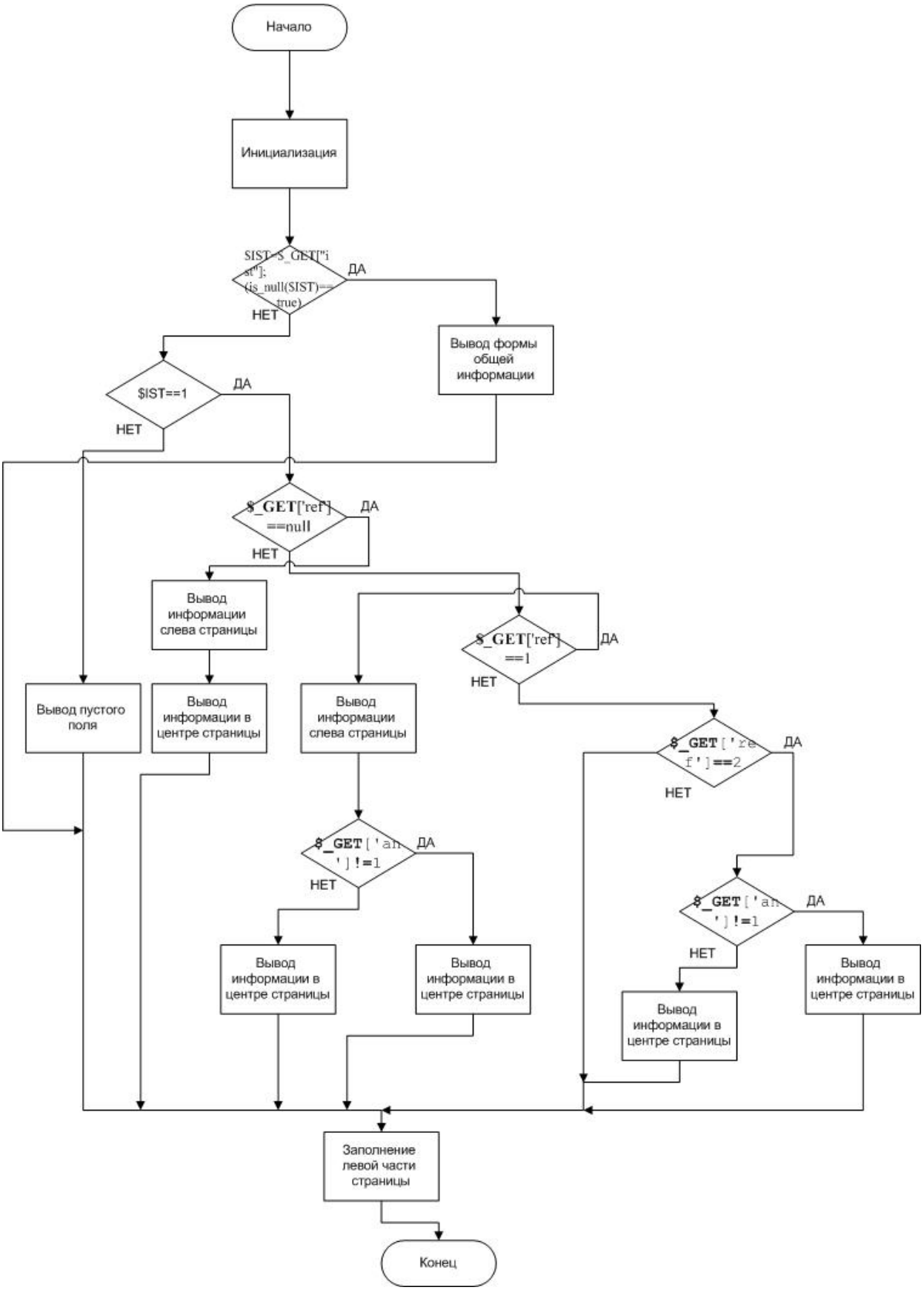


Рисунок 4.27 – Алгоритм роботи сторінки logic.php

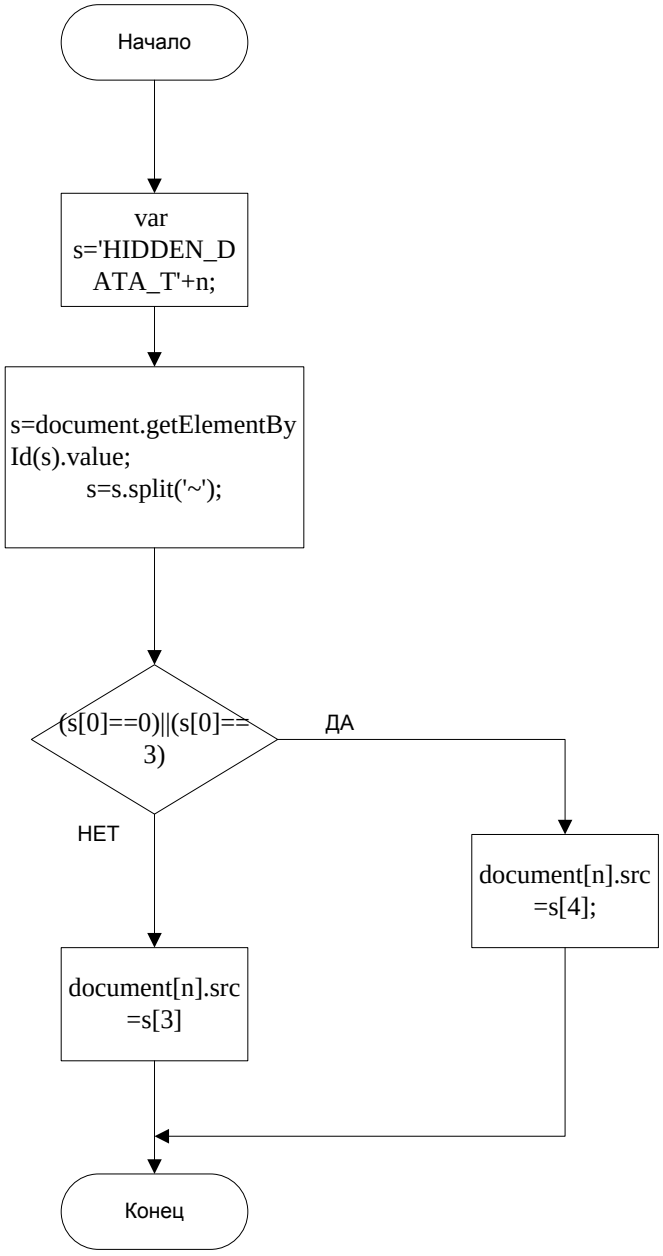


Рисунок 4.28 – Алгоритм роботи функції OverMouse

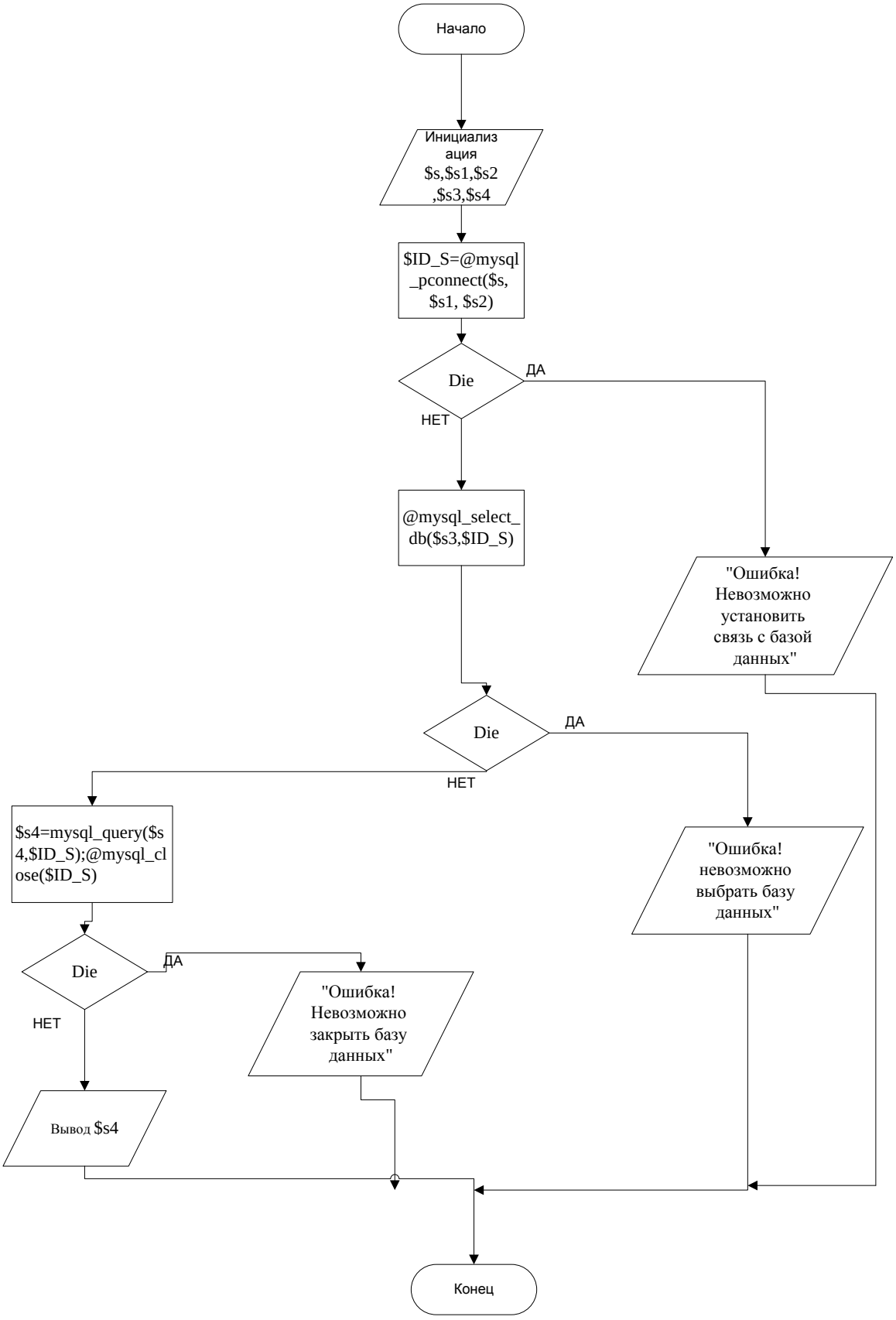


Рисунок 4.29 – Алгоритм роботи функції SQLWork

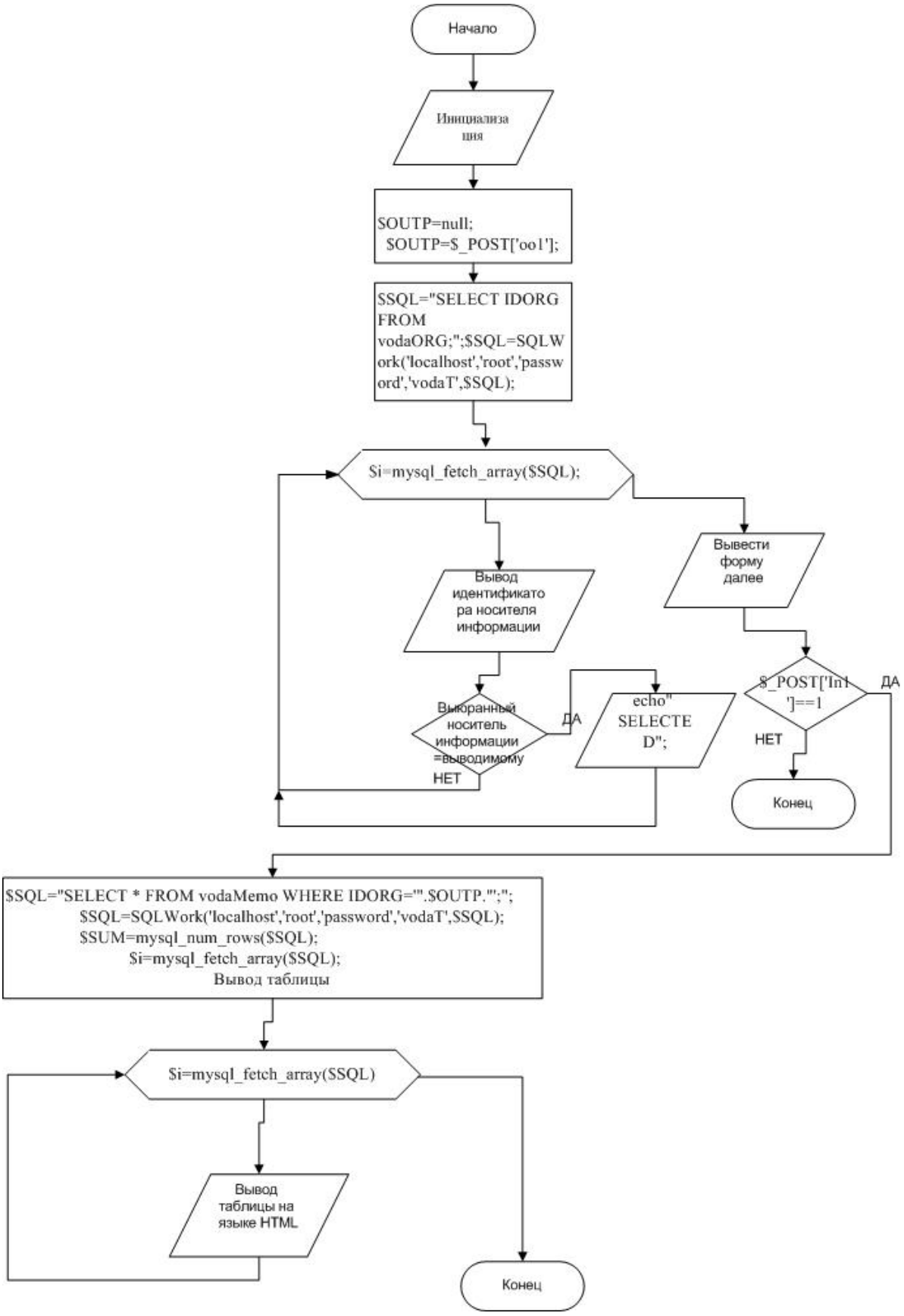


Рисунок 4.30 – Алгоритм роботи функції OutRecords

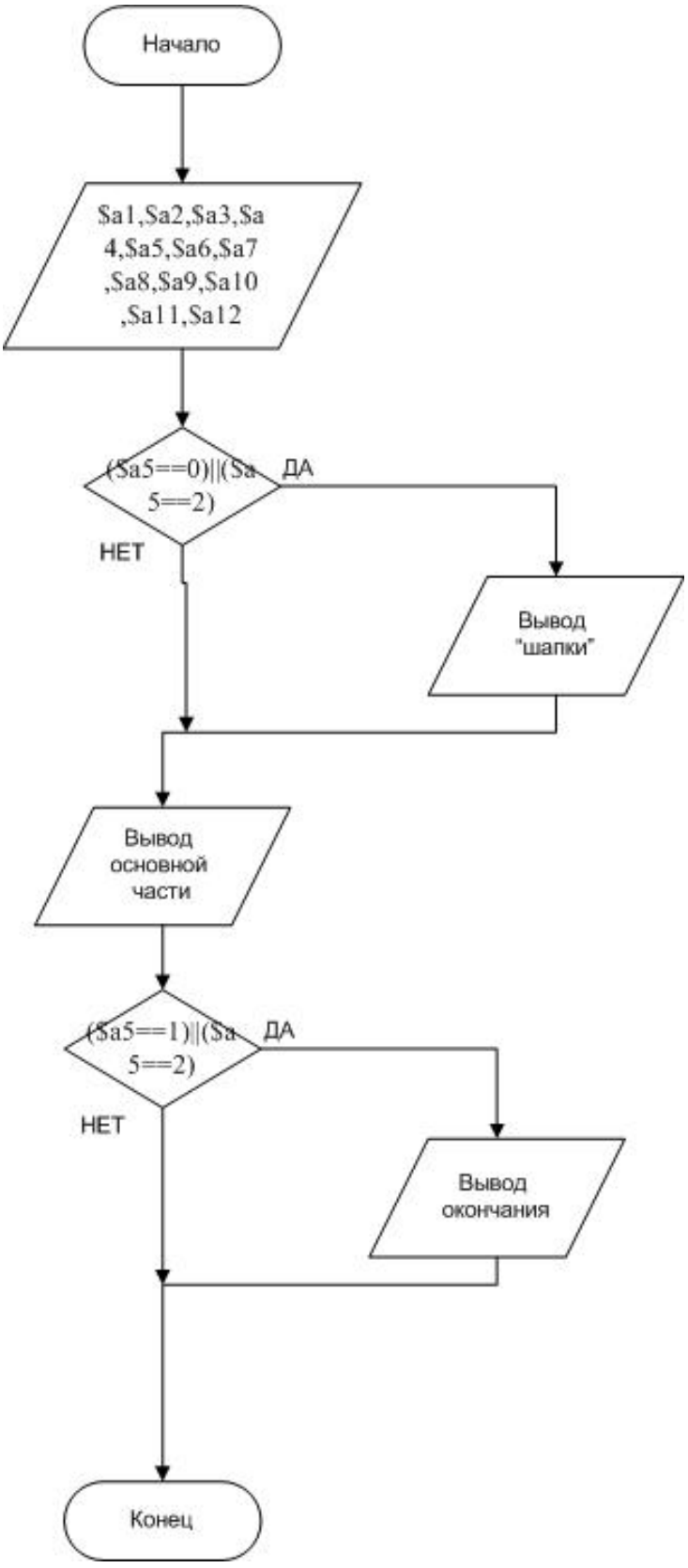


Рисунок 4.31 –Алгоритм роботи функції DrowGraphLine

4.6. Порівняльна характеристика програмного забезпечення на основі технологій PHP, XML, HTML, MYSQL, AHACHE, AJAX, JAVAScript, CSS,UML та ASP.NET, IIS, C#, XML, SQL2000, AJAX, CSS, UML

а) Розробка программного забезпечення з технологією PHP:

Переваги:

1) простий та інтуїтивно зрозумілий код, що спрощую процес стабілізації та пошуку альтернативних рішень;

2) стабільна робота системи;

3) досить проста установка та інсталяція серверу;

4) більш швидка робота серверу, тому що дані не кешуються.

Недоліки:

1) маємо простоту настроювання та стабільно працюючий "движок", але розроблюючи великий проект треба ретельно слідкувати за усім кодом, тому що код пишеться разом з кодом браузерної частини, - це робить останній важким для читання;

2) маємо великий по обсягу код, але досить великий відсоток переробки усього коду в наслідок виявлення помилок;

3) дуже примітивна система класів, що недає можливості організувати більш складну та зручну систему класів;

4) труднощі з типами даних, зокрема, коли треба вивести істотне число. Дуже мало інструментів впливу на авто-приведення типу змінної, що являє собою додаткові труднощі написання коду для приведення типів даних.

б) Розробка программного забезпечення з технологією ASP.NET:

Переваги:

1) відокремлення коду від браузерної частини, що є дуже зручним;

2) маємо об'єктну модель, що потрібно для розробки;

3) маємо роботу серверу за подіями, що являє собою більшу зручность;

4) маємо вже розроблену бібліотеку стандартних об'єктів, наприклад календар та ін.;

5) маємо форму для розташування об'єктів розробки задля можливості візуальнї оцінки;

6) досить стабільний та інтуйовано зрозумілий компілятор, який може компілювати водночас декілька мов проектування, наприклад XML та HTML,

javaScript,C#, та аналізує проект водночас на декількох браузерах, наприклад Opera,IExplorer,FFox;

7) досить візуально зрозумілий та інтуїтивно зрозумілий інтерфейс розробки;

8) якщо пишеться великий проект на довгострокову перспективу, то завдяки добре організованій системі класів маємо у порівнянні з PHP кодом – менший обсяг коду;

9) набагато легше знайти помилку та переробити знов код в данному разі, ніж розбирати код PHP,який пишеться водночас з іншим кодом;

10) досить багато різноманітних бібліотек з вже готовим кодом.

Недоліки:

1) якщо користувач не дуже знакомий з програмуванням, то потрібен деякий час, аби навчитися програмувати у ASP.NET;

2) сервер IIS менш захищений від взлому ніж сервер Apache;

3) має місце стандартний для Mscrosoft ефект “нав’язування” коли, дещо являє собою нове, але не має можливості корегування, як звик користатися користувач;

4) середа розробки потребує глибокого знання особливостей та теорії системи. Тому задля можливості розробки приходиться корегувати та настроювати середу розробки.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

5.1 Аналіз потенційних небезпечних і шкідливих виробничих факторів при роботі з персональним комп'ютером

Основними характеристиками персонального комп'ютера є наступні:

- робоча напруга U=+220У +-5%;
- робочий струм I=2A;
- споживана потужність P=240 Вт.

Роботу користувача розробленої підсистеми варто віднести до категорії Іа (легені фізичні роботи) відповідно до [1]. До даної категорії ставляться всі види діяльності, вироблені сидячи й не потребуючі фізичного напруги.

Приміщення для роботи користувача по ступені небезпеки поразки електричним струмом відповідно до [2] варто віднести до приміщень із підвищеною небезпекою, оскільки в ньому є струмопровідні підстави (залізобетон).

При експлуатації даного програмного продукту відповідно до [3] існують наступні небезпечні й шкідливі виробничі фактори:

а) фізичні:

- підвищений рівень напруги електричного кола, замикання якого може відбутися через тіло людини;
- підвищена або знижена вологість повітря;
- підвищена або знижена рухливість повітря;
- підвищений рівень статичної електрики;
- підвищена напруженість електричних і магнітних полів;
- відсутність або недолік природного світла;
- знижена освітленість робочої зони;
- підвищений рівень на робочому місці шуму;
- підвищений рівень електромагнітного випромінювання;
- знижена контрастність;

б) психофізіологічні:

1) фізичні перевантаження:

- статичні;
- динамічні;

2) нервово-психічні перевантаження:

- розумова перенапруга;
- монотонність праці;
- перенапруга аналізаторів;
- емоційні перевантаження.

5.2 Заходи щодо техніки безпеки

Основним небезпечним фактором при роботі з ЕОМ є небезпека поразки людини електричним струмом, що збільшується тим, що органи почуттів людини не можуть на відстані виявити наявності електричної напруги на встаткуванні.

Проходячи через тіло людини, електричний струм робить на нього складний вплив, що є сукупністю термічного (нагрівання тканин і біологічних середовищ), електролітичного (розкладання крові й плазми) і біологічного (роздратування й порушення нервових волокон і інших органів тканин організму) впливів.

Вага поразки людини електричним струмом залежить від цілого ряду факторів:

– значення сили струму,
– електричного опору тіла людини й тривалості протікання через нього струму,
– роду й частоти струму,
– індивідуальних властивостей людини й навколишнього середовища.

Відповідно до ДЕРЖСТАНДАРТУ 12.1.013-78 «ССБТ. Будівництво. Електробезпечність. Загальні вимоги», приміщення для ЕОМ ставиться до приміщень без підвищеної небезпеки, тобто до приміщень, у яких відсутньої умови, що створюють підвищену або особливу небезпеку. Небезпека поразки електричним струмом існує всюди, де використовується електроустановки, тому приміщення без підвищеної небезпеки не можна назвати безпечними[14].

Електробезпечність забезпечується:

– відповідною конструкцією електроустановок,
– застосуванням технічних способів і засобів захисту,
– організаційними й технічними заходами.

Конструкція електроустановок відповідає умовам їхньої експлуатації й забезпечує захист персоналу від зіткнення зі струмоведучими частинами.

Основними технічними способами й засобами захисту від поразки електричним струмом, використовуваними окремо або в сполученні один з одним, є:

- захисне заземлення;
- звіроферма;
- вирівнювання потенціалів;
- мала напруга;
- електричний поділ мереж;
- захисне відключення;
- ізоляція струмоведучих частин;
- компенсація струмів замикання на землю;
- огороджувальні пристрої;
- попереджувальна сигналізація, блокування, знаки безпеки;
- ізолюючі захисні й запобіжні пристосування.

Основними технічними способами й засобами захисту від поразки електричним струмом, що передбачаються в даному дипломному проекті, є:

– захисне заземлення,
– звіроферма,
– захисне відключення,
– ізоляція струмоведучих частин.

Захисному заземленню або звіропермі підлягають металеві частини електроустановок, доступні для дотику людини й не мають інших видів захисту, що забезпечують електробезпечність. Щоб захистити людину від поразки електричним струмом, захисне заземлення задовольняє ряду вимог. Ці вимоги залежать від напруги електроустановок і потужності джерела харчування. Опір заземлюючого проводу не повинне перевищувати 4 Ом.

Звіроферма в цей час є основним засобом забезпечення електробезпечності. Звіроферма застосовується в мережі із заземленою нейтральною напругою до 1000 У. Звичайно це мережі 220/127 У. Захист людини від поразки електричним струмом у мережах зі звіропермою здійснюється тим, що при замиканні однієї з фаз на звірство корпус у ланцюзі цієї фази виникає струм короткого замикання, що впливає на фотополяриметр захист (плавкий запобіжник, автомат), у результаті чого відбувається автоматичне відключення аварійної ділянки від ланцюга (захисне відключення). Таким чином, зменшується напруга дотику й обмежується час, у плині якого людина, доторкнувшись до корпуса, може потрапити під дію напруги.

Організаційні й технічні заходи щодо забезпечення електробезпечності полягають в основному у відповідному навчанні, інструктажі й допуску до роботи з

електроустановками осіб, що пройшли медичний огляд; виконання ряду технічних мір при проведенні робіт з відключенням напруги в діючих електроустановках або поблизу їх.

Розрахунок струму однофазного короткого замикання вироблятися по формулі:

$$Ik = \frac{Uф}{Zn + \frac{Zm}{3}} \tag{5.1}$$

де Uф - номінальна фазна напруга мережі, В

Zn - повний опір петлі, створеної фазними й нульовими проводами, Ом

Zт – повний опір трансформатора, Ом. Воно рівняється: $Zm = 0{,}1Ом$

Для проводів і жив кабелю:

$$Zn = \sqrt{Rn^2 + Xn^2} \tag{5.2}$$

де $Rn = Rф + Ro$ - сумарний активний опір петлі проведення, Ом

Xn - індуктивний опір петлі проведення, Ом

Перетин мідного проведення $S = 2{,}5мм$;

$$Ro = 7{,}55Ом \cdot км;$$

$$Rф = 7{,}55Ом \cdot км;$$

$$Xn = 0{,}11Ом \cdot км;$$

$$Rn = 7{,}55 + 7{,}55 = 15{,}1Ом \cdot км;$$

$$Zn = \sqrt{15{,}1^2 + 0{,}11^2} = 15{,}1Ом \cdot км$$

Одержуємо: $Ik = \frac{220}{15.1 + 0.1} = 14.5A$

На ЕОМ дію плавкої вставки запобіжника забезпечується, якщо виконується співвідношення:

$$Ik > k \cdot Iн \tag{5.3}$$

де $k = 3$ для плавких вставок

$Iн$ - номінальний струм спрацьовування плавкої вставки, А

$$Iн = \frac{P}{U} \tag{5.4}$$

де P – споживана потужність, рівна 220 Вт;

U - робоча напруга, рівне 220 У

Одержуємо: $Iн = \frac{220}{220} = 1A$

$$Ik = 14{,}5 > k \cdot Iн = 3 \cdot 1 = 3$$

З отриманих даних видно, що умова виконується, завдяки чому відбувається спрацьовування захисного апарата, що забезпечує безпеку роботи обслуговуючого персоналу.

5.3 Міри, що забезпечують виробничу санітарію та гігієну праці

Трудова діяльність людини завжди протікає в певних метеорологічних умовах, які визначаються сполученням температури повітря, швидкості його руху й відносної вологості, тиском і тепловим випромінюванням від нагрітих поверхонь. Тому що експлуатація проектованого програмного засобу відбувається в приміщенні, те ці показники в сукупності (за винятком тиску) називаються мікрокліматом виробничого приміщення. У цей час основним нормативним документом нормалізації мікроклімату є ДЕРЖСТАНДАРТ 12.1.005-88 «ССБТ. Загальні санітарно-гігієнічні вимоги до повітря робочої зони».

Вага праці характеризує сукупний вплив всіх елементів, що становлять умови праці, на працездатність людини, його здоров'я, життєдіяльність і відновлення робочої сили. У такому поданні поняття ваги праці однаково застосовано як до розумового, так і до фізичної праці. Відповідно ДО ДЕРЖСТАНДАРТУ 12.1.005-88 вага роботи персоналу, що обслуговує ЕОМ, ставиться до легкої категорії 1б (роботи, вироблені сидячи, не потребуючі систематичного фізичного напруги й перенесення ваг) [15]. Оптимальні норми мікроклімату в робочій зоні, забезпечувані для робіт легкої категорії 1б наведені в таблиці 5.1.

Таблиця 5.1 - Оптимальні норми мікроклімату

Період року	Температура, ºС	Відносна вологість, %	Швидкість руху повітря, м/с, не більше
Холодний і перехідний	20 – 23	60 - 40	0,2
Теплий	22 – 25	60 – 40	0,2

У приміщенні, де перебуває ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти) і установки кондиціонера БК-2000. Цей метод забезпечує приплив потрібної кількості свіжого повітря, обумовленого в Снип (30 кубічних метрів у годину на один працюючого).

Для захисту від електромагнітного випромінювання передбачаються наступні міри:

– використання захисного екрана (захисні властивості екрана засновані на ефекті ослаблення напруженості електричного поля в просторі),

– віддалення робочого місця не менш, ніж на 0,4-0,5 м, тому що напруженість електричного поля зменшується при віддаленні від джерела поля,

– установлення раціональних режимів роботи персоналу (обмеження часу перебування),

– раціональне розміщення в робочому приміщенні встаткування, що випромінює електромагнітну енергію.

Тому що рівень шуму не перевищує гранично припустимих величин, установлених санітарними нормами, заходу для зниження шуму не проводяться.

Для зниження стомлюваності обслуговуючого персоналу в приміщеннях, де розташовані обчислювальні засоби передбачається використовувати спокійні колірні сполучення й покриття, що не дають відблисків.

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Гарне освітлення діє тонізуюче, створює гарний настрій, поліпшує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть у тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко утомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

У розроблювальному проекті передбачається використовувати сполучене освітлення. У світлий час доби буде використовуватися природне освітлення приміщення через віконні прорізи, в інший час буде використовуватися штучне освітлення. Штучне освітлення створюється лампами накалювання або газорозрядних ламп.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла у світильниках загального висвітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалим терміном служби (до 10000 годин), спектральною сполукою випромінюваного світла, близьким до сонячного. При експлуатації ЕОМ виробляється зорова робота IVго розряду точності (середня точність). При цьому нормована освітленість на робочому місці (Ен) дорівнює 200 лк. Джерелом природного освітлення є сонячне світло. У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає Снип 11-4-79 [16]. Регулярно повинен вироблятися контроль освітленості, що підтверджує, що рівень освітленості задовольняє Снип і для даного приміщення у світлий час доби досить природного освітлення. Світильники загального освітлення

розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого становить 10 м, ширина 10 м, світильниками ЛПО2П, оснащеними лампами типу ЛБ (дві по 80 Вт) зі світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється за коефіцієнтами використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному висвітленні. Розрахунок кількості світильників виробляється по формулі (4.5):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M} \qquad (5.5)$$

де N - число світильників;

E - нормована освітленість;

S - площа підлоги, кв.м;

Z - поправочний коефіцієнт світильника (для стандартних світильників Z = 1.1 - 1.3);

K - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації;

U - коефіцієнт використання, що залежить від типу світильника, показника індексу приміщення й т.п. (U = 0.55 - 0.6);

M - число люмінесцентних ламп у світильнику;

F - світловий потік.

Підставивши числові значення у формулу (4.5), виходить:

$$n = \frac{200 \cdot 10 \cdot 10 \cdot 1{,}2 \cdot 1{,}5}{5400 \cdot 0{,}57 \cdot 2} = 5{,}8$$

Вибирається кількість світильників N рівним 6. Схема розташування світильників показана на рисунку 5.1.

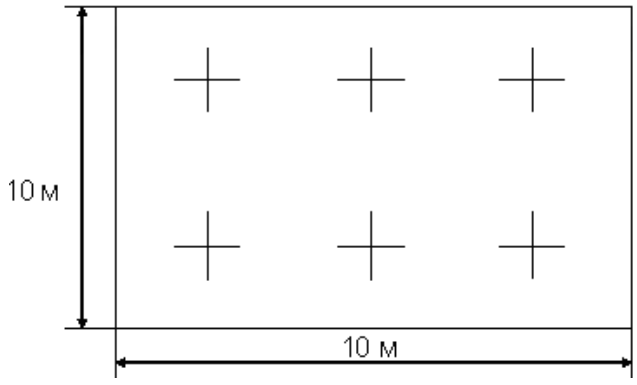


Рисунок 5.1 Схема розташування світильників

5.4 Рекомендації з пожежної безпеки

Виникнення пожежі можливо, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Для оцінки пожежної небезпеки варто проаналізувати ймовірність взаємодії цих трьох факторів.

Горючими матеріалами в приміщенні, де розташовані ЕОМ є:

– поліамід - матеріал корпуса мікросхем, пальне речовина, температура самозапалювання 420 (З,

– полівінілхлорид - ізоляційний матеріал, пальне речовина, температура запалення 335 (З, температура самозапалювання 530 (З,

– стеклотекстолит ДЦ - матеріал друкованих плат, важко займистий матеріал, показник горючості 1.74, не схильний до температурного самозапалювання,

– пластикат кабельний N°.489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1,

– деревина - будівельний і оздоблювальний матеріал, матеріал, з якого виготовлена меблі, горючий матеріал, показник горючості більше 2.1, температура запалення 255 (З, температура самозапалювання 399 (С.

Згідно ОНТП 24-86 таке приміщення ставиться до категорії "В" (пожеже небезпечне) [17].

Потенційними джерелами запалювання можуть бути:

а) іскри й дуги короткого замикання;

б) електрична іскра при замиканні й розмиканні ланцюгів;

в) перегріви від тривалого перевантаження,

г) відкритий вогонь і продукти горіння,

д) наявність речовин, нагрітих вище температури самозапалювання,

е) розрядо - статична електрика.

Причинами можливого загоряння й пожежі можуть бути:

а) несправність електроустановки;

б) конструктивні недоліки встаткування;

в) коротке замикання в електричних мережах;

г) запалення горючих матеріалів, що перебувають у безпосередній близькості від електроустановки.

Виділеннями на пожежі продуктами згоряння є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромин; фосген; хлор і ін.

При горінні пластмас, крім звичайних продуктів згоряння, виділяються різні продукти термічного розкладання: хлорангідридні кислоти; формальдегіди; хлористий водень; фосген; синильна кислота; аміак; фенол; ацетон; стирол.

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем вентиляції й кондиціювання, розвитий системою електроживлення ЕОМ. Небезпека загоряння в ЕОМ зв'язана зі значною кількістю щільно розташованих на монтажних платах і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Висока щільність елементів в електронних схемах приводить до значного підвищення температури окремих вузлів (80...100 (З), що може служити причиною запалення ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів впливу температури може викликати порушення ізоляції й привести до короткого замикання.

Пожежна безпека при застосуванні ЕОМ відповідно до ДЕРЖСТАНДАРТ 12.1.004-91 «Пожежна безпека» забезпечується [18]:

– системою запобігання пожежі,

– системою протипожежного захисту,

– організаційно - технічними заходами.

Запобігти утворенню горючого середовища (замінити горючі речовини й матеріали на негорючі й важко займисті) не надається технічно можливим. Тому проектом передбачаються способи й засоби запобігання утворення (або внесення) у горюче середовище джерел запалювання, таких як:

- застосування електроустаткування, що відповідає пожеже небезпечній і вибухонебезпечній зонам відповідно до ПУЕ;

- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалювання;

- виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної й вище мінімальної енергії запалювання.

Для запобігання пожежі в обчислювальних центрах необхідно виконувати наступні вимоги:

– електроживлення ЕОМ повинне мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження й кондиціювання,

– система вентиляції обчислювальних центрів повинна бути обладнана пристроями, що блокують, що забезпечує її відключення у випадку пожежі. Система повинна обладнатися вогнеперешкоджающими клапанами,

– після закінчення роботи, перед закриттям приміщення, всі електроустановки й персональні комп'ютери необхідно відключити від мережі електроживлення,

– у приміщеннях обчислювальних центрів забороняється:

1) улаштовувати електророзетки на спаленних основах;
2) використовувати синтетичні доріжки й килими;
3) користуватися побутовими електронагрівальними приладами;
4) захаращувати евакуаційні виходи й проходи;
5) улаштовувати на вікнах глухі решітки;
6) залишати без нагляду включену в електромережу апаратуру, використовувану для вимірів і нагляду.

Для зниження пожежної небезпеки передбачається використовувати первинні засоби пожежогасіння, а також систему автоматичної пожежної сигналізації із застосуванням датчиків - оповісків типу ИДФ-1М, які розраховані для контролю площі до 100 кв. м. при висоті стелі до 4-х м. Як первинні засоби пожежогасіння пропонується використовувати:

– ручний вогнегасник ОУ-5,

– повітряно - пінний вогнегасник ОВП-5,

– азбестове полотно 1.5х2 м.

У якості організаційно - технічних мір рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

ВИСНОВКИ

У магістерській роботі було досліджено існуючі методи та технології розробки програмного забезпечення екологічного сервера.

В роботі проведений аналіз існуючих технологій розробки програмних систем і розробка по декількох з них, де враховувалися різні параметри: тип інформації, безпека, особливості розробки програмного забезпечення, надійність і здатність відповідати необхідним характеристикам. Розроблено ряд алгоритмів для забезпечення безпечної передачі інформації в системі, алгоритми статистичного та екологічного аналізу, побудови графіків, підсистеми ухвалення рішень. Для реалізації програмного забезпечення використано Microsoft Visual Studio 2010 та бібліотеки ATL і STL.

У першому та другому розділі відображено короткий екологічний огляд, загальні визначення, розрахунки основних показників, обернена увага на різні особливості і аспекти побудови і аналізу систем екологічного моніторингу, обґрунтовано технічне завдання і показані можливі варіанти його вирішення.

У третьому розділі досліджені існуючі методи та технології розробки програмного забезпечення, а також недоліки і переваги кожного з них. Досліджено C# та PHP на швидкість виконання операцій. За отриманими результатами зроблено висновки.

У четвертому розділі обґрунтовані можливі варіанти вирішення завдання, виходячи з результатів досліджень. Розроблено програмне забезпечення кожного з обраних варіантів вирішення завдання. Програмне забезпечення було спроектовано за допомогою мови UML, далі обґрунтовано та обрано структуру бази даних з можливих варіантів її реалізації. Програмне забезпечення розроблено на основі технологій ASP.NET та PHP. Проведено порівняльний аналіз отриманих результатів та зроблено висновки.

У п’ятому розділі відображена охорона праці.

Дослідження та програмне забезпечення сервера регіональної системи екологічного моніторингу, що відображено у магістерській роботі, повністю відповідають завданню на магістерську роботу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Трельсен Е. Модель COM і вживання ATL 3.0; Пер. з англ. - Спб.: БХВ-Петербург, 2001. - 928 с.: мул.

2. Пітрек М. Секрети системного програмування в Windows - К.: Діалектика, 1996. - 448 с., мул.

3. OPC Data Access Custom Interface Specification. Date : Oct 13 1998. Version 2.0, Author : Opc Foundation

4. Кейт Г. Іспользованіє Visual C#. Спеціальне видання.: Пер. з англ. - К.: Діалектика 1997, - 816 с.: мул.

5. Бьерн Страуструп. Мова програмування C#. Третє видання./Пер. з английского:- Спб.;- М.: «Невський Діалект»-«іздательство БІНОМ»-2000 р. - 991 с.

6. Майкл Дж. Янг: Visual C#. Повне керівництво: Пер. з англ. - К.: Видавнича група BHV, 2000 р. - 1056 с.

7. Дональд Бокс: Суть языка Java/css . Бібліотека програміста. - Спб.: Пітер, 2001. - 400 с.: мул.

8. С. Маклін, Дж. Нафтел, К. Уїльямс : Microsoft .NET Framework 2.0. «Русская редакция», 2003 - 456с.: мул.

9. Матеріал мережі Інтернет

10. Щербаков Е.В., Щербакова М.Е., Охрамовіч В.К.: Автоматизированное проектирование ППО КСУ на базе пакета программ “КВАРЦ” «Наукове видання»,2003-200с.:ил.

11. Типове положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України (НАПБ Б.02.005-2003) [Електронний ресурс] / Законодавство України - Режим доступу: www.URL: http://zakon0.rada.gov.ua/laws/show/z1148-03 - 21.12.2017 р.

12. Санітарні норми мікроклімату виробничих приміщень (ДСН 3.3.6.042.-99) [Електронний ресурс] / Законы Украины - Режим доступу: www.URL: http://uazakon.com/documents/date_42/pg_ikcfxj.htm - 22.12.2017 р.

ДОДАТОК А

Лістинг програми тестування C# та PHP

А.1 Тестування C#

```
<%@ Page Language="C#" ClassName="SpeedTestASP" %>
<%@ assembly Src="SpeedTest.cs" %>
<%@ import Namespace="System.IO" %>
<script runat="server">

   // Insert page code here
      //

      void Page_Load(Object sender, EventArgs e) {
         StringWriter Out = new StringWriter();
         SpeedTest.SpeedTest.Perform(Out);
         Label.Text = Out.ToString();
      }

</script>
<html>
<head>
</head>
<body>
   <form runat="server">
      <pre><asp:Label id="Label" runat="server">Label</asp:Label></pre>
   </form>
</body>
</html>
using System;
using System.IO;
using System.Text;
using System.Globalization;
using System.Reflection;
using System.Diagnostics;
using System.Runtime.InteropServices;

0
namespace SpeedTest
{
        public class SpeedTest
        {
                [STAThread]
                static void Main(string[] args)
                {
                        Perform(Console.Out);
                        #if (DEBUG)
                          Console.ReadLine();
                        #endif
                }

                public static void Perform(TextWriter Out)
                {
                        Out.WriteLine("Performing tests...");
                        Out.WriteLine("{0,-28} {1,10} {2,13} {3,13}
{4,10}","","Time,ms","PassTime,ms","Passes/s","MemUsed,Kb");

                        double FullTime = 0;
                        Type BT  = typeof(BaseTest);
                        Assembly A = Assembly.GetExecutingAssembly();
                        Type[] AT = A.GetTypes();
                        foreach (Type T in AT) {
                                if (T.IsSubclassOf(BT) && !T.IsAbstract) {
                                        ConstructorInfo CI = T.GetConstructor(new Type[0]);
                                        BaseTest Test = (BaseTest)CI.Invoke(new Object[0]);

                                        Out.Write("{0,-28} "," "+Test.Title);

                                        bool   Failed = false;
                                        double      Time = 0;
                                        double  Mem = 0;
                                        HiResTimer C = new HiResTimer();
                                        long S0 = C.Ticks;
```

```
                                                try {
                                                        GC.Collect(GC.MaxGeneration);
                                                        Test.Prepeare();
                                                        Test.Test();
                                                        Test.Overhead();
                                                        Test.Cleanup();

                                                        GC.Collect(GC.MaxGeneration);
                                                        Test.Prepeare();
                                                        long M1 = GC.GetTotalMemory(true);
                                                        long S1 = C.Ticks;
                                                        Test.Test();
                                                        long S2 = C.Ticks;
                                                        long M2 = GC.GetTotalMemory(false);
                                                        long S3 = C.Ticks;
                                                        Test.Overhead();
                                                        long S4 = C.Ticks;
                                                        Test.Cleanup();
                                                        GC.Collect(GC.MaxGeneration);
                                                        Mem  = (M2-M1)/1000.0;
                                                        Time = (S2-S1)*1000.0/C.Frequency;
//                                                      Time = (S2-S1-(S4-S3))*1000.0/C.Frequency;
                                                }
                                                catch (Exception) {
                                                        Failed = true;
                                                }

                                                FullTime += Time;
                                                if (!Failed)
                                                        Out.WriteLine("{0,10:F4} {1,10:F4}E-3 {2,13:N0} {3,10:N1}",
Time, Time/Test.PassCount*1000, Test.PassCount*1000.0/Time, Mem);
                                                else
                                                        Out.WriteLine("{0,10}", "Failed");
                                        }
                                }
                                Out.WriteLine("{0,-28} {1,10:F4}","Done. Common execution time:",FullTime);
                        }
                }

                // High Resolution Timer class
                class HiResTimer
                {
                        private long frequency = 1;
                        private long ticks = 0;

                        [DllImport("Kernel32.dll")]
                        private static extern int QueryPerformanceCounter(ref long Counter);
                        [DllImport("Kernel32.dll")]
                        private static extern int QueryPerformanceFrequency(ref long Freq);

                        // Public method
                        public HiResTimer() {
                          QueryPerformanceFrequency(ref frequency);
                        }
                        public long Frequency {
                          get {return frequency;}
                        }
                        public long Ticks {
                          get {
                                QueryPerformanceCounter(ref ticks);
                                return ticks;
                          }
                        }
                }

                // Base test class
                abstract class BaseTest
                {
                        public string Title = "Unknown";
                        public int PassCount = 2000000;

                        public virtual void Prepeare() {}
                        public abstract int Test();
                        public virtual int Overhead()
                        {
                                unchecked {
```

```
                int i = 0;
                for (; i<PassCount; i++) {}
                return i;
            }
        }
        public virtual void Cleanup() {}
}

// Tests
class EmptyLoopTestC: BaseTest
{
    public EmptyLoopTestC() : base() {
        Title = "Empty loop (checked)";
        PassCount *= 10;
    }

    public override int Test()
    {
        checked {
            int i = 0;
            for (; i<PassCount; i++) {
            }
            return i;
        }
    }
}
class EmptyLoopTestU: BaseTest
{
    public EmptyLoopTestU() : base() {
        Title = "Empty loop (unchecked)";
        PassCount *= 10;
    }

    public override int Test()
    {
        unchecked {
            int i = 0;
            for (; i<PassCount; i++) {
            }
            return i;
        }
    }
}
class LightLoopTestC: BaseTest
{
    public LightLoopTestC() : base() {
        Title = "Light loop (checked)";
        PassCount *= 10;
    }

    public override int Test()
    {
        checked {
            int i = 0;
            int j = 0;
            for (; i<PassCount; i++) {
                j += 1;
                j += 2;
                j += 3;
            }
            return j;
        }
    }
}
class LightLoopTestU: BaseTest
{
    public LightLoopTestU() : base() {
        Title = "Light loop (unchecked)";
        PassCount *= 10;
    }

    public override int Test()
    {
        unchecked {
            int i = 0;
            int j = 0;
            for (; i<PassCount; i++) {
```

```
                                        j += 1;
                                        j += 2;
                                        j += 3;
                                    }
                                    return j;
                                }
                        }
}
class ReflectionMethodCallTestA: BaseTest
{
    public ReflectionMethodCallTestA() : base() {
                        Title = "Reflection method call (A)";
                        PassCount /= 1000;
                    }

                    class InnerA
                    {
                                public int MethodA()
                                {
                                        return 0;
                                }
                                public int MethodB()
                                {
                                        return 1;
                                }
                                public int MethodC()
                                {
                                        return 2;
                                }
                    }

                    public override int Test()
                    {
                        unchecked {
                                        Type TI;
                                        MethodInfo MIA, MIB, MIC;
                                        InnerA A = new InnerA();
                                        Object[] Args = new Object[0];
                                        int i = 0;
                                        int j = 0;
                                        for (; i<PassCount; i++) {
                                                TI = A.GetType();
                                                MIA = TI.GetMethod("MethodA");
                                                j += (int)MIA.Invoke(A, Args);
                                                TI = A.GetType();
                                                MIB = TI.GetMethod("MethodB");
                                                j += (int)MIB.Invoke(A, Args);
                                                TI = A.GetType();
                                                MIC = TI.GetMethod("MethodC");
                                                j += (int)MIC.Invoke(A, Args);
                                        }
                                        return j;
                                }
                    }
}
class ReflectionMethodCallTestB: BaseTest
{
    public ReflectionMethodCallTestB() : base() {
                        Title = "Reflection method call (B)";
                        PassCount /= 1000;
                    }

                    class InnerA
                    {
                                public int MethodA()
                                {
                                        return 0;
                                }
                                public int MethodB()
                                {
                                        return 1;
                                }
                                public int MethodC()
                                {
                                        return 2;
                                }
                    }
```

```
		public override int Test()
		{
			unchecked {
				Type TI;
				MethodInfo MIA, MIB, MIC;
				InnerA A = new InnerA();
				TI = A.GetType();
				MIA = TI.GetMethod("MethodA");
				MIB = TI.GetMethod("MethodB");
				MIC = TI.GetMethod("MethodC");
				Object[] Args = new Object[0];
				int i = 0;
				int j = 0;
				for (; i<PassCount; i++) {
					j += (int)MIA.Invoke(A, Args);
					j += (int)MIB.Invoke(A, Args);
					j += (int)MIC.Invoke(A, Args);
				}
				return j;
			}
		}
}
class VirtualMethodCallTest: BaseTest
{
   public VirtualMethodCallTest() : base() {
		Title = "Virtual method call";
		PassCount *= 1;
		}

		class InnerA
		{
			public virtual int MethodA()
			{
				return 0;
			}
		}
		class InnerB: InnerA
		{
			public override int MethodA()
			{
				return 1;
			}
		}
		class InnerC: InnerB
		{
			public override int MethodA()
			{
				return 2;
			}
		}

		public override int Test()
		{
			unchecked {
				InnerA A = new InnerA();
				InnerA B = new InnerB();
				InnerA C = new InnerC();
				int i = 0;
				int j = 0;
				for (; i<PassCount; i++) {
					j += A.MethodA();
					j += B.MethodA();
					j += C.MethodA();
				}
				return j;
			}
		}
}
class NormalMethodCallTest: BaseTest
{
   public NormalMethodCallTest() : base() {
		Title = "Normal method call";
		PassCount *= 10;
		}

		class InnerA
```

```
		{
			public int MethodA()
			{
				return 1;
			}
		}
		class InnerB
		{
			public int MethodA()
			{
				return 2;
			}
		}
		class InnerC
		{
			public int MethodA()
			{
				return 3;
			}
		}

		public override int Test()
		{
		   unchecked {
				InnerA A = new InnerA();
				InnerB B = new InnerB();
				InnerC C = new InnerC();
				int j = 0;
				int i = 0;
				for (; i<PassCount; i++) {
					j += A.MethodA();
					j += B.MethodA();
					j += C.MethodA();
				}
				return j;
			}
		}
}
class StaticMethodCallTest: BaseTest
{
   public StaticMethodCallTest() : base() {
		Title = "Static method call";
		PassCount *= 10;
		}

		class InnerA
		{
			static public int MethodA()
			{
				return 1;
			}
		}
		class InnerB
		{
			static public int MethodA()
			{
				return 2;
			}
		}
		class InnerC
		{
			static public int MethodA()
			{
				return 3;
			}
		}

		public override int Test()
		{
		   unchecked {
				int j = 0;
				int i = 0;
				for (; i<PassCount; i++) {
					j += InnerA.MethodA();
					j += InnerB.MethodA();
					j += InnerC.MethodA();
				}
```

```
                        return j;
                    }
                }
}

class StringAppendTestA: BaseTest
{
    public StringAppendTestA() : base() {
                Title = "String append (A)";
                PassCount /= 200;
                }

                public override int Test()
                {
                    unchecked {
                        String S = "";
                        int i = 0;
                        for (; i<PassCount; i++) {
                            S += "1";
                        }
                        return i;
                    }
                }
}
class StringAppendTestB: BaseTest
{
    public StringAppendTestB() : base() {
                Title = "String append (B)";
                PassCount /= 1000;
                }

                public override int Test()
                {
                    unchecked {
                        String S = "";
                        int i = 0;
                        for (; i<PassCount; i++) {
                            S += "1234567890";
                        }
                        return i;
                    }
                }
}

class StringBuilderTestA: BaseTest
{
    public StringBuilderTestA() : base() {
                Title = "String builder (A)";
                PassCount /= 2;
                }

                public override int Test()
                {
                    unchecked {
                        StringBuilder S = new StringBuilder();
                        int i = 0;
                        for (; i<PassCount; i++) {
                            S.Append("1");
                        }
                        return i;
                    }
                }
}
class StringBuilderTestB: BaseTest
{
    public StringBuilderTestB() : base() {
                Title = "String builder (B)";
                PassCount /= 10;
                }

                public override int Test()
                {
                    unchecked {
                        StringBuilder S = new StringBuilder();
                        int i = 0;
                        for (; i<PassCount; i++) {
                            S.Append("1234567890");
```

```
                        }
                        return i;
                }
        }
}

class ClassConstrTest: BaseTest
{
    public ClassConstrTest() : base() {
        Title = "Class construction";
        PassCount /= 10;
    }

        class InnerA
        {
                int A;
                double B;
        }
        InnerA[] A;

        public override int Test()
        {
                unchecked {
                        A = new InnerA[PassCount];
                        int i = 0;
                        for (; i<PassCount; i++) {
                                A[i] = new InnerA();
                        }
                        return i;
                }
        }
        public override void Cleanup()
        {
                A = null;
        }
}
class ClassCDTest: BaseTest
{
    public ClassCDTest() : base() {
        Title = "Class constr. & destr.";
        PassCount /= 10;
    }

        class InnerA
        {
                int A;
                double B;
        }
        InnerA[] A;

        public override int Test()
        {
                unchecked {
                        A = new InnerA[PassCount];
                        int i = 0;
                        for (; i<PassCount; i++) {
                                A[i] = new InnerA();
                        }
                        A = null;
                        GC.Collect();
                        return 0;
                }
        }
        public override void Cleanup()
        {
                A = null;
        }
}
class StructConstrTest: BaseTest
{
    public StructConstrTest() : base() {
        Title = "Struct construction";
        PassCount *= 1;
    }

        struct InnerA
        {
```

```
                int A;
                double B;
        }
        InnerA[] A;

        public override int Test()
        {
                unchecked {
                        A = new InnerA[PassCount];
                        return 0;
                }
        }
        public override void Cleanup()
        {
                A = null;
        }
}
class StructCDTest: BaseTest
{
   public StructCDTest() : base() {
         Title = "Struct constr. & destr.";
         PassCount *= 1;
        }
        struct InnerA
        {
                int A;
                double B;
        }
        InnerA[] A;
        public override int Test()
        {
                unchecked {
                        A = new InnerA[PassCount];
                        A = null;
                        GC.Collect();
                        return 0;
                }
        }
        public override void Cleanup()
        {
                A = null;
        }
}
class IntCalcTestC: BaseTest
{
   public IntCalcTestC() : base() {
         Title = "Integers (checked)";
         PassCount *= 1;
        }
        public override int Test()
        {
                checked {
                        int A = 20;
                        int i = 0;
                        for (; i<PassCount; i++) {
                                A += (A*i)%10;
                        }
                        return i;
                }
        }
}
class IntCalcTestU: BaseTest
{
   public IntCalcTestU() : base() {
         Title = "Integers (unchecked)";
         PassCount *= 1;
        }

        public override int Test()
        {
                unchecked {
                        int A = 20;
                        int i = 0;
                        for (; i<PassCount; i++) {
                                A += (A*i)%10;
                        }
                        return i;
```

```
                }
            }
        }

        class FloatCalcTestC: BaseTest
        {
            public FloatCalcTestC() : base() {
                Title = "Floats (checked)";
                PassCount /= 10;
            }

            public override int Test()
            {
                checked {
                    double A = 20;
                    int i = 0;
                    for (; i<PassCount; i++) {
                        A += A+(i/A*i);
                    }
                    return i;
                }
            }
        }
        class FloatCalcTestU: BaseTest
        {
            public FloatCalcTestU() : base() {
                Title = "Floats (unchecked)";
                PassCount /= 10;
            }

            public override int Test()
            {
                checked {
                    double A = 20;
                    int i = 0;
                    for (; i<PassCount; i++) {
                        A += A+(i/A*i);
                    }
                    return i;
                }
            }
        }
}
```

A.2 Тестування PHP

```
<html>
<head>
</head>
<body>
  <pre><? include("SpeedTest.phl") ?></pre>
</body>
</html>
<?php

function GetMicroTime()
{
 $mtime = explode(" ", microtime() );
 return  (double)($mtime[1] + $mtime[0]);
}

function Perform()
{
 printf("Performing tests...\r\n");
 printf("%-28s %10s %13s %13s\r\n","","Time,ms","PassTime,ms","Passes/s");
 $FullTime = 0;
 $Classes = get_declared_classes();
 foreach ($Classes as $k=>$Class) {
   if (get_parent_class($Class)=="basetest") {
     $Test = new $Class;
     printf("%-28s "," ".$Test->Title);
     // Test
     $Test->Prepeare();
     $T1 = GetMicroTime();
     $Test->Test();
     $T2 = GetMicroTime();
     $Test->Cleanup();
```

```
    // Calc
    $T   = $T2-$T1;
    $T1P = $T / $Test->PassCount;
    $PPS = 1 / $T1P;
    $FullTime += $T;
    printf("%5.4f %5.4fE-3 %13.0f\r\n",$T*1000,$T1P*1000000,$PPS);
   }
  }
  printf("%-28s %5.4f\r\n","Done. Common execution time:",$FullTime);
}

@set_time_limit(0);
Perform();
// Base test class
class BaseTest
{
  var $PassCount = 20000;
  var $Title = "Undefined";
  function Prepeare() {}
  function Test() {}
  function Cleanup() {}
}

// Empty loop test
class EmptyLoopTest extends BaseTest
{
  function EmptyLoopTest()
  {
    $this->Title = "Empty loop";
    $this->PassCount *= 10;
  }
  function Test()
  {
    $PC = $this->PassCount;
    for ($i = 0; $i<$PC; $i++) {
    }
  }
}
// Light loop test
class LightLoopTest extends BaseTest
{
  function LightLoopTest()
  {
    $this->Title = "Light loop";
    $this->PassCount *= 10;
  }
  function Test()
  {
    $PC = $this->PassCount;
    $j = 0;
    for ($i = 0; $i<$PC; $i++) {
      $j+=1;
      $j+=2;
      $j+=3;
    }
  }
}
// Virtual method call test
class CVM1
{
  function M1() {return 1;}
}
class CVM2 extends CVM1
{
  function M1() {return 2;}
}
class CVM3 extends CVM2
{
  function M1() {return 3;}
}
class VirtualMethodCallTest extends BaseTest
{
  function VirtualMethodCallTest()
  {
    $this->Title = "Virtual method call";
    $this->PassCount *= 1;
  }
```

```
  function Test()
  {
    $PC = $this->PassCount;
    $O1 = new CVM1;
    $O2 = new CVM2;
    $O3 = new CVM3;
    $j = 0;
    for ($i = 0; $i<$PC; $i++) {
      $j+=$O1->M1();
      $j+=$O2->M1();
      $j+=$O3->M1();
    }
  }
}

// Normal method call test
class CM1
{
  function M1() {return 1;}
}
class CM2
{
  function M1() {return 2;}
}
class CM3
{
  function M1() {return 3;}
}
class NormalMethodCallTest extends BaseTest
{
  function NormalMethodCallTest()
  {
    $this->Title = "Normal method call";
    $this->PassCount *= 1;
  }
  function Test()
  {
    $PC = $this->PassCount;
    $O1 = new CVM1;
    $O2 = new CVM2;
    $O3 = new CVM3;
    $j = 0;
    for ($i = 0; $i<$PC; $i++) {
      $j+=$O1->M1();
      $j+=$O2->M1();
      $j+=$O3->M1();
    }
  }
}
// Function call test
function F1() {return 1;}
function F2() {return 2;}
function F3() {return 3;}
class FunctionCallTest extends BaseTest
{
  function FunctionCallTest()
  {
    $this->Title = "Function call";
    $this->PassCount *= 1;
  }
  function Test()
  {
    $PC = $this->PassCount;
    $j = 0;
    for ($i = 0; $i<$PC; $i++) {
      $j+=F1();
      $j+=F2();
      $j+=F3();
    }
  }
}
// String append (A)
class StringAppendTestA extends BaseTest
{
  function StringAppendTestA()
  {
    $this->Title = "String append (A)";
```

```
    $this->PassCount *= 10;
  }
  function Test()
  {
    $PC = $this->PassCount;
    $S = '';
    for ($i = 0; $i<$PC; $i++) {
      $j+='1';
    }
  }
}
// String append (B)
class StringAppendTestB extends BaseTest
{
  function StringAppendTestB()
  {
    $this->Title = "String append (B)";
    $this->PassCount *= 10;
  }
  function Test()
  {
    $PC = $this->PassCount;
    $S = '';
    for ($i = 0; $i<$PC; $i++) {
      $j+='1234567890';
    }
  }
}
// Class construction
class CC1
{
  var $A;
  var $B;
}
class ClassConstructionTest extends BaseTest
{
  var $A;
  function ClassConstructionTest()
  {
    $this->Title = "Class construction";
    $this->PassCount *= 1;
  }
  function Prepeare()
  {
    $this->A = range(0,$PC-1);
  }
  function Test()
  {
    $A =& $this->A;
    $PC = $this->PassCount;
    for ($i = 0; $i<$PC; $i++) {
      $A[$i] = new CC1;
    }
  }
}
// Class construction & destruction
class ClassCDTest extends BaseTest
{
  var $A;
  function ClassCDTest()
  {
    $this->Title = "Class constr. & destr.";
    $this->PassCount *= 1;
  }
  function Prepeare()
  {
    $this->A = range(0,$PC-1);
  }
  function Test()
  {
    $A =& $this->A;
    $PC = $this->PassCount;
    for ($i = 0; $i<$PC; $i++) {
      $A[$i] = new CC1;
    }
    unset($A);
    unset($this->A);
```

```
  }
}
// Int calc
class IntsTest extends BaseTest
{
  function IntsTest()
  {
    $this->Title = "Ints";
    $this->PassCount *= 10;
  }
  function Test()
  {
    $PC = $this->PassCount;
    $A = 20;
    $i = 0;
    for ($i = 0; $i<$PC; $i++) {
      $A += ($A*$i)%10;
    }
  }
}

// Float calc
class FloatsTest extends BaseTest
{
  function FloatsTest()
  {
    $this->Title = "Floats";
    $this->PassCount *= 5;
  }
  function Test()
  {
    $PC = $this->PassCount;  $A = 20.0;    $i = 0; for ($i = 0; $i<$PC; $i++) {  $A += $A+($i/$A*$i);   } }}?>
```

ДОДАТОК Б

ЛІСТИНГ ПРОГРАМИ

Б.1 Лістинг програми розробленої на основі технологій ASP.NET, IIS, C#,XML,SQL2000,AJAX,CSS,UML

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="ExampleAsp._Default" %>
<html>
<head>
<title>Аналіз стану забружнення атмосферного повітря</title>
</head>
<body>
<script type="text/javascript" src="../wz_jsgraphics.js"></script>
<div align="center"><font size="4" ><b>Інформація про стан забружнення атмосферного повітря
м.Сєвєродонецьк</b></font><br />
<br />
<form id="outme" method="post" runat="server">
<asp:CheckBox ID="CheckBox" Checked="false" runat="server"/><font><b>Листопад</b></font>
<asp:CheckBox ID="CheckBox1" Checked="false" runat="server"/><font><b>Жовтень</b></font>
<asp:CheckBox ID="CheckBox2" Checked="false" runat="server"/><font><b>Вересень</b></font>
<asp:Button ID="but" runat="server" Text="Відобразити" BorderColor="Red"
   onclick="but_Click" style="margin-left: 45px" />
<div align="right">
<asp:Calendar ID="cal" runat="server"  NextPrevStyle-Wrap="true" ></asp:Calendar></div>
<asp:Button ID="REZ" runat="server" Text="Розрахувати" onclick="REZ_Click" />
</form>
</div>
</body>
</html>

using System;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
namespace ExampleAsp
{
   public partial class _Default : System.Web.UI.Page
   {
      // Шапка
      String[] INFO = {"№","Шкідливі домішки","ПЗС 1,2","Концентрація, мг/м","Серед.міс.","Макс. разова",
                  "ГДК, мг/м","Серед.добова","Макс разова","Загальна кількість спостережень","Кількість сопстережень з
концентраціями","1 ГДК і вище","5 ГДК і вище","10 ГДК і вище",
                  "Примітка"};
      // Данные за Октябрь
      Double[] INF0 = { 1, 0.1, 0.3, 0.15, 0.5, 60, 0, 0, 0, 1, 0.012, 0.04, 0.05, 0.5, 119, 0, 0, 0, 1, 4, 6, 3, 5, 60, 0, 0, 0, 1 , 0.03, 0.07,
0.04, 0.085, 119, 0, 0, 0, 1, 0.03, 0.9, 0.2, 0.2, 119, 0, 0, 0, 1, 0.02, 0.08, 0.04, 0.2, 119, 0, 0, 0, 1, 0.01, 0.026, 0.003, 0.035, 60, 0, 0, 0 };
      // Данные за Сентябрь
      Double[] INF1 = { 1, 0.1, 0.3, 0.15, 0.5, 50, 0, 0, 0, 1, 0.012, 0.034, 0.05, 0.5, 95, 0, 0, 0, 1, 4, 6, 3, 5 , 50, 0, 0, 0, 1, 0.03, 0.07,
0.04, 0.085, 95, 0, 0, 0, 1, 0, 03, 0.09, 0.2, 0.2, 95, 0, 0, 0, 1, 0.02, 0.07, 0.04, 0.2, 95, 0, 0, 0, 1, 0.009, 0.033, 0.03 3, 0.035, 50, 0, 0, 0 };
      // Данные за Август
      Double[] INF2 = { 1, 0.1, 0.3, 0.15, 0.5, 50, 0, 0, 0, 1, 0.012, 0.037, 0.05, 0.5, 96, 0, 0, 0, 1, 4, 5, 3, 5, 50, 0, 0, 0, 1, 0.02, 0.07,
0.04, 0.085, 96, 0, 0, 0, 1, 0.03, 0.08, 0.2, 0.2, 96, 0, 0, 0, 1, 0.02, 0.7, 0.04, 0.2, 96, 0, 0, 0, 1, 0.008, 0.023, 0.003, 0.035, 50, 0, 0, 0 };
      String[] INF3 = { "Пил", "Оксид сірки", "Диоксид сірки", "Оксид вуглецю", "Диоксид азоту", "Хлористий водень",
"Аміак", "Формальдегід" };
      String[] INF4 = { "Середні за рік", "Клас небезпеки", "Константа Ci", "ГДКсс", "ГДКмр", "Ссс", "Смакс", "Qi",
                  "V по Ссс", "Vпо С макс", "См", "См по С макс", "Qмр по Ссс", "Qмр по Смакс",
                  "Li", "КИЗА"};
      Double[] INF5 =
{4,0.9,0.15,0.5,0.1,0.275,0,0,0,0.1,0.3,0.118,0.446,0.694,3,1,0.05,0,0.013,0 .039,0.0005,0.04,0.013,0.013,0.045,0.013,0.042,0.245,4,0.9,3,5,3.75,
5.5,0.5,0.133,
      0.09,4,6,58.83,124.86,1.22,2,1.3,0.04,0.85,0.023,0.07,0.005,0.18,0.07,0.03,0.08,0.03,0.08,0.6,0.2,2,1.3,0.02,0.2,0.03,0.08,0, 0,0,0.03,0.
09,0.03,0.09,1.69,4,0.9,0.04,0.2,0.02,0.08,0.005,0.2,0.07,0.3,0.08,0.02,0.09,0.59,2,1.3,0.003,1.35,0.01,0.02,0.002,0.22,0.07,0.013,0.033,0.01,0.0
3,4.78};
      int Opozn;
      public void OutTB(Double[] INF21)
      {
         HtmlGenericControl ont = new HtmlGenericControl("div");
```

```
ont.Attributes.Add("align","center");
switch (Opozn ) {
   case 1:
      ont.InnerHtml = "<b>Листопад</b>";
      break;
   case 2:
      ont.InnerHtml = "<b>Жовтень</b>";
      break;
   case 3:
      ont.InnerHtml = "<b>Вересень</b>";
      break;
};
Controls.Add(ont);
int[] mas1={3,4,5,6,8,9,10};
Table tb1 = new Table();
tb1.BorderWidth = 1;
TableRow r1 = new TableRow();
TableRow r2 = new TableRow();
for (int i = 0; i <= 11; i++) {
   if (i == 11)
   {
      TableHeaderCell hc = new TableHeaderCell(); hc.BorderWidth = 1;
      hc.Text = INFO[i + 3];
      r1.Cells.Add(hc);
   };
   if ((i == 9)||(i==10))
   {
      TableHeaderCell hc1 = new TableHeaderCell();
      hc1.BorderWidth = 1;
      hc1.Text = INFO[i + 3];
      r2.Cells.Add(hc1);
   };
   if (i == 8)
   {
      TableHeaderCell hc = new TableHeaderCell(); hc.BorderWidth = 1;
      hc.Text = INFO[i + 2];
      TableHeaderCell hc1 = new TableHeaderCell();
      hc1.BorderWidth = 1;
      hc1.Text = INFO[i + 3];
      r1.Cells.Add(hc);
      r2.Cells.Add(hc1);
   };
   if (i == 7)
   {
      TableHeaderCell hc = new TableHeaderCell(); hc.BorderWidth = 1;
      hc.Text = INFO[i + 2];
      r1.Cells.Add(hc);
   };

   if (i == 6)
   {
      TableHeaderCell hc1 = new TableHeaderCell();
      hc1.BorderWidth = 1;
      hc1.Text = INFO[i + 2];
      r2.Cells.Add(hc1);
   };
   if (i == 5) {
      TableHeaderCell hc = new TableHeaderCell(); hc.BorderWidth = 1;
      hc.Text = INFO[i + 1];
      TableHeaderCell hc1 = new TableHeaderCell();
      hc1.BorderWidth = 1;
      hc1.Text = INFO[i + 2];
      r1.Cells.Add(hc);
      r2.Cells.Add(hc1);
   };

   if (i == 3)
   {
      TableHeaderCell hc = new TableHeaderCell(); hc.BorderWidth = 1;
      hc.Text = INFO[i];
      TableHeaderCell hc1 = new TableHeaderCell();
      hc1.BorderWidth = 1;
      hc1.Text = INFO[i + 1];
      r2.Cells.Add(hc1);
      r1.Cells.Add(hc);
   };
   if (i==4) {
```

```
                    TableHeaderCell hc1 = new TableHeaderCell();
                    hc1.BorderWidth = 1;
                    hc1.Text = INFO[i + 1];
                    r2.Cells.Add(hc1);
                  };
                  if ((i == 0) || (i == 1) || (i == 2))
                  {
                    TableHeaderCell hc = new TableHeaderCell(); hc.BorderWidth = 1;
                    hc.Text = INFO[i];
                    r1.Cells.Add(hc);
                  };
                };
                r1.Cells[0].RowSpan = 2;
                r1.Cells[1].RowSpan = 2;
                r1.Cells[2].RowSpan = 2;
                r1.Cells[3].ColumnSpan= 2;
                r1.Cells[4].ColumnSpan = 2;
                r1.Cells[5].RowSpan = 2;
                r1.Cells[6].ColumnSpan = 3;
                r1.Cells[7].RowSpan = 2;
                tb1.Rows.Add(r1);
                tb1.Rows.Add(r2);
                ////////////////////////////////////////////////
        //Данные
                for (int j = 0; j <= 6; j++)
                {
                  TableRow tr = new TableRow();
                  for (int i = 0; i <= 11; i++)
                  {
                    TableCell td = new TableCell(); td.BorderWidth = 1; td.HorizontalAlign =
System.Web.UI.WebControls.HorizontalAlign.Center;
                    if (i == 0) { td.Text = (j+1).ToString(); tr.Cells.Add(td); continue; };
                    if (i == 1) { td.Text = INF3[j].ToString(); tr.Cells.Add(td); continue; };
                    if (i==11) { td.Text = "Немає"; tr.Cells.Add(td); continue; };
                    td.Text = INF21[i - 2+9*j].ToString();
                    tr.Cells.Add(td);
                  };
                  tb1.Rows.Add(tr);
                };
                 Controls.Add(tb1);
                 HtmlGenericControl ccd = new HtmlGenericControl("br");
                 Controls.Add(ccd);
              }
              private void Stat() {

                HtmlGenericControl ccd4 = new HtmlGenericControl("div");
                ccd4.Attributes.Add("align", "center");
                ccd4.InnerHtml = "<b>Розрахунок</b>";
                HtmlGenericControl ccd6 = new HtmlGenericControl("HR");
                Controls.Add(ccd4);
                Controls.Add(ccd6);
                Table tbRez = new Table();
                tbRez.BorderWidth = 1;
                TableHeaderRow Row1 = new TableHeaderRow();
                for (int i = 0; i <= 14; i++) {
                  TableCell TdRez1 = new TableCell(); TdRez1.BorderWidth = 1;
                  TdRez1.Text = INF4[i];
                  Row1.Cells.Add(TdRez1);
                };
                tbRez.Rows.Add(Row1);

                // Выводится результат расчетов

                for (int j = 0; j <= 6; j++)
                {
                  TableRow Row2 = new TableRow();
                  for (int i = 0; i <= 14; i++)
                  {
                    TableCell tdcell = new TableCell(); tdcell.BorderWidth = 1; tdcell.HorizontalAlign =HorizontalAlign.Center;
                    if (i == 0) { tdcell.Text = INF3[j].ToString(); Row2.Cells.Add(tdcell); continue; };
                    tdcell.Text = INF5[i - 1 + 14 * j].ToString();
                    Row2.Cells.Add(tdcell);
                  };
                  tbRez.Rows.Add(Row2);
                };
                TableRow Row3 = new TableRow();
                TableCell tdcell1 = new TableCell(); tdcell1.BorderWidth = 1; tdcell1.HorizontalAlign = HorizontalAlign.Center;
```

```
    tdcell1.Text = "КИЗА = 9.85";
    Row3.Cells.Add(tdcell1);
    tbRez.Rows.Add(Row3);
    Controls.Add(tbRez);
    HtmlGenericControl ccd = new HtmlGenericControl("br");
    Controls.Add(ccd);
}
private void OutGraph() {
    Double[] Buf1 = { 1.5, 0.5, 3, 5, 0.8, 2, 4.6 };
    Double[] Buf2 = { 2.5, 0.9, 7, 3.8, 0.27, 0.82, 4.52 };
    Table tbG = new Table(); tbG.BorderWidth = 0;
    TableRow xc = new TableRow();
    for(byte i=0;i<7;i++){
        TableCell tc = new TableCell();
        Image img = new Image();
        img.ImageUrl = "black.jpg";
        img.Height = Convert.ToInt32(150 / 7 * Buf1[i]);
        img.Width = 15;
        tc.Controls.Add(img);
        tc.BorderWidth = 0;
        tc.HorizontalAlign = HorizontalAlign.Center;
        tc.VerticalAlign = VerticalAlign.Bottom;
        xc.Cells.Add(tc);
        TableCell tc1 = new TableCell();
        Image img1 = new Image();
        img1.ImageUrl = "red.jpg";
        img1.Height = Convert.ToInt32(150 / 7 * Buf2[i]);
        img1.Width = 15;
        tc1.Controls.Add(img1);
        tc1.BorderWidth = 0;
        tc1.HorizontalAlign = HorizontalAlign.Center;
        tc1.VerticalAlign = VerticalAlign.Bottom;
        TableCell tc2 = new TableCell();
        tc2.BorderWidth = 0;
        tc2.HorizontalAlign = HorizontalAlign.Center;
        tc2.VerticalAlign = VerticalAlign.Bottom;
        tc2.Width = 5;
        xc.Cells.Add(tc1);
        xc.Cells.Add(tc2);
    };
    tbG.Rows.Add(xc);
    HtmlGenericControl Ramka = new HtmlGenericControl("div");
    Ramka.Attributes.Add("align", "right");
    Ramka.Controls.Add(tbG);
    Controls.Add(Ramka);
    HtmlGenericControl ccd3 = new HtmlGenericControl("br");
    Controls.Add(ccd3);
}
protected void Page_Load(object sender, EventArgs e)
{
    REZ.Style.Value = "visibility:hidden";
    if (CheckBox.Checked == true)
    {
        Opozn = 1;
        OutTB(INF0);
    };
    if (CheckBox1.Checked == true)
    {
        Opozn = 2;
        OutTB(INF1);
    };
    if (CheckBox2.Checked == true)
    {
        Opozn = 3;
        OutTB(INF2);
    };
    if ((CheckBox.Checked == true) && (CheckBox1.Checked == true) && (CheckBox2.Checked == true))
    {
        REZ.Style.Value = "visibility:visible";
    };
}
protected void but_Click(object sender, EventArgs e)
{
}
protected void REZ_Click(object sender, EventArgs e)
{
    Stat();
```

```
OutGraph();     } } }
```

Б.2 Лістинг програми розробленої на основі технологій PHP,XML,HTML,MYSQL,AHACHE,AJAX, JAVAScript,CSS,UML

analization.php

```
<script type="text/javascript" src="resource/wz_jsgraphics.js"></script>
<script language="javascript">
function changt(a){
Leave_Data_Delete('0','0','~','~','1',window.document.getElementById(a).value,'a','ist','ref','an');
};
function ch1(a){
if(window.document.getElementById(a).checked){
window.document.getElementById(window.document.getElementById(a).value).style.visibility="visible";}else{
window.document.getElementById(window.document.getElementById(a).value).style.visibility="hidden";};
};
</script>

<?php

function DrowGraphLine($a1,$a2,$a3,$a4,$a5,$a6,$a7,$a8,$a9,$a10,$a11,$a12){
    if(($a5==0)||($a5==2)):
        ?>
        <div id="GR<?echo $a8?>"  style="position:relative;top:<?echo $a9?>px;left:<?echo $a10?>px;"><?echo $a11?> </div>
        <script type="text/javascript">
        <!--
        <?echo $a12?>=new jsGraphics("GR<?echo $a8?>");
        <?
    endif;
    ?>
    <?echo $a12?>.setColor("<?print $a6;?>");
    <?echo $a12?>.setStroke(<?print $a7;?>);
    <?echo $a12?>.drawLine(<?echo "$a1,$a2,$a3,$a4";?>);
    <?echo $a12?>.paint();
    <?
    if(($a5==1)||($a5==2)):
        ?>
        //-->
        </script>
        <?
    endif;
    return 0;
};


//данный модуль представляет собой общий статистический анализ данных
function Graph($MAS,$L,$MAS1){
    ?>
    <table border="0">
    <tr>
    <td colspan="<?echo count($MAS);?>" align="center" valign="middle">
    <b><font size="4"><?echo $L?></font></b>
    </td>
    </tr>
    <tr>
    <?
    $i=0;
    while($i<count($MAS)):
        ?>
            <td valign="bottom" align="center"><div><b><font size="2"><?$UU=round($MAS[$i],3); echo $UU;?></font></b></div>
        <img  src="/images/stolb.jpg" height="<?$KOL=round($MAS[$i]*233/max($MAS)); echo $KOL;?>" width="28" border="3">
        </td>
        <?

        $i++;
    endwhile;
    ?>
    </tr>

    <tr>
    <?
    $i=0;
    while($i<count($MAS1)):
```

```
                ?>
                <td valign="bottom" align="center">
                <font size="1"><b><?echo $MAS1[$i];?></b></font>
                </td>
                <?
                $i++;
        endwhile;
        ?>
        </tr>
        </table>

        <?
        return 0;
};

function OutRecords(){
        $OUTP=null;
        $OUTP=$_POST['oo1'];
        ?>
        <form name="pt" method="POST">
        <table border="1" id="tdb" >
        <tr>
        <td>
        Оберіть номер<br> носія інформації<br>
        <select size="1" name="oo1">
        <?
        $SQL="SELECT IDORG FROM vodaORG;";
        $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
        while($i=mysql_fetch_array($SQL)):
                ?>
                <option value="<?echo $i['IDORG'];?>" <?
                if($i['IDORG']==$OUTP): echo" SELECTED"; endif;?> > <?echo $i['IDORG'];?></option>
                <?
                $i++;
        endwhile;
        ?>
        </select>
        </td>

        <td>
        <input type="submit" value="Відобразити">
        <input type="hidden" value="1" name="In1">
        </td>


        </tr>
        </table>
        </form>

        <?
        //Вывести данные с учетом сортировки
        if($_POST['In1']==1){
                $SQL="SELECT * FROM vodaMemo WHERE IDORG='".$OUTP."';";
                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                $SUM=mysql_num_rows($SQL);
                $i=mysql_fetch_array($SQL);
                ?>
                <table border="1">
                <tr>
                <td>
                <b><font size="2"> <?echo $i['P1'];?></font></b>
                </td>
                <td>
                <b>     <font size="2" >    <?echo $i['P3'];?></font></b>
                </td>
                <td>
                <b>     <font size="2" >    <?echo $i['P5'];?></font></b>
                </td>
                <td>
                <b>     <font size="2" >    <?echo $i['P7'];?></font></b>
                </td>
                </tr>
                <?
                while($i=mysql_fetch_array($SQL)):
                        ?>
                        <tr>
                        <td>
```

```
                                        <font size="2" >      <?echo $i['P2'];?></font>
                                        </td>
                                        <td>
                                        <font size="2" >      <?echo $i['P4'];?></font>
                                        </td>
                                        <td>
                                        <font size="2" >      <?echo $i['P6'];?></font>
                                        </td>
                                        <td>
                                        <font size="2" >      <?echo $i['P8'];?></font>
                                        </td>
                                        </tr>
                                        <?
                                endwhile;
                                ?>
                                </table>
                                <?
        };};

        function OutTableForGraph(){

                global $SET1;
                $SET1=null;
                $SET1=$_POST['SelGr'];
                global $SET2;
                $SET2=null;
                $SET2=$_POST['SelGr1'];
                ?>
                <table border="0">
                <tr>
                <td colspan="2" align="center" valign="middle">
                <font size="3"><b>Відобразити динаміку зміни концентрації речовину часі</b></font>
                </td>
                </tr>
                <tr>
                <th>
                <font size="3">
                Речовина з
                </font>
                </th>
                <th>
                <font size="3">
                Речовина до
                </font>
                </th>
                </tr>
                <tr>
                <td>
                <select name="SelGr" size="1">
                <? for($i=0;$i<15;$i++){?>
                        <option <?if($SET1==$i+1): echo"SELECTED"; endif; ?> value="<?echo $i+1;?>"><?echo
Titul($i);?></option>
                        <?
                };
                ?>
                </select>
                </td>
                <td>
                <select name="SelGr1" size="1">
                <? for($i=0;$i<15;$i++){?>
                        <option <?if($SET2==$i+1): echo"SELECTED"; endif; ?> value="<?echo $i+1;?>"><?echo
Titul($i);?></option>
                        <?
                };
                ?>
                </select>
                </td>
                </tr>
                </table>
                <?
                return true;
        };

        function SELRecords(){
                $ID_ORG=null;
                $ID_ORG=$_GET['a'];
                global $IDORGSELECTED;
```

```
                $IDORGSELECTED=$ID_ORG;
                ?>
                <form name="poiskt" method="POST">
                <table border="1" id="tdb"  >
                <tr align="left" valign="bottom">
                <td>
                <font ><b>              Оберіть номер<br> носія інформації</b></font><br>
                <select id="nt1" size="1" name="oo1" onchange="javascript:changt(this.id);">
                <option selected value=null>#</option>
                <?
                $SQL="SELECT IDORG, NameORG FROM vodaORG;";
                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);

                while($i=mysql_fetch_array($SQL)):
                        ?>
                        <option <?if($i['IDORG']==$ID_ORG): echo"SELECTED"; endif; ?> value="<?echo
$i['IDORG'];?>"><?$s=$i['IDORG'];$s1=$i['NameORG'];echo $s,"-",$s1; ?></option>
                        <?
                        $i++;
                endwhile;
                ?>
                </select>
                </td>

                <td>
                <b>Оберіть пост<br> спостереження з</b><br> <?$ZXv=$_POST['xz']; ?>
                <select  name="xz" size="1">
                <?
                if($ID_ORG){
                        $SQL="SELECT DatchN FROM vodaORG WHERE (IDORG='".$ID_ORG."');";
                        $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                        $SQL=mysql_fetch_array($SQL);
                        $SQL=$SQL['DatchN'];
                        $i=1;
                        while($i<=$SQL):
                                ?>
                                <option <?if($ZXv==$i): echo"SELECTED"; endif; ?> value="<?echo $i;?>" ><?echo
$i;?></option>
                                <?
                                $i++;
                endwhile;}else{ ?><option >#</option><?};
                ?>
                </select>
                </td>
                <td>
                <b>до</b><br><?$ZXv111=$_POST['xz1']; ?>
                <select  name="xz1" size="1">
                <?
                if($ID_ORG){
                        $i=1;
                        while($i<=$SQL):
                                ?>
                                <option <?if($ZXv111==$i): echo"SELECTED"; endif; ?> value="<?echo $i;?>"><?echo
$i;?></option>
                                <?
                                $i++;
                endwhile;}else{ ?><option >#</option><?};
                ?>
                </select>
                </td>
                <td>
                <b>Рік з</b><br> <?$godt=$_POST['godt']; ?>
                <select name="godt" size="1">
                <option <?if($godt==9): echo"SELECTED"; endif; ?> value="9">2009</option>
                <option <?if($godt==10): echo"SELECTED"; endif; ?> value="10">2010</option>
                <option <?if($godt==11): echo"SELECTED"; endif; ?> value="11">2011</option>
                </select>
                </td>

                <td>
                <b>до</b><br><?$godt123=$_POST['godt1']; ?>
                <select name="godt1" size="1">
                <option <?if($godt123==9): echo"SELECTED"; endif; ?> value="9">2009</option>
                <option <?if($godt123==10): echo"SELECTED"; endif; ?> value="10">2010</option>
                <option <?if($godt123==11): echo"SELECTED"; endif; ?> value="11">2011</option>
                </select>
                </td>
```

```
<td>
<b>Місяць з</b><br>
<? $montht=$_POST['montht']; ?>
<select name="montht" size="1">
<option <?if($montht==1): echo"SELECTED"; endif; ?> value="1">Січень</option>
<option <?if($montht==2): echo"SELECTED"; endif; ?> value="2">Лютий</option>
<option <?if($montht==3): echo"SELECTED"; endif; ?> value="3">Березень</option>
<option <?if($montht==4): echo"SELECTED"; endif; ?> value="4">Квітень</option>
<option <?if($montht==5): echo"SELECTED"; endif; ?> value="5">Травень</option>
<option <?if($montht==6): echo"SELECTED"; endif; ?> value="6">Червень</option>
<option <?if($montht==7): echo"SELECTED"; endif; ?> value="7">Липень</option>
<option <?if($montht==8): echo"SELECTED"; endif; ?> value="8">Серпень</option>
<option <?if($montht==9): echo"SELECTED"; endif; ?> value="9">Вересень</option>
<option <?if($montht==10): echo"SELECTED"; endif; ?> value="10">Жовтень</option>
<option <?if($montht==11): echo"SELECTED"; endif; ?> value="11">Листопад</option>
<option <?if($montht==12): echo"SELECTED"; endif; ?> value="12">Грудень</option>
</select>
</td>


<td>
<b>до</b><br><?$montht112=$_POST['montht1']; ?>
<select name="montht1" size="1">
<option <?if($montht112==1): echo"SELECTED"; endif; ?> value="1" >Січень</option>
<option <?if($montht112==2): echo"SELECTED"; endif; ?> value="2">Лютий</option>
<option <?if($montht112==3): echo"SELECTED"; endif; ?> value="3">Березень</option>
<option <?if($montht112==4): echo"SELECTED"; endif; ?> value="4">Квітень</option>
<option <?if($montht112==5): echo"SELECTED"; endif; ?> value="5">Травень</option>
<option <?if($montht112==6): echo"SELECTED"; endif; ?> value="6">Червень</option>
<option <?if($montht112==7): echo"SELECTED"; endif; ?> value="7">Липень</option>
<option <?if($montht112==8): echo"SELECTED"; endif; ?> value="8">Серпень</option>
<option <?if($montht112==9): echo"SELECTED"; endif; ?> value="9">Вересень</option>
<option <?if($montht112==10): echo"SELECTED"; endif; ?> value="10">Жовтень</option>
<option <?if($montht112==11): echo"SELECTED"; endif; ?> value="11">Листопад</option>
<option <?if($montht112==12): echo"SELECTED"; endif; ?> value="12">Грудень</option>
</select>
</td>


</tr>

<tr >
<td colspan="15" align="center" valign="bottom" >
<?
OutTableForGraph();
?>
<input type="submit" value="Відобразити">
<input type="hidden" value="1" name="Inp800">
</td>


</tr>

</table>
</form>

<br>
<?

?>
<table border="1">
<tr>
<th>
<font size="1">Місяць</font>
</th>

<th>
<font size="1">Номер<br>посту<br>спостереження</font>
</th>

<th>
<font size="1">Об'єкт<br>спостереження</font>
</th>

<th>
<font size="1">Зважена<br>речовина<br>мГ/Дм<sup>3</sup></font>
```

```
</th>
<th>
<font size="1">Кіслород<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Сульфати<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Хлоріди<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Кальцій<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">ХПК<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">БПК<sub>5</sub><br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Азот<br>амонійний<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Азот<br>нітрітний<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Азот<br>нитратний<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Фосфор<br>фосфатний<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Хром-6<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">СПАВ<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Фенол<br>мГ/Дм<sup>3</sup></font>
</th>
<th>
<font size="1">Нафто<br>продукти<br>мГ/Дм<sup>3</sup></font>
</th>
</tr>
<?
//Вывести данные с учетом сортировки
if(($_POST['Inp800']==1)&&($ID_ORG!='#')){

        $SQL="SELECT * FROM vodaOperativ WHERE IDORG='".$_POST['oo1']."'AND month<='".
                $_POST['montht1']."'AND month>='".$_POST['montht']."' AND god<='".
                $_POST['godt1']."' AND god>='".$_POST['godt']."' AND Number>='".$_POST['xz'].
                "' AND Number<='".$_POST['xz1']."';";
        $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
        $SUM=mysql_num_rows($SQL);
        if($SUM!=0){
                $i=1;global $DATAREQUEST;
                while($i<=$SUM):
                        $MIN[$i-1]=0;
                        $MAX[$i-1]=0;
                        $SUM1[$i-1]=0;
                        $i++;
                endwhile;
                $SUM=1;
                while($i=mysql_fetch_array($SQL)):
                        $DATAREQUEST[$SUM-1][0]=$i['P1'];
                        $DATAREQUEST[$SUM-1][1]=$i['P2'];
                        $DATAREQUEST[$SUM-1][2]=$i['P3'];
                        $DATAREQUEST[$SUM-1][3]=$i['P4'];
                        $DATAREQUEST[$SUM-1][4]=$i['P5'];
                        $DATAREQUEST[$SUM-1][5]=$i['P6'];
                        $DATAREQUEST[$SUM-1][6]=$i['P7'];
                        $DATAREQUEST[$SUM-1][7]=$i['P8'];
                        $DATAREQUEST[$SUM-1][8]=$i['P9'];
                        $DATAREQUEST[$SUM-1][9]=$i['P10'];
                        $DATAREQUEST[$SUM-1][10]=$i['P11'];
                        $DATAREQUEST[$SUM-1][11]=$i['P12'];
                        $DATAREQUEST[$SUM-1][12]=$i['P13'];
```

```
$DATAREQUEST[$SUM-1][13]=$i['P14'];
$DATAREQUEST[$SUM-1][14]=$i['P15'];
if($SUM==1){$MAX[0]=$i['P1'];$MIN[0]=$i['P1'];
            $MAX[1]=$i['P2'];$MIN[1]=$i['P2'];
            $MAX[2]=$i['P3'];$MIN[2]=$i['P3'];
            $MAX[3]=$i['P4'];$MIN[3]=$i['P4'];
            $MAX[4]=$i['P5'];$MIN[4]=$i['P5'];
            $MAX[5]=$i['P6'];$MIN[5]=$i['P6'];
            $MAX[6]=$i['P7'];$MIN[6]=$i['P7'];
            $MAX[7]=$i['P8'];$MIN[7]=$i['P8'];
            $MAX[8]=$i['P9'];$MIN[8]=$i['P9'];
            $MAX[9]=$i['P10'];$MIN[9]=$i['P10'];
            $MAX[10]=$i['P11'];$MIN[10]=$i['P11'];
            $MAX[11]=$i['P12'];$MIN[11]=$i['P12'];
            $MAX[12]=$i['P13'];$MIN[12]=$i['P13'];
            $MAX[13]=$i['P14'];$MIN[13]=$i['P14'];
            $MAX[14]=$i['P15'];$MIN[14]=$i['P15'];
};
if(round($i['P1'])>round($MAX[0])){$MAX[0]=round($i['P1']);};
if(round($i['P2'])>round($MAX[1])){$MAX[1]=round($i['P2']);};
if(round($i['P3'])>round($MAX[2])){$MAX[2]=round($i['P3']);};
if(round($i['P4'])>round($MAX[3])){$MAX[3]=round($i['P4']);};
if(round($i['P5'])>round($MAX[4])){$MAX[4]=round($i['P5']);};
if(round($i['P6'])>round($MAX[5])){$MAX[5]=round($i['P6']);};
if(round($i['P7'])>round($MAX[6])){$MAX[6]=round($i['P7']);};
if(round($i['P8'])>round($MAX[7])){$MAX[7]=round($i['P8']);};
if(round($i['P9'])>round($MAX[8])){$MAX[8]=round($i['P9']);};
if(round($i['P10'])>round($MAX[9])){$MAX[9]=round($i['P10']);};
if(round($i['P11'])>round($MAX[10])){$MAX[10]=round($i['P11']);};
if(round($i['P12'])>round($MAX[11])){$MAX[11]=round($i['P12']);};
if(round($i['P13'])>round($MAX[12])){$MAX[12]=round($i['P13']);};
if(round($i['P14'])>round($MAX[13])){$MAX[13]=round($i['P14']);};
if(round($i['P15'])>round($MAX[14])){$MAX[14]=round($i['P15']);};

if(round($i['P1'])<round($MIN[0])){$MIN[0]=round($i['P1']);};
if(round($i['P2'])<round($MIN[1])){$MIN[1]=round($i['P2']);};
if(round($i['P3'])<round($MIN[2])){$MIN[2]=round($i['P3']);};
if(round($i['P4'])<round($MIN[3])){$MIN[3]=round($i['P4']);};
if(round($i['P5'])<round($MIN[4])){$MIN[4]=round($i['P5']);};
if(round($i['P6'])<round($MIN[5])){$MIN[5]=round($i['P6']);};
if(round($i['P7'])<round($MIN[6])){$MIN[6]=round($i['P7']);};
if(round($i['P8'])<round($MIN[7])){$MIN[7]=round($i['P8']);};
if(round($i['P9'])<round($MIN[8])){$MIN[8]=round($i['P9']);};
if(round($i['P10'])<round($MIN[9])){$MIN[9]=round($i['P10']);};
if(round($i['P11'])<round($MIN[10])){$MIN[10]=round($i['P11']);};
if(round($i['P12'])<round($MIN[11])){$MIN[11]=round($i['P12']);};
if(round($i['P13'])<round($MIN[12])){$MIN[12]=round($i['P13']);};
if(round($i['P14'])<round($MIN[13])){$MIN[13]=round($i['P14']);};
if(round($i['P15'])<round($MIN[14])){$MIN[14]=round($i['P15']);};
$SUM1[0]=$SUM1[0]+$i['P1'];
$SUM1[1]=$SUM1[1]+$i['P2'];
$SUM1[2]=$SUM1[2]+$i['P3'];
$SUM1[3]=$SUM1[3]+$i['P4'];
$SUM1[4]=$SUM1[4]+$i['P5'];
$SUM1[5]=$SUM1[5]+$i['P6'];
$SUM1[6]=$SUM1[6]+$i['P7'];
$SUM1[7]=$SUM1[7]+$i['P8'];
$SUM1[8]=$SUM1[8]+$i['P9'];
$SUM1[9]=$SUM1[9]+$i['P10'];
$SUM1[10]=$SUM1[10]+$i['P11'];
$SUM1[11]=$SUM1[11]+$i['P12'];
$SUM1[12]=$SUM1[12]+$i['P13'];
$SUM1[13]=$SUM1[13]+$i['P14'];
$SUM1[14]=$SUM1[14]+$i['P15'];
$SUM++;
?>
<tr>
<td>
<font size="2" >      <?$sss=numberMonth($i['month']);echo $sss;?></font>
</td>
<td>
<font size="2" >      <?echo $i['Number'];?></font>
</td>
<td>
<font size="2" >      <?echo $i['Tip'];?></font>
</td>
<td>
```

```
                <font size="2" >      <?echo $i['P1'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P2'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P3'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P4'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P5'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P6'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P7'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P8'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P9'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P10'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P11'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P12'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P13'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P14'];?></font>
                </td>
                <td>

                <font size="2">       <?echo $i['P15'];?></font>
                </td>
                </tr>

                <?
endwhile;
?>
</table>

<br>
<table border="1"  id="max2" class="hideT" >
<tr >
<td colspan="100" align="center" >
<font size="2"><b>Максимум</b></font>
</td>
</tr>
<tr >
<?
$i=0;
while($i<15):
                ?>
```

```
                <td>
                <font size="2">
                <? echo $MAX[$i];?>
                </font>
                </td>
                <?
                $i++;
endwhile;
?>
</tr>
</table>
<br>
<table border="1"id="max3" class="hideT">
<tr>
<td colspan="20" align="center">
<font size="2"><b>Мінімум</b></font>
</td>
</tr>
<tr>
<?
$i=0;
while($i<15):
                ?>
                                <td>
                                <font size="2">
                <?echo $MIN[$i]; ?>
                </font>
                </td>
                <?
                $i++;
endwhile;
?>

</tr>

</table>
<table border="1" id="max4" class="hideT">

<tr>
<td colspan="20" align="center">
<font size="2"><b>Середнє значення</b></font>
</td>
</tr>
<tr>
<?
$i=0;
while($i<15):
                ?>
                <td>
                <font size="2">
                <? $SUM1[$i]=(double)$SUM1[$i]/($SUM-1); $SUM1[$i]=round($SUM1[$i],3);
                echo $SUM1[$i];
                ?>
                </font>
                </td>
                <?
                $i++;
endwhile;
?>
</tr>

</table>
<table border="1" id="max5" class="hideT">
<tr>
<td colspan="20" align="center">
<font size="2"><b>Розкид</b></font>
</td>
</tr>
<tr>
<?
$i=0;
while($i<15):
                ?>
                <td>
                <font size="2">
                <?  echo $MAX[$i]-$MIN[$i];?>
                </font>
```

```
                                        </td>
                                        <?
                                        $i++;
                                endwhile;
                                ?>

                                </tr>

                                </table>
                                <!---
                                Настройки
                                //-->
                                <table border="1">
                                <tr>
                                <td>
                                <input type="checkbox"  id="set1" value="max2" OnClick="javascript:ch1(this.id);"><font size="1"><b>Відобразити<br>максимум</b></font>

                                </td>

                                <td>
                                <input type="checkbox" id="set2" value="max3" OnClick="javascript:ch1(this.id);" ><font size="1"><b>Відобразити<br>мінімум</b></font>

                                </td>

                                <td>
                                <input type="checkbox" id="set3" value="max4" OnClick="javascript:ch1(this.id);" ><font size="1"><b>Відобразити<br>середнє<br>значення</b></font>

                                </td>

                                <td>
                                <input type="checkbox" id="set4" value="max5" OnClick="javascript:ch1(this.id);" ><font size="1"><b>Відобразити<br>розкид</b></font>

                                </td>
                                </tr>
                                </table>

                                <?
                                global $MAX_R; global $MIN_R; global $SUM_R;
                                $MAX_R=$MAX;$MIN_R=$MIN;$SUM_R=$SUM1;
                        return 1;}else{return 0;};
                }else{  return 0;};
        };
        function GetGraph($First,$Number){
                global $MAX_R;
                global $DATAREQUEST;

                $ikl=round(300/count($DATAREQUEST));
                ?>
                <table border="1">
                <tr>
                <td style="border: 1px solid black;" width="350" height="400" align="left" valign="bottom" colspan="20">
                <div align="center"><b><font size="2">Зображений колір відповідає коліру назви речовини<br> що відображено наступного рядку</font></b></div>
                <?

                $D=0;
                for($j=$First-1;$j<=$Number-1;$j++){
                        if($D<round($MAX_R[$j])){$D=round($MAX_R[$j]);};
                };

                $D=round(300/$D);
                $color[0]="red";
                $color[1]="blue";
                $color[2]="brown";
                $color[3]="burlywood";
                $color[4]="magenta";
                $color[5]="navy";
                $color[6]="mediumseagreen";
                $color[7]="aqua";
                $color[8]="lightsalmon";
                $color[9]="limegreen";
                $color[10]="crimson";
                $color[11]="darkolivegreen";
```

```
$color[12]="plum";
$color[13]="lightcoral";
$color[14]="violet";
DrowGraphLine(0,0,0,-330,0,"black",3,1,-50,10,"","obj1"); //вертикальная линия
DrowGraphLine(0,-330,-3,-323,5,"black",3,1,300,0,"","obj1");
DrowGraphLine(0,-330,3,-323,5,"black",3,1,300,0,"","obj1");
DrowGraphLine(0,0,300,0,5,"black",3,1,300,0,"","obj1"); //горизонтальная линия
DrowGraphLine(300,0,293,3,5,"black",3,1,300,0,"","obj1");
DrowGraphLine(300,0,293,-3,5,"black",3,1,300,0,"","obj1");
for($j=1;$j<10;$j++){
        DrowGraphLine(0,-30*$j,290,-30*$j,5,"black","Stroke.DOTTED",1,300,0,"","obj1");
        DrowGraphLine(30*$j,0,30*$j,-300,5,"black","Stroke.DOTTED",1,300,0,"","obj1");
};
$i=0;
$j=$First-1;
$loc=false;

while($j<$Number):
        while($i<count($DATAREQUEST)):
                if($i==0){
                        DrowGraphLine(0,0,($i+1)*$ikl,-
$DATAREQUEST[$i][$j]*$D,5,$color[$j],2,2,300,0,"","obj1");
                        $i++;continue;
                };
                if(($i+1==count($DATAREQUEST)&&($j+1==$Number))){
                        DrowGraphLine($i*$ikl,-$DATAREQUEST[$i-1][$j]*$D,($i+1)*$ikl,-
$DATAREQUEST[$i][$j]*$D,1,$color[$j],2,2,50,100,"","obj1");
                        $loc=true;
                }else{
                DrowGraphLine($i*$ikl,-$DATAREQUEST[$i-1][$j]*$D,($i+1)*$ikl,-
$DATAREQUEST[$i][$j]*$D,5,$color[$j],2,2,-50,-100,"","obj1");          };
                $i++;
        endwhile;
        $j++;
        $i=0;
endwhile;
if($loc==false){
        ?>
        //-->
        </script>
        <?
};

?>

</td>
</tr>


<tr>
<?
$i=$First-1;$loc=false;
while($i<=$Number-1):
        if($i%5==0){
                if($loc==true){$loc=false;
                        ?>
                        <tr>
                        <?
                }else{$loc=true;
                        ?>
                        </tr>
                        <?
                        continue;
                };
        };

        ?>
        <td valign="bottom" align="center">
        <font color="<?echo $color[$i];?>"><b>
        <?
        echo Titul($i);
        ?>
        </b></font>
        </td>
        <?
        $i++;
endwhile;
```

```
		if($loc==false){?> </tr> <?};
		?>
		<tr>
		<td colspan="20" align="center" valign="middle">
		<font size="2"><b>
		<?
		$CD=round(30/$D,2);
		echo "В одному діленні ".$CD." одиниць";
		?>
		</b></font>
		</td>

		</tr>
		</table>
		<?
		return true;
};

?>
```

analization1.php

```
<?php
//Экологический специализированный анализ относящийся к анализу водных ресурсов
function Titul($number){
		switch(true){
			case($number==0):
				return "<font size=\"1\">Зважена<br>речовина<br>мГ/Дм<sup>3</sup></font>";
			break;
			case($number==1):
				return "<font size=\"1\">Кіслород<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==2):
				return "<font size=\"1\">Сульфати<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==3):
				return "<font size=\"1\">Хлоріди<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==4):
				return "<font size=\"1\">Кальцій<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==5):
				return "<font size=\"1\">ХПК<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==6):
				return "<font size=\"1\">БПК<sub>5</sub><br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==7):
				return "<font size=\"1\">Азот<br>амонійний<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==8):
				return "<font size=\"1\">Азот<br>нітрітний<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==9):
				return "<font size=\"1\">Азот<br>нітратний<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==10):
				return "<font size=\"1\">Фосфор<br>фосфатний<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==11):
				return "<font size=\"1\">Хром-6<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==12):
				return "<font size=\"1\">СПАВ<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==13):
				return "<font size=\"1\">Фенол<br>мГ/Дм<sup>3</sup></font>";
				break;
			case($number==14):
				return "<font size=\"1\">Нафто<br>продукти<br>мГ/Дм<sup>3</sup></font>";
				break;
			default:return null;
			};
		};

?>
```

map.htm

```
<HTML>
	<HEAD>

		<TITLE></TITLE>
	</HEAD>
	<BODY>
	<img src="/images/uch.gif" />

	</BODY>
</HTML>
```

stop.php

```
<b>ОФІЦІЙНИЙ САЙТ <br>
СИСТЕМИ СЄВЄРОДОНЕЦЬКОГО МІСЬКОГО ЕКОЛОГІЧНОГО МОНІТОРІНГУ<br></b><br>
<b>Сєвєродонецьк</b> - місто обласного значення в Луганській області України.<br>
Розташований на лівому березі річки Сіверський Донець <br>
при впаданні в неї річки Борової. Відстань до Луганська - 110 км. <br>
Найважливіший промисловий центр Донбасу. <br><br><br>

<b>НОВИНИ ЩОДО ЕКОЛОГІЇ СЄВЕРОДОНЕЦЬКА ТА ДЕРЖАВИ ЗАГАЛОМ </b><br>
<b>Стрічка новин</b>
<br>
<br>
<div align="left">
<?
$aCC="<b>2009-05-12</b>
		Донецька область завалена промисловими відходами

		За 2008 рік на підприємствах і в організаціях Донецької області утворилося 37,5 млн. тонн різних виробничих і побутових відходів. В порівнянні з попереднім роком це менше на 5%, що пов'язане із скороченням обсягів промисло вого виробництва. В той же час із-за погіршення якості вугілля відходів вуглевидобування і вуглезбагачення утворилося на 8% більше, ніж рік тому, - 14,3 млн. тонн, або 38% всіх відходів. Значні об'єми склали також шлаки доменного виробництва - 6,8 млн. тонн, вапняні і вапняно-магнієві відходи - 3,8 млн. тонн, зола і золошлаковые відходи теплових електростанцій - 3,2 млн. тонн, стільки ж - сталеплавильні шлаки, залізовмісні відходи - 2,7 млн. тонн.

		Як повідомляє Головне управління статистики в Донецькій області, низьким залишається рівень використання виробничих відходів. В цілому по області в 2008 році залучено в господарський зворот (для дорожнього будівництва, виробництва будівельних матеріалів і тому подібне) 10,5 млн. тонн відходів, що менше попереднього року на 9% і склало 28% об'єму, що утворився за рік.

		Якщо рівень використання шлаків доменного і сталеплавильного виробництва склав 59%, а залізовмісних відходів металургійного виробництва - 100%, то відходів вуглевидобування і вуглезбагачення було перероблено що лише 5% утворилися за рік і стільки ж - золи і золошлаковых відходів електростанцій.

		Низький рівень використання відходів веде до збільшення їх об'ємів у відвалах і на звалищах. На початок 2009 року в регіоні накопичилися 794 млн. тонн відходів, з яких понад 70% доводиться на відходи вуглевидобування і вуглезбагачення, дев'ята частина - на вапняних і вапняно-магнієві відходи, десята частина - на золу і золошлаковые відходи, що служить постійним джерелом забруднення землі, води і повітряного простору. В середньому на один квадратний кілометр території області доводиться майже 30 тис. тонн різних відходів, з розрахунку на одного жителя - по 176 тонн. Під відходами у відвалах і накопичувачах зайнято більше 7 тис. гектарів земельних угідь. Річні витрати на зберігання і знищення відходів виробництва склали 157 млн. грн.
		";
$INF1= newinf();
if($_GET['info']!=1){
	$INF1-> Init(4,150,"black",100,1,"NOT",$aCC,"<a href=\"javascript:Leave_Data_Delete('0','0','~','~','1','1','info','stor','info1','info2','info3');true\" id=\"knn11\"class=\"offDecoration\" onmousemove=\"javascript:mouseMovel(this.id)\"
		onmouseout=\"javascript:mouseOutl(this.id)\"><font size=\"3\"><b>Донецька область завалена промисловими відходами</b></font></a>","","~');
	$INF1->show();
	?>

	<a href="javascript:Leave_Data_Delete('0','0','~','~','1','1','info','stor','info1','info2','info3');true" id="knn1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
		onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Докладніше</b></font></a><br>
	<br><br><?
}else{
	$INF1-> Init(4,150,"black",100,3,"NOT",$aCC,"<b>Донецька область завалена промисловими відходами</b>","","~');

	$INF1->show();
	?>
	<a href="javascript:Leave_Data_Delete('0','0','~','~','2','~','~','info');true" id="knn1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
		onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Сгорнути</b></font></a><hr color="darkblue">
	<br><?
```

};

$aCC="У Білорусії, на Україні і в Росії 26 квітня відзначали річницю аварії на ЧАЕС і згадували її жертви. Президент України Віктор Ющенко взяв участь в церемонії покладання кольорів до меморіального кургану «Героям Чорнобиля» в Києві. Білоруський президент Олександр Лукашенко відправився в традиційну поїздку по чорнобильських районах, а білоруська опозиція пройшла «Чорнобильським маршем» по вулицях Мінська - самим нечисленним за 20 років його історії.

У Києві церемонія пам'яті пройшла в ніч на 26 квітня. Після покладання кольорів Ющенком взяв участь в молебні по жертвах Чорнобильської катастрофи. Глава України і голова Верховної ради поставили по дві свічки - на згадку про загиблих в результаті катастрофи на Чорнобильській АЕС і за здравіє тих, КТО брав участь в ліквідації наслідків аварії.

Президент Білорусії традиційно перед річницею аварії вирушає в Чорнобильську зону. «Відроджувати цей край ми будемо, чого б це нам не коштувало. Ми відновлюватимемо країну в цій її частині, а не стогнати і плакати», - заявив Лукашенко 25 квітня в Гомелській області.

«Я вірю людям і вважаюся на них, адже вони краще за інших знають, де можна розробляти землю, а де немає», - додав глава Білорусії. За його словами, «якщо ви бачите, де можна відкривати землю, - відкривайте». При цьому Лукашенко підкреслив необхідність «підсилити контроль над продовольством, яке тут проводиться», наводить його слова «Інтерфакс-захід».

Висловився білоруський президент і про опозиційний «Чорнобильському шляхе». «Завтра вони хочуть маршем, що фашиствує, бути схожим по вулиці, продемонструвати. Йди, демонструй в зону», - цитує Лукашенко ІА «БЕЛАПАН».

Втім, це 26 квітня для білоруського президента декілька незвичайне. Він вже повернувся з поїздки по чорнобильських районах, щоб увечері вилетіти до Риму і Ватикану.

Вчора вдень в Мінську пройшов тригодинний «Чорнобильський шлях» - традиційна акція білоруської опозиції, приурочена до річниці аварії на ЧАЕС. Вже заявляється, що шлях був самим нечисленним за всю двадцятирічну історію - кількість учасників оцінюється, правда, по-різному: від 200 чоловік (МВС) до 1000 (самі організатори) або декількох тисяч (деякі спостерігачі). Найчисленнішим був «Чорнобильський шлях» в 1996 році, коли на вулиці Мінська вийшли близько 30 тисяч чоловік.

Лідер Партії БНФ Льовон Борщевський називає «минулу акцію акцією спаду». Політик заявив про готовність узяти на себе відповідальність «за ті недоробки, які сталися під час підготовки до «Чорнобильського шляху», розділивши її з політиками, що знаходяться у від'їзді».

Учасники шляха намагалися пройти несанкціонованим маршрутом - від місця збору в центр міста і тільки потім до храму-пам'ятника ікони Божої Матері «Стягнення загиблих», де традиційно закінчується «Чорнобильський шлях». На проспекті Незалежності прохід заблокував спецназ, і після півгодинних безрезультатних переговорів опозиція обернула по санкціонованому маршруту.

Про затримання в ході акції нічого не відомо, як і про зіткнення з міліцією. ІА «БЕЛАПАН» повідомляє про блокування роботи опозиційного сайту «Хартія 97» - міліціонери увійшли до квартири, щоб провести розшук велосипеда, який вкрали у сусідів, і знаходилися там декілька годин.

Крім того, повідомляється про превентивне затримання активіста незареєстрованої партії «Білоруська християнська демократія» Дениса Садовського, молоду людину визволили через декілька годинників без складання протоколу.

У православних і католицьких храмах Білорусії сьогодні моляться про загиблих в результаті аварії на Чорнобильській АЕС. «Ця молитва на згадку про національну трагедію традиційно здійснюється у всіх храмах Білоруської православної церкви вже протягом більше 20 років», - повідомили в Мінському єпархіальному управлінні Білоруської православної церкви «Інтерфакс-захід».

У 1986 році вночі 26 квітня в 1.23 мск на 4-му енергоблоці ЧАЕС в результаті серії теплових вибухів був зруйнований реактор. По розрахунках експертів, сумарний вихід радіоактивних матеріалів склав 50 млн кюрі, що рівнозначно наслідкам вибухів 500 атомних бомб, скинутих в 1945 році на Хіросіму.

Радіоактивна хмара пройшла над європейською частиною СРСР (Україна, Білорусія, Росія), Східною Європою, Скандинавією, Великобританією і східною частиною США. Радіоактивними нуклідами було забруднено більше 145 тис. кв. км. території України, Білорусії і Росії - близько 5 тис. населених пунктів. У Росії найбільш забрудненими територіями є Брянська область (11800 кв. км.), Тульська (11600 кв. км.), Калузька (4900 кв. км.) і Орловська області (8900 кв. км.).

Аварія стала найбільшою техногенно-екологічною катастрофою сучасності.

У 1986-1987 роках в роботах по ліквідації наслідків катастрофи взяли участь близько 600 тис. чоловік, 200 тис. з них отримали підвищені дози опромінення.

Сьогодні МНС Росії заявляє про поліпшення радіаційної обстановки в російських регіонах, постраждавших від чорнобильської аварії. «Радіаційна обстановка на територіях, постраждавших в результаті катастрофи на Чорнобильській АЕС, помітно покращується в результаті проведення захисних заходів і через природний розпад радіоактивних елементів», - приводить повідомлення МНС Росії «Інтерфакс».

У свою чергу екологи заявляють, що реальні наслідки чорнобильської аварії до цих пір замовчуються. «Принаймні, ще 150 років наслідку цієї аварії відчуватимуться. Це сім поколінь», - сказав лідер фракції «Зелена Росія» демократичної партії «Яблуко» еколог Олексій Яблоков.

```
                                        Він стверджує, що наслідки чорнобильської аварії офіційні органи замовчують. «Реальні наукові дані не публікують. Насправді реальна картина дуже несприятлива. Наслідки набагато страшніші, ніж говорять атомники. Вони закликають забути Чорнобиль. Забути Чорнобиль не можна», - уклав Яблоков.
                                        ";
        $INF1= newinf();
        if($_GET['info1']!=1){
                $INF1-> Init(4,264,"black",100,1,"NOT",$aCC,"<a href=\"javascript:Leave_Data_Delete('0','0','~','~','1','1','info1','stor','info','info2','info3');true\" id=\"knn21\"class=\"offDecoration\" onmousemove=\"javascript:mouseMovel(this.id)\"
                                                onmouseout=\"javascript:mouseOutl(this.id)\"><font size=\"3\"><b>В неділю виконалося 23 року з дня аварії на Чорнобильській АЕС</b></font></a>","",'~');
                $INF1->show();
                ?>

        <a href="javascript:Leave_Data_Delete('0','0','~','~','1','1','info1','stor','info','info2','info3');true" id="knn2"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Докладніше</b></font></a><br>
                <br><br><?
        }else{
                $INF1-> Init(4,254,"black",100,3,"NOT",$aCC,"<b>В неділю виконалося 23 року з дня аварії на Чорнобильській AEC</b>","",'~');
                $INF1->show();
                ?>
                <a href="javascript:Leave_Data_Delete('0','0','~','~','2','~','~','info1');true" id="knn2"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Сгорнути</b></font></a><hr color="darkblue">
                <br><?
        };
        $aCC1="
                                Циолковського на рівні четвертого поверху на дереві вже
                                декілька днів знаходиться котеня. На що Мчсники відповіли, що  кішок з
                                дерев не знімають. Тоді смілива дівчинка залізла на дерево, узяла
                                котеняти і від туди подзвонила Мчсникам, щоб її зняли.


                                Пожежники були в Шоке від такого вчинку 11-річній дівчинки, яка
                                дзвонила з кішкою з висоти приблизно четвертого поверху.


                                Із звіту МНС: \"К місцю виклику були направлені рятувальники на пожежній
                                автосходам. Після прибуття в 11:25 було встановлено, що на висоті 15
                                метрів на дереві знаходиться учениця ОШ №3 6-\"А\" класу, яка не може
                                самостійно злізти з дерева.


                                За допомогою спецтехніки рятувальник допоміг дівчинці перебратися на сходи і
                                підтримуючи її, спустився разом з нею вниз. Після цього на висоту 20 м-кодів
                                висунули сходи. По ній рятувальник піднявся і зняв кішку з дерева
                                <br>
                                <img src=\"images/exm21.jpg\" >
                                ";
        $aCC="8 квітня в 11:00 на пульт зв'язку Константіновського міського відділу МНС
                                від школярки вчинило незвичайне повідомлення про те, що біля п'ятиповерхового
                                удома по вулиці
                                ";
                                $INF1= newinf();
        if($_GET['info2']!=1){
                $INF1-> Init(4,138,"black",100,1,"NOT",$aCC,"<a href=\"javascript:Leave_Data_Delete('0','0','~','~','1','1','info2','stor','info','info1','info3');true\" id=\"knn31\"class=\"offDecoration\" onmousemove=\"javascript:mouseMovel(this.id)\"
                                                onmouseout=\"javascript:mouseOutl(this.id)\"><font size=\"3\"><b>Учениця школи № 3 зробила зелений подвиг</b></font></a>","",'~');
                $INF1->show();
                ?>

        <a href="javascript:Leave_Data_Delete('0','0','~','~','1','1','info2','stor','info','info1','info3');true" id="knn3"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Докладніше</b></font></a><br>
                <br><br><?
        }else{
                $INF1-> Init(4,150,"black",100,3,"NOT",$aCC1,"<b>Учениця школи № 3 зробила зелений подвиг</b>","",'~');
                $INF1->show();
                ?>
                <a href="javascript:Leave_Data_Delete('0','0','~','~','2','~','~','info2');true" id="knn3"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
```

```
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Сгорнути</b></font></a><hr
color="darkblue">
                <br><?
        };
        $aCC="Шановні колеги,

                                        НЕК \"Укренерго\" планує побудову 2-ох нових ділянок високовольтних ліній електропередач у
Одеській обл., та Запорізькій і Херсонській областях (від<br>
                                        Запорізької АЕС!).<br>

                                        Фінансування проекту передбачається здійснити за рахунок кредитних коштів<br>
                                        ЄБРР, що вимагає від Укренерго дотримання вимог банку, зокрема це стосується<br>
                                        проведення справжніх консультацій з громадськістю на стадії підготовки<br>
                                        Оцінки впливу проекту на навколишнє середовище.<br>

                                        Перший раунд консультацій буде проведено вже у квітні-травні, вчора ми<br>
                                        отримали відповідні запрошення, а також ця інформація є на сайті Укренерго.<br>
        (див. запрошення у додатку).

                                        Представиники НЕЦУ обовязково візьмуть участь у консультаціях у Києві.<br>
                                        Але ми також просимо всі зацікавлені НУО, в першу чергу регіональні, взяти<br>
        участь у консультаціях на місцях. У запрошеннях є контакти відповідальних осіб, до яких треба звертатися за деталями та
запитувати матеріали до
        слухань.
                                        Графік консультацій:<br>

                                        ЛЕП Енергодар-Каховка<br>
                                        27 квітня 2009р.  Запоріжжя, вул. Гребельна, 2, о 14.00<br>
                                        29 квітня 2009р.  Нова Каховка, проспект Дніпровський, 23, о 14.00<br>
                                        13 травня 2009 р.  Київ, офіс ЄБРР, вул. Софіївська, 27/23, о 14.30<br>
                                        ЛЕП Новоодеська-Арциз<br>
                                        27 квітня 2009 р.  - Одеса, вул. Коблівська, 11, о 14.00<br>
                                        29 квітня 2009 р. - Арциз, вул. Орджонікідзе, 46, о 11.00<br>
                                        13 травня 2009 р. - Київ, офіс ЄБРР, вул. Софіївська, 27/23, о 10.30<br>
                                        Будь ласка, також розповсюдьте ці запрошення серед ваших колег, особливо у<br>
                                        областях, де будівництва плануються.<br>

                                        з повагою, Ірина Головко<br>
         Національний екологічний центр України";
        $INF1= new inf();
        if($_GET['info3']!=1){
                $INF1-> Init(4,141,"black",100,1,"NOT",$aCC,"<a
href=\"javascript:Leave_Data_Delete('0','0','~','~','1','1','info3','stor','info1','info2','info');true\" id=\"knn41\"class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                                                onmouseout=\"javascript:mouseOutl(this.id)\"><font size=\"3\"><b>Укрэнерго
запрошує на громадські слухання</b></font></a>","","~');
                $INF1->show();
                ?>

        <a href="javascript:Leave_Data_Delete('0','0','~','~','1','1','info3','stor','info1','info2','info');true" id="knn4"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Докладніше</b></font></a><br>
                <br><br><?
        }else{
                $INF1-> Init(4,149,"black",100,3,"NOT",$aCC,"<b>Укрэнерго запрошує на громадські слухання</b>","","~');
                $INF1->show();
                ?>
                <a href="javascript:Leave_Data_Delete('0','0','~','~','2','~','~','info3');true" id="knn4"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Сгорнути</b></font></a><hr
color="darkblue">
                <br><?
        };
        ?>
        </div>
        stor1.php
        <b>ЕКОЛОГІЯ МІСТА ТА ЗАБРУДНЕННЯ ПОВІТРЯ</b>
        <br>
        <br>

        <div align="left">
        <img src="/images/image.jpg">
        </div>

        stor2.php
        <b>ЕКОЛОГІЯ МІСТА ТА ЗАБРУДНЕННЯ ВОДИ</b>
        <br>
```


stor3.php
ЕКОЛОГІЯ МІСТА ТА ЗАБРУДНЕННЯ ЗЕМЛІ

<?
ДОПОВІДІ ТА СЕМІНАРИ ЩОДО ЕКОЛОГІЇ МІСТА

Одними з найбільш розповсюджених у промисловості
сильнодіючих отруйних речовин є хлор, що використовується для
очищення води на усіх водопровідних станціях (ВС) міста.
Наприклад, на Дніпровській та Деснянській ВС м. Києва п
остійно зберігається до 100 т рідкого хлору, що при техногенній
катастрофі може викликати хімічне зараження великої території
міста, а при локальних аваріях з викидом хлору на цих ВС можуть
постраждати мешканці, прилеглих до ВС районів. Тому перша черга
реалізації ПОКК-4 запланована саме на ВС міста Києва.

<?
$aCC="Все те, що нас оточує; елементи живої і
неживої природи (гірські породи, грунти, рослинний і тваринний мир, річки
озера, морить, повітря, Сонце). Все в природі знаходиться у взаємозв'язку і
взаємозалежності. Ніщо само по собі не існує, починаючи від потужних
дерев і крупних тварин до невидимих неозброєним оком
мікроорганізмів. Адже не може будь-яка рослина жити без грунту, вологаі
необхідної кількості тепла. Кожна рослина вимагає певні умови
для свого життя. Дуб любить простір і світло. Сосна теж любить світло, але не
вимагає багато вологи. Вона добре зростає і на пісках, а ялини потрібні знижені
добре зволожені місця. У сосновому лісі шалено розвивається підлісок
(трава, чагарник), бо в такому лісі ясно. А в ялиновому? Там похмурий
і сиро, сильно поширені мохи і лишайники. Кожному з вас зрозуміло, що
полин не зростатиме у воді, як очерет і очерет в жаркій і посушливій
степи. Коли будете на екскурсії в лісі, то звернете увагу на те, де
зростає мох на стволах дерев. Лише на північній стороні. Він не любить яскравий
світло сонця. Йому не потрібно багато світла і тепло. (Це один із способів
орієнтування що заблукав в лісі).
А чи існує взаємозв'язок і взаємозалежність між рослинами і
тваринами? Звичайно, існує. Всі тварини залежать від рослин. Є пряма
залежність між рослинами і травоїдними тваринами, як домашніми (кінь
корова, вівця, коза, свиня), так і дикими (кабан, косуля, олень, лось). Адже
рослинність для них - основна їжа, без якої вони існувати не
можуть. Така ж залежність і між рослинами і рослиноїдними домашніми
і дикими птицями.
А як же хижаки? Вони безпосередньо від рослин не залежать. Але якщо не буде
косулі, зайця, кабана, адже то вовк і лисиця загинуть. І тут щонайтісніша
взаємозалежність. У природі існує певна рівновага між
хижаками і травоїдними тваринами. Знищення всіх хижаків не завжди
приводить до добрих результатів, оскільки вони знищують слабких і хворих, а
сильні, спритні і здорові залишаються для потомства. Можна побудувати таку
екологічну піраміду: дубовий ліс, миші і сови. Дуб розмножується насінням
(жолуді), жолудями харчуються миші, сови - лісовими мишами. Не буде дуба -
нічим харчуватися мишам, вони загинуть. Тоді і сова залишиться без їжі. Є
сова, вона знищує частину мишей, деяка кількість жолудів залишається
зростають дублення. Ось такий щонайтісніший взаємозв'язок і взаємозалежність
існує в природі, між її складовими частинами.
Природа, людина і суспільство - нерозривне ціле. Люди не можуть
існувати на землі без природи, що оточує їх. Вона дає їм пищу, одяг
взуття, світло, кисень для дихання, воду, сонячне тепло.
Щорік, 5-го червня, наголошується Усесвітній день охорони довкілля. Цей
день встановлений вирішенням Організації Об'єднаних Націй (ООН) в 1972 році для
мобілізації зусиль всіх народів країн світу на захист природи і її ресурсів
зміцнення міжнародних контактів в області охорони довкілля.
Основу розвитку кожної держави складають його природні корисні
копалини, які, на жаль, не відновлюються. Тому використовувати
їх треба економно.
Все, або майже все, чим займається людство на землі при вирішенні питань
свого життєзабезпечення, шкідливо впливає на місце існування людини.
Виробництво продуктів харчування, промислових товарів, видобуток корисних

копалин, а також сільське господарство, енергетика, транспорт і багато інших напрями діяльності людини приносять непоправну шкоду тієї, що оточує середі.

Одним із забрудників повітряної середи, що постійно діють, на сьогоднішній день є автомобільний транспорт. Автомашини викидають в атмосферу щорік тисячі тонн шкідливих речовин, небезпечних для здоров'я людини. Джерела забруднення повітря - промислові підприємства, об'єкти жилищно-комунального господарства. Найбільшої шкоди завдають хімічні машинобудівні, цукрові заводи. На багатьох з них діє застаріле устаткування. Технологія очищення відходів від шкідливих домішок морально і фізично застаріла або взагалі відсутній. Неблагополучна обстановка по охороні повітряного басейну склалася в Краснодарі і Армавірі, Новоросійську і Туапсе. Лише промислові підприємства цих міст викидають в атмосферу більше 180 тисяч тонн шкідливих речовин в рік. Новоросійські цементні заводи постійно забруднюють довкілля шкідливими речовинами і цементним пилом.

Забруднюють повітря щорічні масові спалювання стерні. Від вогню і диму гинуть звіри, птиці, вигоряє верхній родючий шар грунту. Кубанські учені стверджують, що цей агроприем (спалювання стерні) не приводить до поліпшення фітосанітарного достатку полів. Якщо солому не спалювати, а заорювати використовуючи як органічне добриво, можна підвищити родючість грунту і добитися надбавки наступного урожаю на 15-20%.

Природна середа забруднюється і пестицидами. Інтенсивне землеробство порушило екологічна рівновага в природі. Були знищені корисні види флори і фауни. Майже зникло з наших полів і садів сонечко, адже саме вона знищує личинки колорадського жука. Під загрозою зникнення знаходиться багато хто співтовариства і види. У Червону книгу краю занесено 157 видів рослин і 100 тварин.

Із зменшенням вживання отрутохімікатів чисельність звірів і птиць стала поступово зростати. Збільшилася кількість зайця-русака, лисиці червоною, єнота.

Вода - основа життя. Запаси її на перший погляд, чималі. На території краї більше 13 тисяч річок, джерел і інших водних об'єктів. Середній річний стік головної водної артерії - річки Кубані - в міста Краснодару (з обліком відбору води Ставропольським краєм) складає 13 кубокілометров. Та все ж в останнім часом частіше і частіше стикаємося з проблемою браку прісної води, до тому ж знижується її якість.

Поряд з Кубанню, значну роль грають малі річки степової зони. Їх спільна протяжність складає майже 14 тисяч кілометрів. Вони обводнюють більше 50% території. Тут є 80 тисяч гектарів зрошуваних земель.

Характеризуючи їх сучасний достаток, екологічну, що склалася обстановку, можна без перебільшення сказати, що ці річки по суті сталі ганебним пам'ятником нашого безрозсудного відношення до природи. Колись повноводні і навіть місцями судноплавні, вони перетворилися на ланцюжки стоячих водоймищ. На малих річках зведено близько 1500 гребель, запруд і всіляких гатей, багато хто з яких побудований без проектів. Відкриття землі до самого урізання води, знищення чагарникової рослинності і лісу, будівництво різних об'єктів по берегах привели до їх замулювання і забруднення. Ці водоймища практично стали потужними випарниками вологи, що веде до швидкої мінералізації води і непридатності її на зрошування. Середня глибина водоймищ не перевищує півтора метрів.

А скільки забруднень потрапляє в наші річки! Стоки з ферм, отрутохімікати з полів. По берегах річок викидається сміття, а з кожного житлового масиву розташованого біля річки, скидаються і каналізаційні стоки з будинків. Багато власники автомашин, не маючи мийних майданчиків, миють автомобілі у випадкових місцях, забруднюючи грунт.

Хай кожен житель району посадить хоч би одне дерево і пам'ятає, що викидання сміття, спалювання його не лише естетично погіршує обстановку але і впливає на здоров'ї - його власне, дітей, що оточують.

Браконьєрить на річках і морях, розорення дітьми пташиних гнізд мурашників, надмірний збір кольорів в лісах і на полях, жорстоке відношення дорослих і дітей до домашніх тварин, безпричинне ламання віток, всі ці безрозсудні дії людей привели до збіднення природних багатств, порушення законів екології.

Серед окремих видів транспорту, ймовірно, найбільший вклад в забруднення ОС вносить автомобільний. І хоча його дія як забрудника локалізовано на вулицях міст і крупних сільських поселень, а також \"канализировано\" 250-300-метровій смугою уподовж найбільш грузо-пассажиронапряженных магістралей в сільській місцевості, природі і чоловічкові його забрудники наносять істотна шкода. Для багатьох міст краю автомобіль - основне джерело забруднення ОС.

У 1989 р., наприклад, в м. Сочі автотранспорту припадало на частку 90,4 % сумарного валового викиду забрудників, в Анапі - 89,8'%, в Лабінське - 86,0 %, Белореченське - 66,5 %, ' у Краснодарі - 65,4 % і так далі Головні причини: недолік, а деколи повна відсутність екологічно чистих видів палива, висока токсичність продуктів згорання у зв'язку з особливим якістю автомобільних бензинів я низькою екологічністю двигунів незадовільний достаток доріг і низька культура обслуговування автомобіля

наднормативні терміни експлуатації автомобілів. Залізничний транспорт на перший погляд надає порівняно невелику негативну дію на ОС. На самій же справі така дія вельми відчутно. Прокладка полотна дороги супроводиться виїмкою грунтів по обидві сторони дороги. Ширіна порушеної смуги з врахуванням смуг відчуження по обидві сторони дороги досягає в середньому 120 м-кодів, або 12 га на 1 км. довжини залізниці. В цілому по краю це складає 26-28 тис. га. На роз'їздах, полустанках, станціях майдан смуги відчуження у багато разів збільшується (що додає ще 18-20 тыс.га) під склади, прокладку додаткових доріг, будівництво вагонних і локомотивних депо, заправних установок для миття рухливого складу і так далі

В ході експлуатації залізниці на всій дорозі дотримання поїздів постійно втрачають або просто викидають сміття, сипкі вантажі, нафта і нафтопродукти, добрива, будівельні матеріали і тому подібне І хоча їх разове кількість невелика, але за роки уздовж залізничної колії накопичується задоволено багато відходів, що надають, особливо при під'їзді до вузлових станціям, значна дія на спільний вигляд і достаток рослинного покриву смуги відчуження.

Роботи в депо в гігієнічному відношенні схожі з роботою в машинобудівному цеху, електротехнічних, дерево- і металообробних підприємствах: зварювальні, забарвлення і інші види робіт усередині вагонів, цистерн контейнерів, машинних відділень локомотивів пов'язані із забрудненням повітря парами розчинників, що містять з'єднання хрому, цинку, свинцю і інших хімічних речовин, миючих розчинів, змащувальних матеріалів; забруднення свинцем на лінії ремонту акумуляторів, кислотами в процесі їх зарядки; при пульверизаторной забарвленню остову і вузлів електричних машин в повітря виділяються пари бензину, толуолу, ксилолу, ацетону, бутил- і амилацетона сольвентнафта, полиэтилдиамина, скипидару і інших розчинників. Ті ж речовини виділяються при забарвленні котушок і якоря, а вага разом викликає різні шкіряні захворювання. До агресивних речовин відносяться також використовувані і процесі ремонту тепловозів ацетон, освітлювальний гас авіабензин Б-70, нітрон, скипидар, сурик, лаки, сода кальцинована і каустична, біхромат натрію і багато інших.

Та ж специфіка дії на ОС і в цехах локомотивних заводів, включаючи підготовку до ремонту локомотивів, путніх машин, вагонів, їх очищення від залишків вантажів, льоду і снігу, розбирання і звільнення від мастила, палива рідин, що охолоджують, миття, знежирення окремих агрегатів і вузлів після розбирання. Таким чином, тут переважає хімічне забруднення виробничої середи і найближчого оточення.

Морський транспорт - джерело, головним чином, біологічного забруднення. У зв'язки з посиленням внутрішніх і міжнародних зв'язків, слабкістю карантинної служби, безконтрольністю в Азово-чорноморському басейні продовжує зростати біологічне забруднення:

зростання чисельності гребневика, завезеного судами з Атлантики, що веде до руйнуванню екосистеми Чорного і Азовського Морея, у тому числі і до скорочення рибних ресурсів.

Відому небезпеку представляють неконтрольовані сливи баластних вод побутове сміття, що викидається командами судів і особливо катастрофічні викиди нафти і її продуктів в разі аварій танкерів, що трапляється майже щорік.

\"Відносно участі авіаційного транспорту в забрудненні ОС маловивченими залишаються наслідки забруднення ним верхніх шарів атмосфери. Головні забруднюючі речовини в аеропортах - вуглеводні і шумове забруднення. Від останнього особливо страждають жителі будинків, прилеглих до зони аеропорту і розташованих в коридорах зльоту і посадки літаків.

Лісопромисловий комплекс і охорона ОС

Те, що ліси - легкі планети, давно стало хрестоматійною істиною. Так само добре відомо, що їх майдан на Землі катастрофічно скорочується під ударами все зростаючих потреб в деревині і в орних майданах у зв'язку із зростанням спільній чисельності людської популяції. Темпи скорочення лесопокрытой майдану вказують на реальну перспективу \"облысения\" планети, про яке давно запобігають екологи. У Краснодарському краю лесопокрытая майдан складає близько 20 % території і в основному локалізована в гірських районах, включаючи Чорноморське побережжя від Новоросійська ка Адлера. Особливо висока питома вага лесопокрытой майдани в Апшеронському, Туапсинськом районах, Геленджікськом і Сочинському міськрадах (від 80 до 89 %), тоді як в 23 правобережних прикубанских степових районах цей показник не піднімається вище 5 %, та і ці насадження в основному штучною характеру. Такий розподіл майдану лісів в краю не випадково. В умовах гірського рельєфу, великої кількості випадних опадів, часто зливового характеру, в гірських районах необхідно прагнути до максимального збереження лісів і скорочення промислової лісосіки. Проте це не завжди удається, оскільки край дає 77 % заготовки ділової деревини всього Північного Кавказу і має в своєму розпорядженні значні потужностями меблевої і деревообробної промисловості. На жаль велика частина стиглої і перестійної деревини збереглася у важкодоступних місцях, а вирубки здійснюються в місцях, де заготовки загрожують екологічною безпеці. Мають місце і інші види порушення екологічності використання лісових ресурсів:

- порушення правил, що діють, і норм лісокористування (великі майдани неочищених лісосік, в яких пропадає до 10 % деревини і порубкових залишків);
- технологія трелювання і вивезення деревини противоречит захисним функціям гірських лісів (вживання гусеничних тракторів), приводить до руйнування грунтового покриву, здиранню лісової підстилки, посиленню ерозійних процесів знищенню підзростання і молодняка;
-при вирубуванні відходу вирубуються здорові дерева, а не ослаблені і пригноблювані що веде до виснаження лісових ресурсів і деградації лісового фонду;
- лесовосстановительные роботи не встигають за вирубкою лісу через погану приживаності посадок, як наслідок в недбалості у відході.

Лісозаготівлі на території краю ведуть понад три десятки підприємств що заготовлюють 1,6-1,7 млн. м-коду деревини. Меблеві, деревообробні тарні підприємства переробляють до 800 тис. м3 круглого ліс;! і 250-270 тыс.м3 пиломатеріалів. Лісопромисловий комплекс по облиште використання відходів виробництва (окрім санітарних вирубувань) дров'яної деревини, тріски, тирса і тому подібне, займає одне з перших місць в краю, та і на Північному Кавказі.

В цьому відношенні слід віддати належне працівникам ЛПК, що постійно шукає можливості скорочення вжитку лісових ресурсів за рахунок повноти використання відходів виробництва.

Місто як джерело забруднення довкілля

Місто - найбільш динамічна форма розселення населення. Темпи зростання чисельності міського населення на планеті перевищують темпи зростання сільського. Міста, особливо великі, розповзаються по прилеглих до них територіях на зразок нафтової плями в океані, поглинаючи передмістя, поширюючи все далі і далі свій вплив. Так, лише за останніх 12 років на сучасній території Краснодарського краю (без Адигейської республіки) чисельність міського населення виросла на 337,5 тис. чоловік, тоді як чисельність сільського лише на 60,0 тис., або в 5,5 разів менше.

У історичному і соціальному розвитку міста представляються не лише найважливішою формою концентрації населення, але і особливою формою взаємин між людиною і природою. Ця особливість викликана тим, що в містах вона досягає високої щільності концентрації на одиницю майдану всіляких джерел забруднення, різних по характеру і вмісту шкідливих викидів у ОС, токсичності, силі негативної дії на природу і людину. У містах високої міри сили і концентрації досягають такі малопоширені види забруднень, як вібрація, шумове електромагнітне, теплове забруднення і тому подібне Крім того, міська екосистема, як правило, розділяється на центральну, щільно забудовану частину і периферію, тобто тіньову, або зовнішню зону, що охоплює окрім околиці самого міста, його приміську зону, що відрізняється за об'ємом і по характеру забруднення наявністю \"колец\" і \"полос\" (румбів) концентрації забрудників, співпадаючих з напрямом транспортних магістральних вулиць і найбільш стійких переміщень повітряних мас.

У складі забрудників центральної частини міст переважають пил вуглеводи, хімічне, теплові, шумове, електромагнітне. а практично весь «букет» сучасних викидів великого міста. На околицях - забруднення важкими металами, твердими відходами виробництва і вжитку промисловими стоками. Сюди ж досить широкий проникають основні забрудники агрокомплексу гербіциди, пестициди і тому подібне).

Недолік грошових коштів для забезпечення зростаючого населення міста послугами і засобами, необхідними для нормального життя, приводить до того що псові більше число поселень не має елементарних санітарно-побутових умов. Досить сказати, що 20 % жителів міст краю живуть в квартирах і будинках, не обладнаних водопроводом, .15 % -канализацией і центральним опалюванням, 47 % - теплопостачанням, тобто майже третина жителів міст не забезпечена елементарними санітарно-гігієнічними умовами життя, що природно, загострює негативні наслідки забруднення міської середи створить додаткову загрозу виникнення епідемій.

Однією з центральних проблем міської екології є проблема накопичення, зберігання і утилізації побутових твердих відходів. Як мінімум середньорічний житель міста щорік викидає до 250 кг сміття, що в цілому по краю складає не менше 640-650 тыс.т щорік. Полігони звалищ побутових відходів такого міста, як Москва, займають більше 500 гектарів землі. Звалища небезпечні як розповсюджувачі інфекцій, забрудники грунтових код. Приблизно до 0,4 % їх вміст складають матеріали високої токсичності (що містять ртуть, сірчану кислоту, фарби, розчинники, ацетон, спирт і т.п.). Присутність цих елементів задоволена часто приводить до пожеж на звалищах, до травмування і отруєння робітників, і навіть після поховання десятиліттями продовжується забруднення підземних вод продуктами вилуговування звалищ, оскільки переважна більшість з них не екрановано і не має інших природоохоронних пристроїв.

";
$INF1= new inf();
if(**$_GET**['info']**!**=1){

```
                $INF1-> Init(4,140,"black",100,1,"NOT",$aCC,"<a
href=\"javascript:Leave_Data_Delete('0','0','~','~','1','1','info','stor');true\" id=\"knn11\"class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                                                onmouseout=\"javascript:mouseOutl(this.id)\"><font size=\"3\"><b>Екологія як
наука</b></font></a>","","~');
                $INF1->show();
                ?>

        <a href="javascript:Leave_Data_Delete('0','0','~','~','1','1','info','stor');true" id="knn1"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Докладніше</b></font></a><br>
                <br><br><?
        }else{
                $INF1-> Init(4,150,"black",100,3,"NOT",$aCC,"<b>Екологія як наука</b>","","~');
                $INF1->show();
                ?>
                <a href="javascript:Leave_Data_Delete('0','0','~','~','2','~','~','info');true" id="knn1"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)"><font size="2"><b>Сгорнути</b></font></a><hr
color="darkblue">
                <br><?
        };

         ?>

        </div>

        Stor6.php

        <b>ЕКОЛОГІЧНІ КАРТИ МІСТА</b>
        <br>
        <br>

        <div align="left">
        <img src="/images/exm1.bmp">
        </div>
        stor7.php
        <b>ЗАКОНОДАВЧІ АКТИ ЩОДО ЕКОЛОГІЇ МІСТА</b>
        <br>
        <br>

        <div align="left">

        </div>
        trace.htm
        <HTML>
                <HEAD>

                        <TITLE></TITLE>
                </HEAD>
                <BODY>
                <img src="/images/image3.gif" />
                </BODY>
        </HTML>
        autorization.php
        <!--// Вход на сайт
        После выхода пользователя с сайта при ?quit=1 - куки $AUTOR
        должно быть обновлено например до Null или удалено.
        Во всех остальных случаях куки принимается в переменную $AUTOR=null.
        Если форма на экране то переменная $show_T=true.
        //--->
        <script language="javaScript">
        function sc(){
        document.aut.auto.value="on";
        return true;
        };
        </script>
        <?
        $INME=null;
        $IDEN=$_POST['auto'];
        if($IDEN=='on'){
                $LOG1=$_POST['login'];
                $PAS=$_POST['password'];
                };
        $FORM_AUT="<b>Авторизація</b><br>
        <div align=\"left\"><form action=\"/index1.php\" name=\"aut\" method=\"post\" >
                                        <b>Логін</b><br>
```

```
                                        <input type=\"text\" name=\"login\"><br>
                                        <b>Пароль</b><br>
                                        <input type=\"text\" name=\"password\"><br>
                                        <input type=\"submit\" value=\"Увійти\" onclick=\"javascript:sc();\">
                                        <input type=\"hidden\" value=\"1\" name=\"auto\">
                                        <font size=\"2\"><a href=\"/index1.php?register=1\" id=\"kp1\" class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                                        onmouseout=\"javascript:mouseOutl(this.id)\">Реєстрація</a></font><br>
                                        </form></div>";
                        //Проверка авторизации
                        //Пользователь отправил данные
                        $Welcame="<font size=\"2\" color=\"clred\"><b>Доброго дня,";
                $Welcame1="</b></font><b></b><br><font size=\"2\"><a href=\"/index1.php?quit=1\" id=\"kng2\" class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                                onmouseout=\"javascript:mouseOutl(this.id)\">Вихід</a></font><br><br>";
                if((session_is_registered('over'))and($_SESSION['over']==$AUTOR)){
                        $I=$_SESSION['name_ID'];
                echo $Welcame."$I".$Welcame1;$INME=1;}else{
                                if($IDEN=="on"){
                                        //Пользователь отправил данные входа на сайт.
                                        $str="SELECT * FROM Users WHERE ((Users.Login='".$LOG1."')&&(Users.Pass='".$PAS."'))";
                                }else{
                                        //Извлечение из БД информации о пользователе
                                        $str="SELECT * FROM Users WHERE (Users.ID='".$AUTOR."')";
                                };
                                //Проверка совпадения с базой данных
                                $str=SQLWork('localhost','root','password','User',$str);
                                $I=mysql_num_rows($str);
                        if($I!=0){
                                //Вывод данных о пользователе
                                $I=mysql_result($str,0,'I');
                                echo "$Welcame"."$I"."$Welcame1";
                                session_register('name_ID');
                                $_SESSION['name_ID']=$I;
                                $I=mysql_result($str,0,'ID');
                                session_register('ses_ID');
                                $_SESSION['ses_ID']=$I;
                                //Проверка на наличие куки
                                $str="SELECT * FROM Users WHERE (Users.ID='".$AUTOR."')";
                                $str=SQLWork('localhost','root','password','User',$str);
                                $I=mysql_num_rows($str);
                                if($I==0){

                                        print"<script language=Javascript>window.location=\"/income.php\";</script>";
                                        exit;
                                };
                                //Вывод непосредственно заданной информации
                                //Создание сессионных переменных для настройки сайта, если их нет, а также инициализация
                                $I=mysql_result($str,0,'temp');
                                session_decode($I);
                                session_register("over");
                                $_SESSION['over']=$_SESSION['ses_ID'];
                                session_register("numberBD");
                                $_SESSION['numberBD']=1;
                                $INME=1;
                        };
                        };

                ?>

                autorization1.php
                <?

                if($INME==null){
                        if($IDEN=="on"){
                                //Пользователь отправил данные входа на сайт.
                                $str="SELECT * FROM vodaUsers WHERE
((vodaUsers.Login='".$LOG1."')&&(vodaUsers.Pass='".$PAS."'))";
                        }else{
                                //Извлечение из БД информации о пользователе
                                $str="SELECT * FROM vodaUsers WHERE (vodaUsers.ID='".$AUTOR."')";
                        };
                        //Проверка совпадения с базой данных
                        $str=SQLWork('localhost','root','password','vodaT',$str);
                        $I=mysql_num_rows($str);
                        if($I!=0){
                                //Вывод данных о пользователе
```

```
                $I=mysql_result($str,0,'I');
                echo "$Welcame"."$I"."$Welcame1";
                session_register('name_ID');
                $_SESSION['name_ID']=$I;
                $I=mysql_result($str,0,'ID');
                session_register('ses_ID');
                $_SESSION['ses_ID']=$I;
                //Проверка на наличие куки
                $str="SELECT * FROM vodaUsers WHERE (vodaUsers.ID='".$AUTOR."')";
                $str=SQLWork('localhost','root','password','vodaT',$str);
                $I=mysql_num_rows($str);
                if($I==0){

                        print"<script language=Javascript>window.location=\"/income.php\";</script>";
                        exit;
                };
                //Вывод непосредственно заданной информации
                //Создание сессионных переменных для настройки сайта, а также инициализация
                //...
                //...
                session_register("over");
                $_SESSION['over']=$_SESSION['ses_ID'];
                $INME=1;
                session_register("numberBD");
                $_SESSION['numberBD']=2;
        } ;

};

        // Дальше проверяем наличие пользователя по всем специализированным базам данных
        //аналогично этому модулю ...
?>
```

autorization2.php

```
<?
if($INME==null){
        if($IDEN=="on"){
                //Пользователь отправил данные входа на сайт.
                $str="SELECT * FROM admT WHERE ((admT.Login='".$LOG1."')&&(admT.Password='".$PAS."'))";
        }else{
                //Извлечение из БД информации о пользователе
                $str="SELECT * FROM admT WHERE (admT.ID='".$AUTOR."')";
        };
        //Проверка совпадения с базой данных
        $str=SQLWork('localhost','root','password','adminT',$str);
        $I=mysql_num_rows($str);
        if($I!=0){
                //Вывод данных о пользователе
                $I=mysql_result($str,0,'I');
                echo "$Welcame"."$I"."$Welcame1";
                session_register('name_ID');
                $_SESSION['name_ID']=$I;
                $I=mysql_result($str,0,'ID');
                session_register('ses_ID');
                $_SESSION['ses_ID']=$I;
                //Проверка на наличие куки
                $str="SELECT * FROM admT WHERE (admT.ID='".$AUTOR."')";
                $str=SQLWork('localhost','root','password','adminT',$str);
                $I=mysql_num_rows($str);
                if($I==0){

                        print"<script language=Javascript>window.location=\"/income.php\";</script>";
                        exit;
                };
                //Вывод непосредственно заданной информации
                //Создание сессионных переменных для настройки сайта, а также инициализация
                //...
                //...
                session_register("over");
                $_SESSION['over']=$_SESSION['ses_ID'];
                $INME=1;
                session_register("numberBD");
                $_SESSION['numberBD']=-1;
        } ;
        if($INME!=1){print"$FORM_AUT"; };
};
```

```
// Дальше проверяем наличие пользователя по всем специализированным базам данных
//аналогично этому модулю ...
?>
```

registration.php

```
<!---//Форма регистрации //--->
<script language="javascript">
function sa(){
document.FormMain.second.value="on";
return true;
};
</script>
<?php
$show_T=false;
$FORM_T="<form action=\"/index1.php\" method=\"post\" name=\"FormMain\">
                <table border=\"0\" cellspacing=\"20\" cellpadding=\"0\">
                <tr >
                <td align=\"left\">
                <b>Прізвище</b><br>
                <input type=\"text\" name=\"F\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td>
                <td align=\"left\">
                <b>Логін</b><br>
                <input type=\"text\" name=\"F4\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td>
                </tr>
                <tr ><td align=\"left\">
                <b>Им'я</b><br>
                <input type=\"text\" name=\"F1\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td><td align=\"left\">
                <b>Пароль</b><br>
                <input type=\"text\" name=\"F5\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td>
                </tr>
                <tr ><td align=\"left\">
                <b>По батькові</b><br>
                <input type=\"text\" name=\"F2\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td><td align=\"left\">
                <b>Пароль ще раз</b><br>
                <input type=\"text\" name=\"F6\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td></tr><tr>
                <td align=\"left\">
                <b>Посада</b><br>
                <input type=\"text\" name=\"F3\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td><td align=\"left\">
                <b>Почта</b><br>
                <input type=\"text\" name=\"F7\" value=\"\" vspace=\"3\" hspace=\"100\">
                </td></tr><tr>
                <td colspan=\"2\" align=\"center\">
                <input type=\"submit\"   value=\"Зареєструватися\" onclick=\"javascript:sa();\">
                <input type=\"hidden\" name=\"second\" value=\"1\"><br>
                <font><b><a href=\"/index1.php\" id=\"knu1\"class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                        onmouseout=\"javascript:mouseOutl(this.id)\">Повернутися назад</a></b></font>
                </td>
                </tr>
                </table>
                </form>";
if($_GET['register']==1){       print("$FORM_T");$show_T=true;         };
//Если регистрационные данные были отправлены на сервер.
if($_POST['second']=="on"){$s=true;
        if($_POST['F']==""){$s=false;echo"Не указано Фамилия ";};
        if($_POST['F1']==""){$s=false;echo"Не указано Имя ";};
        if($_POST['F2']==""){$s=false;echo"Не указано Отчество ";};
        if($_POST['F3']==""){$s=false;echo"Не указано Должность ";};
        if($_POST['F4']==""){$s=false;echo"Не указано Логин ";};
        if($_POST['F5']==""){$s=false;echo"Не указано Пароль ";};
        if($_POST['F6']==""){$s=false;echo"Не указано Подтверждение пароля ";};
        if($_POST['F7']==""){$s=false;echo"Не указано Почтовый адрес ";};
        if($_POST['F5']!=$_POST['F6']){$s=false;echo"Пароли не совпадают";};
        if(strlen($_POST['F5'])<8){$s=false;echo"Пароль должен быть не менее 8 сиволов ";};
        //Если ошибок ввода регистрационных данных нет
        if($s==true){
                // Записуем в базу данных данные о регистрации
                srand((double)microtime()*1000000);
                $ID_T=uniqid(rand(),true);
                //Пример регистрации одной из сессионных переменных пользователя
```

```
                        session_register('mapsShow');
                        $_SESSION['mapsShow']=5;
                        // ..........
                        //..........
                        //..........
                        $code_L=session_encode();
                        //Связь с базой данных
                        $ID_S=@mysql_pconnect("localhost", "root", "password") or die("Could not connect to MySQL server!");
                        @mysql_select_db("user",$ID_S) or die("Could not select company database!");
                        $st="INSERT INTO Users(F,I,O,D,Login,Pass,Pass1,Mail,ID,temp) values('".$_POST['F']."','".
                                $_POST['F1']."','".$_POST['F2']."','".$_POST['F3']."','".$_POST['F4']."','".
                                $_POST['F5']."','".$_POST['F6']."','".$_POST['F7']."','".$ID_T."','".$code_L."')";
                        //Вставка данных в БД.
                        @mysql_query($st,$ID_S);
                        mysql_close($ID_S);
                        session_register('sessionID');

                        $_SESSION['sessionID']=$ID_T;
                        print"<script language=Javascript>window.location=\"/registered.php\";</script>";
                        exit;
                } else{print"$FORM_T";$show_T=true;};
};
?>
```

library.php

```
<style type="text/css">
.default{
cursor:pointer;
}
.hideT{
visibility: hidden;
}
.showTd{
display: block;
}
.hideTd{
display: none;
}
.DefaultRectandle{
border: 2px solid darkblue
}
</style>

<script language=javascript>
<!--

function RectSet(a){
document.getElementById(a).style.border="2px solid darkmagenta";
return 1;
};
function RectDelete(a){
document.getElementById(a).style.border="2px solid darkblue";
return 1;
};


function Leave_Data_Delete(){
var s="";
if(arguments[1]==0){s=location.href;};
if(arguments[1]==1){var z=arguments[2]; s=document.getElementById(z).value;}
if(arguments[1]==2){s=arguments[2];};

var j=6;
var i=0;
var sm=s.indexOf('#');
if(sm!=-1){
s=s.split('#');
if((arguments[3]!='~')&&(arguments[3]!='0')){sm='#'+s[1];s=s[0];}else{s=s[0];sm='';};
}else{sm='';};


var s1=s.indexOf("?",0);
if(s1!=-1){
s=s.split("?");
s1=s[1];
s=s[0];
s1=s1.split("&");
```

```
var s2='?';
var tl;
var flag=0;
while(s1[i]){

while(arguments[j]){
tl=arguments[j]+'=';
if(s1[i].indexOf(tl,0)!=-1){
if(((j==6)&&(arguments[4]==0))||((j==6)&&(arguments[4]==1))){s2=s2+tl+arguments[5]+'&'; flag=1;}else{
if(((j!=6)&&(arguments[4]==1))||((j!=6)&&(arguments[4]==3))){s2=s2+s1[i]+'&';};};
tl='nop';break;
};
j++;
};

j=6;
if(tl!='nop'){if ((arguments[4]==0)||(arguments[4]==2)){s2=s2+s1[i]+'&';};};
i++;};

if(((arguments[4]==0)&&(flag==0))||((arguments[4]==1)&&(flag==0))){s2=s2+arguments[6]+'='+arguments[5];}else{s2=s2.substr(0,
s2.length-1);};
if(s2!='?'){s=s+s2;};

} else{ if((arguments[4]==0)||(arguments[4]==1)){s=s+'?'+arguments[6]+'='+arguments[5];};};
s=s+sm;
if(arguments[0]!=0){return s;}else{location.href=s;};
};

function OverMouse(n){
var s='HIDDEN_DATA_T'+n;
s=document.getElementById(s).value;
s=s.split('~');
if ((s[0]==0)||(s[0]==3)){document[n].src=s[4]; } else{ document[n].src=s[3];};
};


function CkickMouse(n){
var s='HIDDEN_DATA_T'+n;
s=document.getElementById(s).value;
s=s.split('~');
if ((s[0]==0)||(s[0]==3)){document[n].src=s[5]; } else{ document[n].src=s[4];};
};


function OutMouse(n){
var s='HIDDEN_DATA_T'+n;
s=document.getElementById(s).value;
s=s.split('~');
if ((s[0]==0)||(s[0]==3)){  document[n].src=s[3]; } else{ document[n].src=s[2];};
};

function UpMouse(n){
var s='HIDDEN_DATA_T'+n;
s=document.getElementById(s).value;
s=s.split('~');
if ((s[0]==0)||(s[0]==3)){document[n].src=s[3]; } else{ document[n].src=s[2];};

if (s[0]==0){
if(s[1]==''){alert("Ошибка ! Не задан URL  адрес кнопки");};
s[4]=s[1].indexOf('#',0);
if(s[4]!=-1){
s[1]=s[1].substr(0,s[4]);
};

s[3]=s[1].indexOf('?',0);

if(s[7]==''){

if(s[3]!=-1){
if(s[6]!=''){s[3]=s[1]+'&'+s[6]+'='+s[2];}else{s[3]=s[1];};
}else{
if(s[6]!=''){ s[3]=s[1]+'?'+s[6]+'='+s[2];}else{s[3]=s[1];};
};

}else{
if(s[3]!=-1){
if(s[6]!=''){s[3]=s[1]+'&'+s[6]+'='+s[2]+s[7];}else{s[3]=s[1]+s[7];};
```

```
}else{
if(s[6]!=''){ s[3]=s[1]+'?'+s[6]+'='+s[2]+s[7];}else{s[3]=s[1]+s[7]; };};
};
location.href=s[3];
};

if(s[0]==1){

if(s[5]==''){alert("Ошибка ! Не задано имя выводимой переменной");};
if(s[1]==''){alert("Ошибка ! Не задано значение выводимой переменной");};

s[2]=location.href;


s[4]=s[2].indexOf('#',0);
if(s[4]!=-1){
s[2]=s[2].substr(0,s[4]);
};
s[3]=s[2].indexOf('?',0);
if(s[6]==''){
if(s[3]!=-1){s[2]=s[2]+'&'+s[5]+'='+s[1];}else{ s[2]=s[2]+'?'+s[5]+'='+s[1];};
}else{
if(s[3]!=-1){s[2]=s[2]+'&'+s[5]+'='+s[1]+s[6];}else{ s[2]=s[2]+'?'+s[5]+'='+s[1]+s[6];};
};

location.href=s[2];
};



if(s[0]==2){

if(s[5]==''){alert("Ошибка ! Не задано имя выводимой переменной");};
if(s[1]==''){alert("Ошибка ! Не задано значение выводимой переменной");};

s[0]=location.href;

s[2]=s[0].indexOf('?',0);

if(s[2]!=-1){
s[1]=s[5]+'='+s[1];
s[5]='';
s[0]=s[0].split('?');
s[2]=s[0][1];
s[0]=s[0][0];


s[3]=s[2].indexOf('#',0);
if(s[3]!=-1){
s[2]=s[2].split('#');
s[5]='#'+s[2][1];
s[2]=s[2][0];
};

s[2]=s[2].split('&');
s[3]=0;
s[4]='?';

while(s[2][s[3]]){
if(s[2][s[3]]!=s[1]){s[4]=s[4]+s[2][s[3]]+'&';};
s[3]++;
};
s[4]=s[4].substr(0,s[4].length-1);
s[0]=s[0]+s[4]+s[5];
};
location.href=s[0];
};

if(s[0]==3){
s[1]='javascript:'+s[1]+'();';
location.href=s[1];
};

if((s[0]>3)||(s[0]<0)){alert('Ошибка ! Неверно задан тип создаваемой кнопки');};
};
//-->
</script>
```

```
<?php
function CreateBUTTON($i1,$i2,$i3,$NAME,$NameBut,$namebutton,$hreff,$Raf,$title,$MODE_URL,$style_T){
	$Name_hidden_T='HIDDEN_DATA_T'.$namebutton;
	if($NameBut=='~'){$NameBut='';};
	if($title=='~'){$title='';};
	if($NAME=='~'){$NAME='';};
	if($Raf!='~'){$Raf='#'.$Raf;}else{$Raf='';};
	if($hreff=='~'){$hreff='';};

	if
(($MODE_URL==0)||($MODE_URL==3)){$s=$MODE_URL.'~'.$hreff.'~'.$NameBut.'~'.$i1.'~'.$i2.'~'.$i3.'~'.$NAME.'~'.$Raf;}else{
	$s=$MODE_URL.'~'.$NameBut.'~'.$i1.'~'.$i2.'~'.$i3.'~'.$NAME.'~'.$Raf; };
	?>
	<img class="<?echo"$style_T";?>" title="<?echo"$title";?>" src="<?echo"$i1";?>" name="<?echo"$namebutton";?>"
OnMouseDown="CkickMouse(this.name);" onmouseover="OverMouse(this.name);" onmouseOut="OutMouse(this.name);"
OnMouseUp="UpMouse(this.name);">
	<input type="hidden" id="<?echo"$Name_hidden_T";?>" value="<?echo"$s";?>">
	<?
};

/*
формальные параметры функции SQLWork
s-host
s1-login
s2-password
s3-database
s4-SQL query
*/
function SQLWork($s,$s1,$s2,$s3,$s4){
	$ID_S=@mysql_pconnect($s, $s1, $s2) or die("Ошибка! Невозможно установить связь с базой данных");
	@mysql_select_db($s3,$ID_S) or die("Ошибка! невозможно выбрать базу данных");
	$s4=mysql_query($s4,$ID_S);
	@mysql_close($ID_S) or die ("Ошибка! Невозможно закрыть базу данных");
	return $s4;
	};
	class inf {
	var $SizZ;
	var $kolSim;
	var $colorZ;
	var $kolzag;
	var $mode;
	var $length;
	var $memo;
	var $tema;
	var $zakladka;

	function cutting($SSm,$sw,$one){
		$slash=0;
		$KOL_POINT=3;
		$kol_point_oper=1;
		$le=strlen($sw);
		$i=0;
		$Sim=0;
		$sum=0;
		$fl=0;
		$fl1=0;
		while($i!=$le):
			if(($i==0)&&($sw[$i]==' ')){$fl1=1;echo"$sw[$i]";$i++;continue;};
			if(($sw[$i]=='>')and($fl==1)):$fl=0;echo"$sw[$i]";$i++;continue;endif;
			if($sw[$i]=='<'):

				if($slash==1){
					$slash_str=substr($sw,$i,4);
				if($slash_str=="<br>"){$i+=4;continue;}; };

				$fl=1;echo"$sw[$i]";$i++;continue; endif;
			if($fl==1):echo"$sw[$i]";$i++;continue;endif;

			if($this->mode!=3){ if($sum==$SSm){$slash=1;
while($kol_point_oper<=$KOL_POINT){echo".";$kol_point_oper++;};$i++;continue;};};


			if($one!='NOT'){ if($Sim==$one) {echo"$sw[$i]"," ";$Sim=0;$fl1=1;$i++;continue;};};

			if($sw[$i]==' '){if($fl1==0){$sum++;$Sim=0;$fl1=1;};echo"$sw[$i]";$i++;continue;};
```

```
                    echo"$sw[$i]";
                    $sum++;
                    $Sim++;
                    $i++;
                    $fl1=0;
                endwhile;

        }


        function show(){
                if($this->colorZ!='NOcolor'){echo "<font size=$this->SizZ color=$this->colorZ>";}else{
                echo "<font size=$this->SizZ >";};

                if(($this->mode==1)||($this->mode==3)) {
                        if($zakladka!='~'): echo"<a name=\"$this->zakladka\" >";endif;
                }else{


                        if($this->mode==0){
                                echo "<a  href=\"$this->href\">";
                        };

                };

                $this->cutting($this->kolzag,$this->tema,$this->length);
                if(($this->mode==1)||($this->mode==3)){
                        if($zakladka!='~'): echo"</a >";endif;
                print("</font>");} else {
                        if($this->mode==0){
                        print("</a>");};
                        print("</font>");
                };

                if (($this->mode==1)||($this->mode==3)) :
                        ?>
                        <br>
                        <?
                        $this->cutting($this->kolSim,$this->memo,$this->length);
                        ?>

                <?endif; }
        function Init($a2,$a3,$a4,$a5,$a6,$a7,$a8,$a9,$a10,$a11){
                $this->SizZ=$a2;
                $this->kolSim=$a3;
                $this->colorZ=$a4;
                $this->kolzag=$a5;
                $this->mode=$a6;
                $this->length=$a7;
                $this->memo=$a8;
                $this->tema=$a9;
                $this->href=$a10;
                $this->zakladka=$a11;
        }

};

function numberMonth($a){
        switch(true):
        case($a==1):
                return "Січень";
                break;
        case($a==2):
                return "Лютий";
        break;
        case($a==3):
                return "Березень";
        break;
        case($a==4):
                return "Квітень";
        break;
        case($a==5):
                return "Травень";
        break;
        case($a==6):
                return "Червень";
        break;
```

```
                case($a==7):
                        return "Липень";
                        break;
                case($a==8):
                        return "Серпень";
                        break;
                case($a==9):
                        return "Вересень";
                        break;
                case($a==10):
                        return "Жовтень";
                        break;
                case($a==11):
                        return "Листопад";
                        break;
                case($a==12):
                        return "Грудень";
                        break;
                default:return 0;
                endswitch;
                };


        ?>
```

shablon.php

```
        <style type="text/css">
        .offDecoration{
        text-decoration: none;
        color:darkblue
        }
        </style>

        <script language="javascript">

        function mouseMovel(a){
        document.getElementById(a).style.textDecoration="Underline";
        return true;
        };

        function mouseOutl(a){
        document.getElementById(a).style.textDecoration="none";
        return true;
        };
        </script>
        <?php

        function shapka(){
                echo"       <html>
                                <head><title>Система міського екологічного моніторінгу</title></head>
                                <body BGCOLOR=\"whitesmoke\">
                                <table width=\"100%\" cellpadding=\"5px\" cellspacing=\"1px\" style=\"border: 2px solid darkblue;
border-collapse: separate \" >

                                <tr style=\"background: skyblue\">
                                <td  align=\"right\" colspan=\"3\" height=\"100px\" valign=\"bottom\" style=\"border: 3px solid
darkblue;\" >
                                <img src=\"/images/1.jpg\" alt=\"Error\" height=\"85px\" >
                                <img src=\"/images/2.jpg\" alt=\"Error\" height=\"85px\" >
                                </td>
                                </tr>
                                <tr >
                                <td BACKGROUND=\"/images/fon.png\" colspan=\"3\" height=\"1px\">
                                <font size=\"2\">
                                <a href=\"/index1.php\" style=\"text-decoration: none;
                                color:white\" id=\"topd1\"
                                onmousemove=\"javascript:mouseMovel(this.id)\"
                                onmouseout=\"javascript:mouseOutl(this.id)\">Головна</a>

                                <a href=\"/index1.php\" style=\"text-decoration: none;
                                color:white;position:relative; left: 15%  \" id=\"topd2\"
                                onmousemove=\"javascript:mouseMovel(this.id)\"
                                onmouseout=\"javascript:mouseOutl(this.id)\">Доповіді</a>

                                <a href=\"/index1.php\" style=\"text-decoration: none;
                                color:white;position:relative; left: 30%  \" id=\"topd3\"
```

```
                                        onmousemove=\"javascript:mouseMovel(this.id)\"
                                        onmouseout=\"javascript:mouseOutl(this.id)\">Екологічні карти</a>

                                        <a href=\"/index1.php\" style=\"text-decoration: none;
                                        color:white;position:relative; left: 45%  \" id=\"topd4\"
                                        onmousemove=\"javascript:mouseMovel(this.id)\"
                                        onmouseout=\"javascript:mouseOutl(this.id)\">Законодавство</a>

                                        <a href=\"/index1.php\" style=\"text-decoration: none;
                                        color:white;position:relative; left: 60%  \" id=\"topd5\"
                                        onmousemove=\"javascript:mouseMovel(this.id)\"
                                        onmouseout=\"javascript:mouseOutl(this.id)\">Гостьова</a>

                                        </font>
                                        </td>

                                        </tr>
                                        <tr style=\"background: skyblue;\" height=\"390\">";
        return 0;
                };

        function endWeb(){
                echo"</tr>
                                        <tr style=\"background: skyblue; \"><td colspan=\"300\" align=\"left\" style=\"border: 1px solid
darkblue;\" >
                        <font color=\"black\" size=\"1\">            Технологічний інститут (м.Сєвєродонецьк 2009)</font>
                                        </td></tr>
                </table></body></html>";
                return 0;
                };
        ?>
```

admin.php

```
        <?php
        include_once"resource/shablon.php";
        // Регистрация амдинистратора
        shapka();
        ?>
        <td height="420px" valign="top" align="left" width="20%" style="border: 1px solid black">
        </td>
        <td style=" background: white;">
        <?
        if(!$_POST['V']){
        ?>
        <form method="POST" name="admT" >
        <table align="center">
        <tr>
        <td align="center">
        <b>Реєстрація адміністратору</b><br>
        <br>
        </td>
        </tr>
        <tr>
        <td>
        Логін
        <input type="text" name="L">
        </td>
        <td>
        Пароль
        <input type="text" name="P">
        </td>
        <td>
        Ім'я
        <input type="text" name="I">
        </td>
        </tr>
        <tr>
        <td>
        <input type="submit" value="Ввести">
        <input type="hidden" value="1" name="V">
        </td>
        </tr>
        </table>
        </form>
                <?
        }else{
```

```
                include_once"resource/library.php";
                srand((double)microtime()*1000000);
                $ID_D=uniqid(rand(),true);
                $SQL="INSERT INTO admT (Login,Password,ID,I) VALUES('".$_POST['L']."','".
                $_POST['P']."','".$ID_D."','".$_POST['I']."');";
                        SQLWork('localhost','root','password','adminT',$SQL);

                ?>
                <b>
                Дані вненсено
                </b>
                <?

                };
        ?>
        </td>
        <td style=" background: white;">
        </td>
        <?
        endWeb();
        ?>
```

income.php

```
        <?php
        session_start();
        setCookie('session',$_SESSION['ses_ID'],time()+3600);

        include_once"resource/shablon.php";
        shapka();
        ?>
        <td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue">
        </td>
        <td style=" background: white;">
        <div align="center"><font size="5">Доброго дня,<?$a=$_SESSION['name_ID'];echo"$a"?></font></div><br>
        <div align="center"><font><b><a href="/index1.php" id="kn1" class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font></div>
        </td>
        <td style=" background: white;">
        </td>
        <?
        endWeb();
        ?>
```

index1.php

```
        <?session_start();
        //Авторизация----------------------------------------------------
        $AUTOR=null;
        $_destroy=false;
        if($_GET['quit']!=1){$AUTOR=$_COOKIE['session'];
                include_once"resource/library.php";
        }else{
                setCookie('session',null,time()+3600);
                //Сохранение всех сессионных данных пользователя
                //.......
                //.......
                //----------------------------------------------
                include_once"resource/library.php";
                $ses=null;
                if(session_is_registered('numberBD')):session_unregister('numberBD');endif;
                if(session_is_registered('over')):session_unregister('over');endif;
                if(session_is_registered('name_ID')):session_unregister('name_ID');endif;
                $ses=$_SESSION['ses_ID'];
                if(session_is_registered('ses_ID')):session_unregister('ses_ID');endif;
                $data=session_encode();
                $str="UPDATE Users SET Users.temp='".$data."' where Users.ID='".$ses."'";
                SQLWork('localhost','root','password','User',$str);
                session_destroy();$_destroy=true;
                echo"<script language=\"javaScript\">window.location=\"/index1.php\";</script>";
                exit;
        };
        include_once"resource/shablon.php";
        shapka();
        ?>
        <!--//----------------------------------------
```

```
Основная часть
----------------------------------------------------------//--->
<td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue;">
<?
include_once"registration/autorization.php";
include_once"registration/autorization1.php";
include_once"registration/autorization2.php";
//.......
//.......
//проверка во всех базах данных.

?>
<!--//
Меню
//--->
<br>

<font><b><a href="/index1.php" id="kn1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Головна</a></b></font>
<br>
<?

?><font><b><a href="/index1.php?stor=1" id="kn2"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Повітря</a></b></font><br>
<?

?><font><b><a href="/index1.php?stor=2" id="kn3"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Вода</a></b></font><br>
<?

?><font><b><a href="/index1.php?stor=3" id="kn4"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Земля</a></b></font><br>
<?

?><font><b><a href="/index1.php?stor=4" id="kn5"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Зелені насадження</a></b></font><br>
<?

?><font><b><a href="/index1.php?stor=5" id="kn6"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Доповіді</a></b></font><br>
<?

?><font><b><a href="/index1.php?stor=6" id="kn7"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Екологічні карти</a></b></font><br>
<?

?><font><b><a href="/index1.php?stor=7" id="kn8"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Законодавство</a></b></font><br>
</td>
<!--//
Основная часть
---------------------------------------------------------- //--->
<td  width="60%" valign="top" align="center"  style=" background: white;">
<?
// Регистрация пользователя
include_once"registration/registration.php";
//Отображение общей информации
if($show_T==false){
        switch(true){
                case ($_GET['stor']==1):{include_once"information/stor1.php";break;};
                case ($_GET['stor']==2):{include_once"information/stor2.php";break;};
                case ($_GET['stor']==3):{include_once"information/stor3.php";break;};
                case ($_GET['stor']==4):{include_once"information/stor4.php";break;};
                case ($_GET['stor']==5):{include_once"information/stor5.php";break;};
                case ($_GET['stor']==6):{include_once"information/stor6.php";break;};
                case ($_GET['stor']==7):{include_once"information/stor7.php";break;};
        default:include_once"information/stor.php";};
};?>
</td>
<!--//
Правая часть
----------------------------------------------------------//--->
<td width="20%" style="background: white">
<?//Зарегистрированные пользователи могут видеть карты Северодонецка. На
//примере использования сиссионных переменных каждого пользователя для
//настройки и отображения данных
$_Maps="<script language=\"javaScript\">
```

```
function wnd(){
open(\"information/map.htm\");
};
function wnd1(){
open(\"information/trace.htm\");
};
</script>
                    <font size=\"2\"><b><a href=\"javascript:wnd();\" id=\"kni0\" class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                    onmouseout=\"javascript:mouseOutl(this.id)\">Карта
міста<br>Сєвєродонецьк</a></b></font><br><br><br>
                    <font size=\"2\"><b><a href=\"javascript:wnd1();\" id=\"kni1\" class=\"offDecoration\"
onmousemove=\"javascript:mouseMovel(this.id)\"
                    onmouseout=\"javascript:mouseOutl(this.id)\">Карта шляху<br> міста Сєвєродонецьк</a></b></font><br>";
if((session_is_registered('mapsShow'))and($_SESSION['mapsShow']==5)):echo"$_Maps"; endif;

//Пример отображения меню пользователей зарегистрированных
//администратором в базе данных по ведению экологического мониторинга водных ресурсов.
if (session_is_registered('numberBD')){
        switch(true){
                case($_SESSION['numberBD']==1):
                        //Пользователь зарегистрирован в общей базе данных
                        //...
                        //...
                        break;
                case($_SESSION['numberBD']==2):
                        //Пользователь зарегистрирован в БД по экологии относящейся к мониторингу водных
                        //ресурсов.
                        ?>
                        <font><b><a href="/workBD.php" id="knnn2"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                        onmouseout="javascript:mouseOutl(this.id)">Додати оперативну
інформацію</a></b></font><br>
                        <?
                        break;
                case($_SESSION['numberBD']==-1)://Числа идентификации могут быть усложнены для повышения
безопасности
                        //Администратор
                        ?>
                        <font><b><a href="/logic.php" id="knnn2"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                        onmouseout="javascript:mouseOutl(this.id)">Екологічний моніторинг</a></b></font><br>
                        <?
                        break;
                default:null;
                };
        };
?>


</td>
<!--//
Нижняя часть
----------//--->
<?
endWeb();
?>
<!--//
----------//--->
```

logic.php

```
<?php
session_start();
include_once"resource/shablon.php";
shapka();
if(($_SESSION['numberBD']==-1)&&(session_is_registered('numberBD'))){
        include_once"resource/library.php";
        include_once"analization/analization.php";
        include_once"analization/analization1.php";
        $IST=$_GET["ist"];
        if(is_null($IST)==true){
                // Титульная часть
?>
<td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue">
<br><br>
```

```
                                        <font><b><a href="/index1.php" id="knc0"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                                onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font><br>

            </td>
            <td style=" background: white;" align="center">

            <table border="0" cellspacing="0">
            <tr align="center" valign="bottom">
            <td width="150">
            <a href="/logic.php?ist=1" class="offDecoration"><img src="/images/water.jpg" id="pict1" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            <td width="150">
            <a href="" class="offDecoration"><img src="/images/gidro.jpg" id="pict2" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            <td width="150">
            <a href="" class="offDecoration"><img src="/images/vodokanal.jpg" id="pict3" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            <td width="150">
            <a href="" class="offDecoration"><img src="/images/xoz.jpg" id="pict4" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            </tr>
            <tr align="center" valign="top" style="line-height:90%">
            <td>
            <b>Водні<br>ресурси</b>
            </td>
            <td>
            <b>Гидрометслужба</b>
            </td>
            <td>
            <b>Міствводоканал</b>
            </td>
            <td>
            <b>Суб'єкти<br> господарювання</b>
            </td>
            </tr>
            <tr align="center" valign="bottom">
            <td>
            <a href="" class="offDecoration"><img src="/images/karantin.jpg" id="pict5" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            <td>
            <a href="" class="offDecoration"><img src="/images/geology.jpg" id="pict6" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            <td>
            <a href="" class="offDecoration"><img src="/images/les.jpg" id="pict7" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            <td>
            <a href="" class="offDecoration"><img src="/images/statistic.gif" id="pict8" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
            </tr>
            <tr align="center" valign="top" style="line-height:90%">
            <td>
            <b>Служба<br> карантину<br> рослин</b>
            </td>
            <td>
            <b>Дані з<br> геології</b>
            </td>
            <td>
            <b>Лісомисливське<br> господарство</b>
            </td>
            <td>
            <b>Відділ<br> статистики</b>
            </td>
            </tr>
            <tr align="center" valign="bottom">
            <td>
            <a href="" class="offDecoration"><img src="/images/epidem.jpg" id="pict9" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
            </td>
```

```
<td>
<a href="" class="offDecoration"><img src="/images/radio.jpg" id="pict10" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
</td>
<td>
<a href="" class="offDecoration"><img src="/images/mchs.jpg" id="pict11" class="DefaultRectandle"
onmousemove="javascript:RectSet(this.id);" onmouseout="javascript:RectDelete(this.id)"></a>
</td>
</tr>
<tr align="center" valign="top" style="line-height:90%">
<td>
<b>Санепідемстанція</b>
</td>
<td>
<b>Радіоаційна<br> лабораторія</b>
</td>
<td>
<b>Служба <br>МНС</b>
</td>
</tr>
</table>

</td>
<td style=" background: white;">

</td>
                    <?
          }else{
                    //-------------------------------------------------------------------
                    ////////////////////////////////////////////////////////////////////////////
                    // Если выбран источник данных
                    ////////////////////////////////////////////////////////////////////////////



                    ////////////////////////////////////////////////////////////////////////////
                    ////////////////////водные ресурсы/////////////////////////////////////////////

                    if($IST==1){

<?
<!---
Левая часть
//-->
<td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue">
                              <?php
                              switch(true){
                                        //+++++++++++++++++++++++++++++++++++++++++++++++++
                                        case ($_GET['ref']==null):
                                                  ?>
                                                  <font><b><a href="/logic.php?ist=1&ref=1" id="knc0"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                              onmouseout="javascript:mouseOutl(this.id)">Порушення<br>Водного
Кодексу</a></b></font><br><br>
                                                  <font><b><a href="/logic.php?ist=1&ref=2" id="knc1"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                              onmouseout="javascript:mouseOutl(this.id)">Оперативні дані</a></b></font><br><br>
                                                  <font><b><a href="/logic.php" id="knc2"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                              onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font><br><br>


                    </td>
                    <td style=" background: white;">
                    <div align="center">
                    <b>Сторінка знаходиться на етапі розробки.<br>Якщо натиснути на кнопку -
                                                            буде відображено дані інформації щодо водного
ресурсу</b><br>
                    <img src="/images/uch1.jpg">
                    </div>

                    </td><td style=" background: white;"></td>
                                                  <?
                                                  break;
                                        //+++++++++++++++++++++++++++++++++++++++++++++++++Нажата кнопка
про правонар. Водн. Код.
                                        case ($_GET['ref']==1):
```

```
                                                    ?>
                                                    <font><b><a href="/logic.php?ist=1" id="knc0"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                    onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font><br><br>


                                </td>

                                    <td align="center" valign="top" style="background: white;">
                                <div align="right">
                                                    <? // Центральная часть

                                                    if($_GET['an']!=1){?>
                                                        <br><div align="center"><b>Сторінка знаходиться на етапі
розробки.<br>Якщо натиснути на кнопку,то
                                                        буде відображено дані<br>загальної
інформації</b></div><br><br>

                                                    <font><b><a href="/logic.php?ist=1&ref=1&an=1"
id="knc1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                    onmouseout="javascript:mouseOutl(this.id)">Докладніше</a></b></font><br><br>
                                                    <?
                                                    }else{
                                                    ?>
                                                    <font><b><a href="/logic.php?ist=1&ref=1"
id="knc1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                    onmouseout="javascript:mouseOutl(this.id)">Сгорнути</a></b></font><br><br>

                                                    <div align="center"><?OutRecords();?></div><?
                                                    };
        ?>
        </div>
        </td> <td style=" background: white;"></td>

                                                    <?
                                                    break;
                                                //++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                                // ОСТАЛЬНОЕ
                                                // Выбрана мониторинговая информация
                                                case ($_GET['ref']==2):
                                                    ?>
                                                    <font><b><a href="/logic.php?ist=1" id="knc0"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                    onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font><br><br>


                                </td>
                                <!---
                                Центральная часть
                                //-->
                                <td align="center" valign="top" style="background: white;">


                                <div align="right">
                                                    <?
                                                    if($_GET['an']!=1){?>
                                                        <br><b><div align="center">Сторінка знаходиться на етапі
розробки.<br>Якщо натиснути на кнопку,то
                                                        буде відображено дані<br>моніторінгової
інформації</b></div><br><br>
                                                    <font><b><a href="/logic.php?ist=1&ref=2&an=1"
id="knc1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                    onmouseout="javascript:mouseOutl(this.id)">Докладніше</a></b></font><br><br>
                                                    <?
                                                    }else{
                                                    ?>
                                                    <font><b><a href="/logic.php?ist=1&ref=2"
id="knc1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                    onmouseout="javascript:mouseOutl(this.id)">Сгорнути</a></b></font><br><br>
                                                    <div align="center">
                                                    <?
                                                    $L=SELRecords(); ?></div> <?

                                                    if($L==1){          global $SUM_R;global $MAX_R;global
$MIN_R;global $DATAREQUEST;

                    ?>
```

```
<br><br><br>
<div align="left">
<font size="2"><b>Індекс забруднення води</b></font>
<br>
<font size="2"><b><i>ІЗВ=С/ГДК*n</i></b></font>
</div><br>
<?
// ПДК заносится в БД или экологический паспорт предприятия и извлекается оттуда.
//Обращаемся к БД за параметрами ПДК для каждого из 15 наблюдаемого параметра
$SQL="SELECT Pole1,Pole2,Pole3,Pole4,Pole5,Pole6,Pole7,Pole8,
Pole9,Pole10,Pole11,Pole12,Pole13,Pole14,Pole15 FROM vodaORG WHERE IDORG='".$IDORGSELECTED."';";
$SQL=SQLWork('localhost','root','password','vodaT',$SQL);
$SQL=mysql_fetch_row($SQL);
?>
<div align="center"> <b>Гранично допустимі концентрації згідно<br>речовин моніторінгу</b>
<br>
<table border="1">
<tr>
<?
$i=0;
while($i<count($SQL)):
?>
<td align="center" valign="bottom">
<b><?echo Titul($i);?></b>
</td>
<?
$i++;
endwhile;
?>
</tr>

<tr>
<?
$i=0;
while($i<count($SQL)):
?>
<td align="center" valign="bottom">
<font size="3"><?echo $SQL[$i];?></font>
</td>

<?
$i++;
endwhile;
?>

</tr>
</table><br><br></div>
<?
$i=0;$L1=null;$SU=null;
while($i<count($SUM_R)):
$SU[$i]=$SUM_R[$i]/$SQL[$i];
$L1[$i]=Titul($i);
$i++;
endwhile;
Graph($SU,"Індекс забруднення води",$L1);
?>
<br>
<div align="center">
<font size="2"><b>Варіація забруднення води</b></font>
</div><br>
<?
$i=0;
while($i<count($SUM_R)):
$SU[$i]=$MAX_R[$i]-$MIN_R[$i];
$i++;
endwhile;
Graph($SU,"Варіація забруднення води",$L1);

// Вывод графика
```

```
                                                                                ?>
                                                                                <?
                                                                                global $SET1;
                                                                                if($SET1!=null){
                                                                                        ?>
                                                                                        <div align="center">
                                                                                <br><br><br>
                                                                                <font size="4"><b>Динаміка зміни концентрації речовини у часі</b></font>
                                                                                        <br><br><br>
                                                                                        <?
                                                                                        GetGraph($SET1,$SET2);        ?>
                                                                                        </div><br><br>
                                                                                        <?
                                                                                };
                                                                        };
                                                                };
        ?>
        </div>
        </td> <td style=" background: white;"></td>

                                                                <?
                                                                break;
                                                        };
                                                ////////////////////водные ресурсы/////////////////////////////////////////
                                                };
                                        };
                                        //-----------------------------------------------------------

        } else{
                                        ?>
                                        <td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue">
        </td>
        <td style=" background: white;"></td>
        <td style=" background: white;"></td>
                                        <?
                                        };
        ?>

        <?endWeb();?>
```

registered.php

```
        <?php
        session_start();
        setCookie('session',$_SESSION['sessionID'],time()+3600);
        session_unregister('sessionID');
        include"resource/shablon.php";
        shapka();
        ?>
        <td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue">
        </td>
        <td style=" background: white;">
        <div align="center"><font size="5">Реєстрацію завершено </font></div><br>
        <div align="center"><font><b><a href="/index1.php" id="kn1" class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font></div>
        </td>
        <td style=" background: white;">
        </td>
        <?
        endWeb();
        ?>
```

registrationNewUser.php

```
        <?
        include_once"resource/shablon.php";
        include_once"resource/library.php";
        include_once"analization/analization1.php";
        $Uir=null;
        $Uir=$_POST['Uir'];
        shapka();
        ?>
        <td height="420px" valign="top" align="left" width="20%" >
        </td>
        <td style=" background: white;">
```

```
<form name="Reestr" method="POST">
<table border="1" bordercolor="red" width="50%" align="center">
<tr>
<td align="center" valign="top" colspan="50">
<img src='/images/image1.jpg' border="3" hspace="5" vspace="5">
</td>
</tr>
<?
switch(true){
        case($Uir==null):
?>
<tr>
<td colspan="50">
<div align="center">
<b>Реєстрація нового користувача,<br>
що є носієм інформації
 </b></div>
</td>
</tr>
<tr>
<td align="center">
Ім'я<br>
<input type="text" name="It">
</td>
<td align="center">
Прізвище<br>
<input type="text" name="Ft">
</td>
<td align="center">
По батькові<br>
<input type="text" name="Ot">
</td>
<td align="center">
Логін<br>
<input type="text" name="Lt">
</td>
</tr>
<tr>
<td align="center">
Пароль<br>
<input type="text" name="Pt">
</td>
<td align="center">
Посада<br>
<input type="text" name="POt">
</td>
<td align="center">
Адреса почти<br>
<input type="text" name="At">
</td>
<td align="center">
Примітка<br>
<input type="text" name="Dt">
</td>
</tr>
<tr align="center">
<td align="center" colspan="2">
Номер носія<br>
<input type="text" name="Nt">
</td>
<td align="center" colspan="2">
Назва закладу<br>
<input type="text" name="NZt">
</td>
</tr>
<input type="hidden" value="1" name="Uir">
        <?break;
        case($Uir==1):
        //Отправка данных о пользователе на сервер.
                srand((double)microtime()*1000000);
                $ID_D=uniqid(rand(),true);
                $NN=$_POST['Nt'];
                $NNZ=$_POST['NZt'];
                $SQL="INSERT INTO vodaUsers (F,I,O,Login,Pass,Mail,Memo,Doljnost,ID,IDORG,NameORG)
VALUES('".
                        $_POST['Ft']."','".$_POST['It']."','".$_POST['Ot']."','".
                        $_POST['Lt']."','".$_POST['Pt']."','".$_POST['At']."','".
```

```
                    $_POST['Dt'].'','.$_POST['POt'].'','.$ID_D.'','.$NN.
                    '','.$NNZ.'');";
                SQLWork('localhost','root','password','vodaT',$SQL);


                //Проверка в БД существующей организации чтобы не дублировать записи.
                $SQL="SELECT * FROM vodaORG WHERE (IDORG='".$NN."');";
                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                $SQL=mysql_num_rows($SQL);
                if($SQL==0){

?>
<tr>
<td>
Номер носія
<input type="text" DISABLED value="<?$a=$_POST['Nt'];echo"$a";?>">
</td>
<td>
Назва закладу
<input type="text" DISABLED value="<?$a=$_POST['NZt'];echo"$a";?>">
</td>

<td>
Кількість постів<br>
спостереження
<input type="text"  name="Ndt">
</td>

<td>
Швидкість оновлення
<input type="text"  name="Nonv">
</td>
</tr>
<tr>
<td colspan="2">
Примітка 1
<textarea cols="50" rows="10" name="PR1">
</textarea>
</td>
<td colspan="2">
Примітка 2
<textarea cols="50" rows="10" name="PR2">
</textarea>
</td>
</tr>
<input type="hidden" value="2" name="Uir">
<input type="hidden" value="<?echo"$ID_D";?>" name="Uir2">
<input type="hidden" value="<?$a=$_POST['Nt'];echo"$a";?>"name="IDORG">
<input type="hidden" value="<?$a=$_POST['NZt'];echo"$a";?>"name="IDO">
                <?          }else{ $Uir=4;
                ?>
<input type="hidden" value="" name="Uir">
                <?          };
                break;

        case($Uir==2):
                //Отправка данных о пользователе на сервер.
                srand((double)microtime()*1000000);
                $ID_D=uniqid(rand(),true);
                $SQL="INSERT INTO vodaORG (IDORG,NameORG,Memo,Memo1,DatchN,Chastota,ID) VALUES('".
                    $_POST['IDORG'].'','.$_POST['IDO'].'','.$_POST['PR1'].'','.
                    $_POST['PR2'].'','.$_POST['Ndt'].'','.$_POST['Nonv'].'','.$ID_D.'');";

                SQLWork('localhost','root','password','vodaT',$SQL);

?>
<tr>
<td>
Назва параметру 1
<input type="text" name="P1t">
</td>
<td>
Назва параметру 2
<input type="text" name="P2t">
</td>
<td>
Назва параметру 3
```

```
<input type="text" name="P3t">
</td>
<td>
Назва параметру 4
<input type="text" name="P4t">
</td>
</tr>

<tr>
<?
$i=1;
                    while($i<=15):
                              $namq="att".$i
?>
<td align="center" valign="bottom"><b><?print Titul($i-1);?></b><br>
<input type="text" name="<?print $namq?>" >
</td>
                              <?
                              if($i%4==0){echo "</tr><tr>";};
                              $i++;
endwhile;
?>
</tr>
<input type="hidden" value="3" name="Uir">
<input type="hidden" value="<?$ID_D=$_POST['IDORG'];echo"$ID_D";?>" name="Uir1">
<input type="hidden" value="<?$ID_D=$_POST['Uir2'];echo"$ID_D";?>" name="Uir2">
                    <?
                    break;
          case($Uir==3):
                    //Отправка данных о пользователе на сервер.
                    srand((double)microtime()*1000000);
                    $ID_D=uniqid(rand(),true);
                    $test=$_POST['Uir1'];
                    $SQL="INSERT INTO vodaMemo (ID,IDORG,P1,P3,P5,P7) VALUES('".
                              $ID_D."','".$test."','".$_POST['P1t']."','".$_POST['P2t']."','".
                              $_POST['P3t']."','".$_POST['P4t']."');";
                    SQLWork('localhost','root','password','vodaT',$SQL);

                    // занесение P1-P15
                    $SQL="UPDATE vodaORG SET Pole1='".$_POST['att1']."',Pole2='".$_POST['att2']."',Pole3='".
                              $_POST['att3']."',Pole4='".$_POST['att4']."',Pole5='".$_POST['att5']."',Pole6='".
                              $_POST['att6']."',Pole7='".$_POST['att7']."',Pole8='".$_POST['att8']."',Pole9='".
                              $_POST['att9']."',Pole10='".$_POST['att10']."',Pole11='".$_POST['att11']."',Pole12='".
                              $_POST['att12']."',Pole13='".$_POST['att13']."',Pole14='".$_POST['att14']."',Pole15='".
                              $_POST['att15']."' WHERE IDORG='".$test."';";
                    SQLWork('localhost','root','password','vodaT',$SQL);
                    ?>
                    <!--
                    Вывод зарегистрированной информации----------------------------
                    //-->
                    <tr>
                    <td align="left" valign="top">
                    <b>
                    Користувача зареєстровано
                    </b><br>
                    <b>Зареєстрована інформація:</b><br>
                    <b>Особиста:</b><br>
                    <?
                    $ID_D=$_POST['Uir2'];
                    $SQL="SELECT * FROM vodaUsers WHERE (ID='".$ID_D."');";
                    $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                    $SQL=mysql_fetch_array($SQL);
                    ?>
                    <b>Ім'я: </b><?echo $SQL['I'];?><br>
                    <b>Фамілія: </b><?echo $SQL['F'];?><br>
                    <b>По батькові: </b><?echo $SQL['O'];?><br>
                    <b>Логін: </b><?echo $SQL['Login'];?><br>
                    <b>Пароль: </b><?echo $SQL['Pass'];?><br>
                    <b>Почтова адреса: </b><?echo $SQL['Mail'];?><br>
                    <b>Посада: </b><?echo $SQL['Doljnost'];?><br>
                    <b>Назва закладу:</b><?echo $SQL['NameORG'];?><br>
                    <b>Ідентифікатор: </b><?echo $SQL['IDORG'];?><br>
                    <b>Примітка: </b><?echo $SQL['Memo'];?><br>
                    </td>
                    <td>
          <b>Інформація щодо закладу роботи</b><br>
                    <?
```

```
                $ID_D=$test;
                $SQL="SELECT * FROM vodaORG WHERE (IDORG='".$ID_D."');";
                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                $SQL=mysql_fetch_array($SQL);
                ?>
                <b>Назва закладу:</b><br>
                <?echo $SQL['NameORG'];?><br>
                <b>Кількість постів спостереження:</b><br>
                <?echo $SQL['DatchN'];?><br>
                <b>Швидкість оновлення:</b><br>
                <?echo $SQL['Chastota'];?><br>
                <b>Примітка 1:</b><br>
                <?echo $SQL['Memo'];?><br>
                <b>Примітка 2:</b><br>
                <?echo $SQL['Memo1'];?><br>
                </td>
                <td>
        <?
                $SQL="SELECT P1,P3,P5,P7 FROM vodaMemo WHERE (IDORG='".$test."');";
                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                $SQL=mysql_fetch_array($SQL);
                ?>
        <b>Параметр моніторінгу 1:</b><br>
        <?echo $SQL['P1'];?><br>
        <b>Параметр моніторінгу 2:</b><br>
        <?echo $SQL['P3'];?><br>
        <b>Параметр моніторінгу 3:</b><br>
        <?echo $SQL['P5'];?><br>
        <b>Параметр моніторінгу 4:</b><br>
        <?echo $SQL['P7'];?><br>
                </td>
                </tr>
                <!--
                ----------------------------------------------
                //-->
                <tr>
                <td colspan="4" align="center">
                <input type="hidden" value="" name="Uir">
                <br>
                <font><b><a href="" id="knq1"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                        onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font>
                </td>
                </tr>
                <?
                break;
        default: null;
        };
if(($Uir!=3)&&($Uir!=4)){
?>
<tr>
<td align="center" colspan="4">
<input type="submit" value="Зареєструвати">
</td>
</tr>
<?};
if($Uir==4){
        //Комментарий о том, что пользователь зарегистрирован, но организация
        //к которой он относится уже имеется в БД.
        ?>
        <tr>
<td align="center" colspan="4">
        <b>Користувача зареєстровано</b>
        </td>
</tr>
        <?
        };
?>
</table>
</form>
</td>
<td style=" background: white;">
</td>
<?
endWeb();
?>
```

workBD.php

```
<?
session_start();
include_once"resource/shablon.php";
include_once"resource/library.php";
shapka();
?>
<td height="420px" valign="top" align="left" width="20%" style="border: 1px solid darkblue">
<font><b><a href="/index1.php" id="kn1a"class="offDecoration" onmousemove="javascript:mouseMovel(this.id)"
                                        onmouseout="javascript:mouseOutl(this.id)">Повернутися назад</a></b></font>

</td>
<td style=" background: white;">
<?
if(session_is_registered('numberBD')){
        switch(true){
                case($_SESSION['numberBD']==2):
?>
<!---
Запись оперативных данных по экологии
//-->
                                        <?
                                        switch(true){
                                                case($_POST['Zavc']!=1):
?>
<form method="POST" name="Oper">
<table align="center" border="1" >
<tr>

<td align="center" colspan="20" >
<b>Форма додавання оперативної інформації</b>
</td>
</tr>
<tr>
<td align="left" valign="bottom">
<b>Рік</b><br>
<select size="1" name="god">
<option value="11">
2011
</option>
<option value="10">
2010
</option>
<option selected value="9">
2009
</option>
</select>
</td>
<td align="left" valign="bottom">
<b>Місяць<br></b>
<select size="1" name="month">
                        <option value="1">Січень</option>
                        <option value="2">Лютий</option>
                        <option value="3">Березень</option>
                        <option value="4">Квітень</option>
                        <option value="5">Травень</option>
                        <option value="6">Червень</option>
                        <option value="7">Липень</option>
                        <option value="8">Серпень</option>
                        <option value="9">Вересень</option>
                        <option value="10">Жовтень</option>
                        <option value="11">Листопад</option>
                        <option value="12">Грудень</option>
                        </select>
</td>
<td align="left" valign="bottom">
                                                <?
                                                $SQL="SELECT IDORG,NameORG FROM vodaUsers WHERE (
ID='".$_SESSION['ses_ID']."');";
                                                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                                                $SQL=mysql_fetch_array($SQL);
                                                $SQL=$SQL['IDORG'];
                                                $pol1=$SQL;
                                                $SQL="SELECT DatchN FROM vodaORG WHERE ( IDORG='".$SQL."');";
                                                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                                                $SQL=mysql_fetch_array($SQL);
                                                $SQL=$SQL['DatchN'];
```

```
                                                $i=1;
                                                ?>
<b>Номер посту<br> спостереження<br></b>
<select size="1" name="post">
                                                <?
                                                while($i<=$SQL):
                                        ?>
                                        <option><?echo $i;?></option>
                                        <?
                                                        $i++;
                                                endwhile;
?>
</select>
</td>
<td align="left" valign="bottom">
<b>Об'єкт спостереження<br></b>
<input type="text" name="Tip">
</td>
</tr>
<tr>
<td align="left" valign="bottom">
<b>Зважені речовини<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z1">
</td>
<td align="left" valign="bottom">
<b>Кіслород<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z2">
</td>
<td align="left" valign="bottom">
<b>Сульфати<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z3">
</td>
<td align="left" valign="bottom">
<b>Хлоріди<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z4">
</td>
<td align="left" valign="bottom">
<b>Кальцій<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z5">
</td>
<td align="left" valign="bottom">
<b>ХПК<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z6">
</td>
</tr>
<tr>
<td align="left" valign="bottom">
<b>БПК-<sob>5</sob><br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z7">
</td>
<td align="left" valign="bottom">
<b>Азот амонійний<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z8">
</td>
<td align="left" valign="bottom">
<b>Азот нітрітний<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z9">
</td>
<td align="left" valign="bottom">
<b>Азот нітратний<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z10">
</td>
<td align="left" valign="bottom">
<b>Фосфор фосфатний<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z11">
</td>
<td align="left" valign="bottom">
<b>Хром-6<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z12">
</td>
</tr>
<tr>
<td colspan="2" align="center" valign="center">
<b>СПАВ<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z13">
</td>
<td colspan="2" align="center" valign="center">
```

```
<b>Фенол<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z14">
</td>
<td colspan="2" align="center" valign="center">
<b>Нафтопродукти<br>конц. мГ/дМ<sup>3</sup><br></b>
<input type="text" name="Z15">
</td>
</tr>

<?

                                                $SQL="SELECT P1,P3,P5,P7 FROM vodaMemo WHERE (
IDORG='".$pol1."');";
                                                $SQL=SQLWork('localhost','root','password','vodaT',$SQL);
                                                $SQL=mysql_fetch_array($SQL);
?>
<tr>
<td colspan="6" align="center">
<font><b>Дані загалом</b></font>
</td>
</tr>
<tr>
<td colspan="2" align="center" valign="center">
<b><font><?echo $SQL['P1'];?></font><br></b>
<input type="text" name="M1">
</td>

<td colspan="2" align="center" valign="center">
<b><font><?echo $SQL['P3'];?></font><br></b>
<input type="text" name="M2">
</td>

<td colspan="2" align="center" valign="center">
<b><font><?echo $SQL['P5'];?></font><br></b>
<input type="text" name="M3">
</td>
</tr>
<tr>
<td colspan="6" align="center" valign="center">
<b><font><?echo $SQL['P7'];?></font><br></b>
<input type="text" name="M4">
</td>
</tr>
<tr>
<td align="center" valign="center" colspan="6">
<b><input type="submit" value="Ввести"></b>
<input type="hidden" value="1" name="Zavc">
<input type="hidden" value="<?echo $pol1;?>" name="Zavc1">
</td>
</tr>
</table>
</form>
                                                <?
                                                break;
                                        case($_POST['Zavc']==1):
                                                srand((double)microtime()*1000000);
                                                $ID_D=uniqid(rand(),true);
                                                $SQL="INSERT INTO vodaOperativ (god,month,Number,Tip,".

"P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,P14,P15,ID,IDORG,edIzm) VALUES('".

$_POST['god']."','".$_POST['month']."','".$_POST['post']."','".$_POST['Tip']

."','".$_POST['Z1']."','".$_POST['Z2']."','".$_POST['Z3']."','".$_POST['Z4']

."','".$_POST['Z5']."','".$_POST['Z6']."','".$_POST['Z7']."','".$_POST['Z8']

."','".$_POST['Z9']."','".$_POST['Z10']."','".$_POST['Z11']."','".$_POST['Z12']
                                                        ."','".$_POST['Z13']."','".$_POST['Z14']."','".$_POST['Z15']
                                                        ."','".$ID_D."','".$_POST['Zavc1']."','*');";
                                                SQLWork('localhost','root','password','vodaT',$SQL);
                                                srand((double)microtime()*1000000);
                                                $ID_D=uniqid(rand(),true);
                                                $SQL="INSERT INTO vodaMemo (P2,P4,P6,P8,ID,IDORG) VALUES('".

$_POST['M1']."','".$_POST['M2']."','".$_POST['M3']."','".$_POST['M4']
                                                        ."','".$ID_D."','".$_POST['Zavc1']."');";
```

```
                                                SQLWork('localhost','root','password','vodaT',$SQL);
                        ?>
                        <div align="center">
                        <b>Дані внесено</b><br>
                        <font><b><a href="/workBD.php" id="knq111"class="offDecoration"
onmousemove="javascript:mouseMovel(this.id)"
                                onmouseout="javascript:mouseOutl(this.id)">Внести дані щє раз</a></b></font></div>
                                        <?
                                        break;
                                default:null;
                                };
                                break;
                        case($_SESSION['numberBD']=3):
                                //Вошел пользователь специализированном БД не относящейся к мониторингу водных
                                //ресурсов.
                                break;
                        default:null;
                };
        };
        ?>
        </td><td style=" background: white;"> </td><?endWeb();?>
```

ДОДАТОК В

Електронні плакати

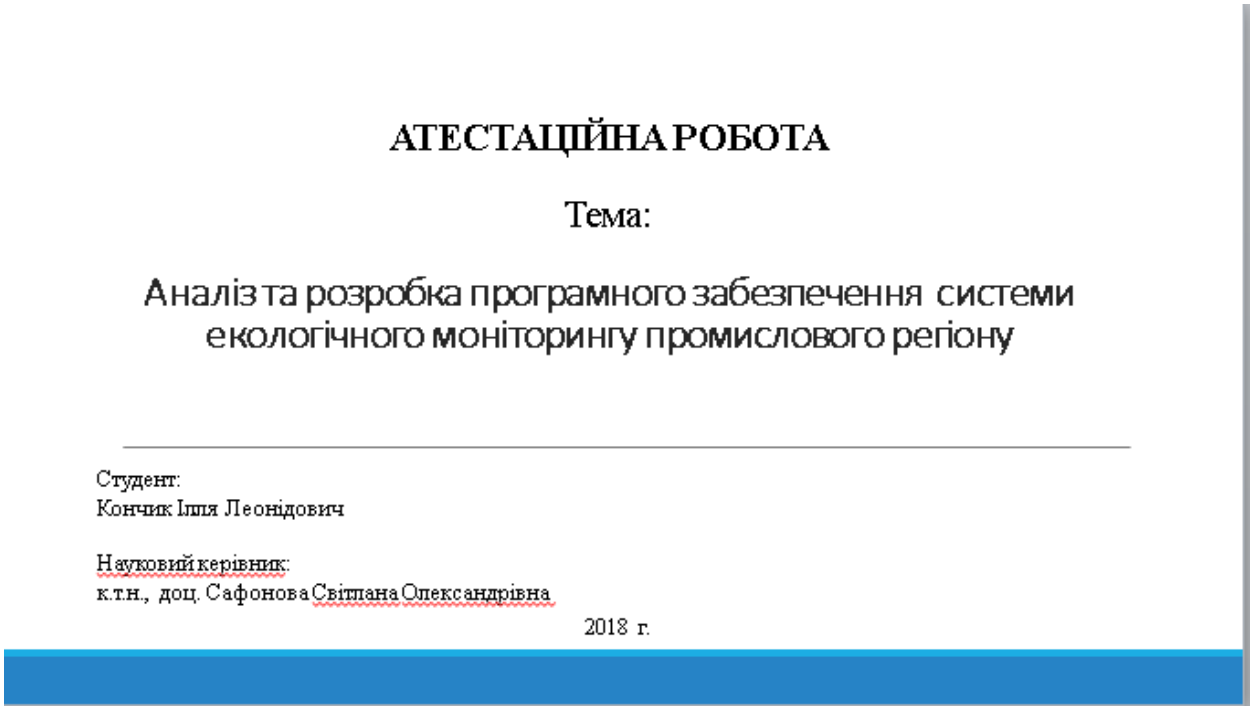


Рисунок В.1- Слайд №1

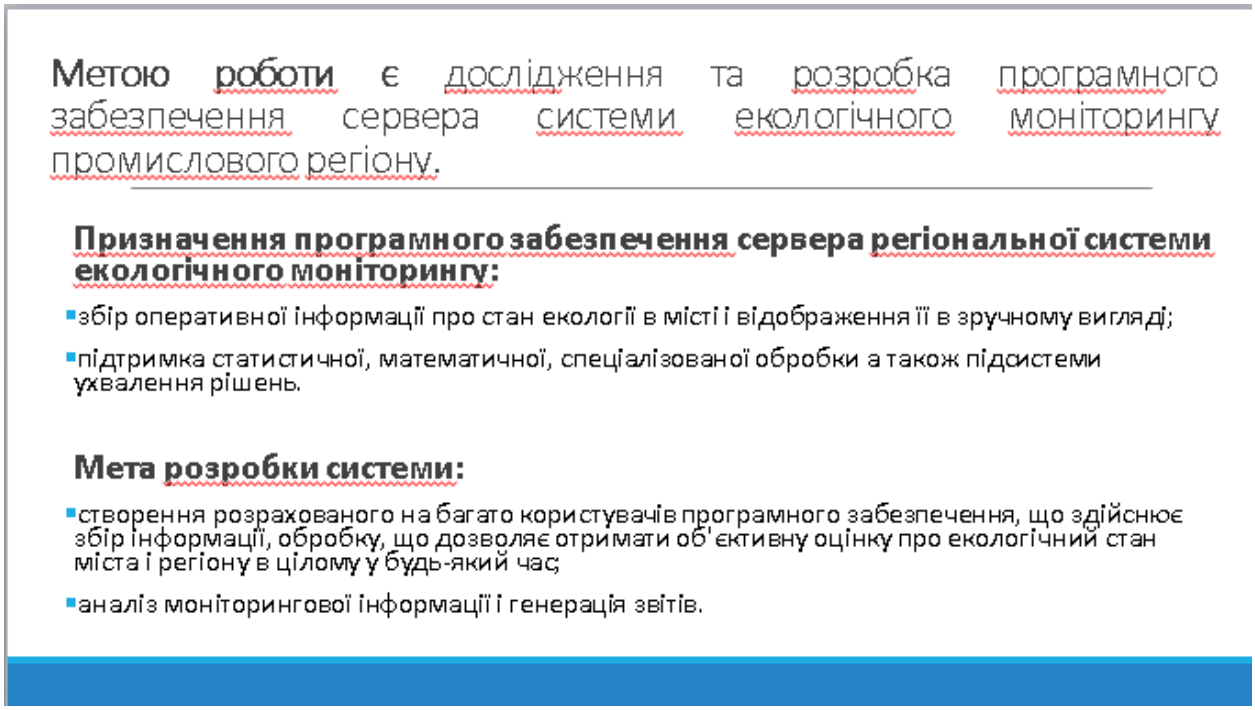


Рисунок В.1- Слайд №2

Рисунок В.1- Слайд №3

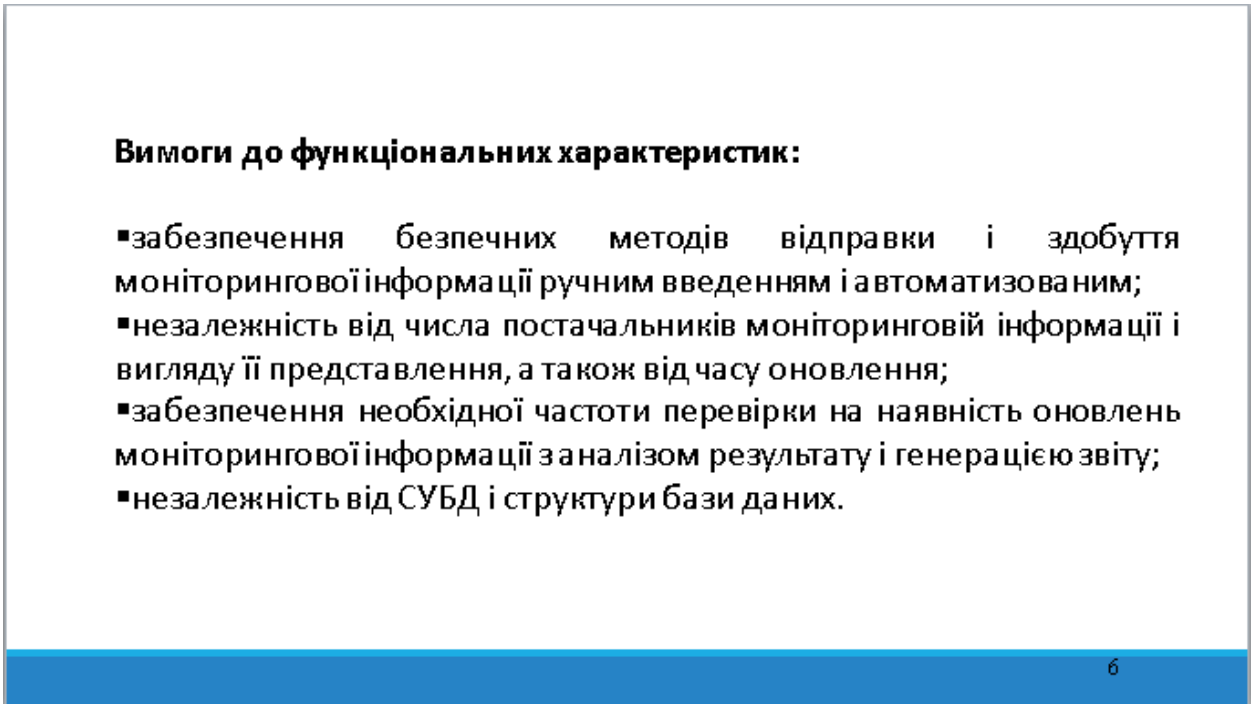


Рисунок В.1- Слайд №4

Постановка завдання до магістерської роботи

Програмне забезпечення сервера регіональної системи екологічного моніторингу повинне відповідати наступним вимогам:

- стежити, щоб моніторингова інформація була коректно і вчасно отримана від заданого постачальника інформації;
- стежити за правами доступу до бази даних користувачів сервера;
- обробляти і аналізувати наявну інформацію в базі даних і вчасно повідомляти про виникнення виняткових ситуацій;
- блокування небезпечних користувачів, шпигунських програм, вірусів, що намагаються дістати доступ до бази даних;
- перевіряти аутентифікацію кожного користувача і чітко розмежовувати його права доступу в системі;
- перевірка серверів на аварійні завершення;
- забезпечення адміністративного режиму;
- підтримка автоматичної підсистеми ухвалення рішень на основі наявної моніторингової інформації з генерацією звіту, що містить прогноз і результати обробки, аналізу.

Рисунок В.1- Слайд №5

ВИБІР ВАРІАНТІВ ВИКОНАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СЕРВЕРУ

Існує декілька варіантів створення такого роду програмного забезпечення:

- Розробка програмного забезпечення на основі технологій ASP.NET, IIS, C#,xml,sql2000,ajax,css,uml.
- Розробка програмного забезпечення на основі технологій RUBY,SQL,HTML,CSS,JAVAScript,UML.
- Розробка програмного забезпечення на основі технологій Python, SQL,HTML,CSS, JAVAScript,UML.
- Розробка програмного забезпечення на основі технологій PERL,XML,HTML,MYSQL,AHACHE,AJAX, JAVAScript,CSS,UML.

Найбільш прийнятний варіант рішення поставленої задачі виходячи з вимог проекту, а також достоїнств і недоліків вхідних в реалізацію технологій. Мова PHP найбільш кращим чином личить для реалізації відносно невеликих, швидких по продуктивності проектів в коротких проміжок часу. Проте має недолік пов'язаний із зайвою простотою реалізації типів і класів, що при побудові великих проектів виявить труднощі в більшому написанні коду, чим з "зв'язкою" з запропонованою в пункті 1.

Рисунок В.1- Слайд №6

Рисунок В.1- Слайд №7

Рисунок В.1- Слайд №8

Рисунок В.1- Слайд №9

Рисунок В.1- Слайд №10

Рисунок В.1- Слайд №11

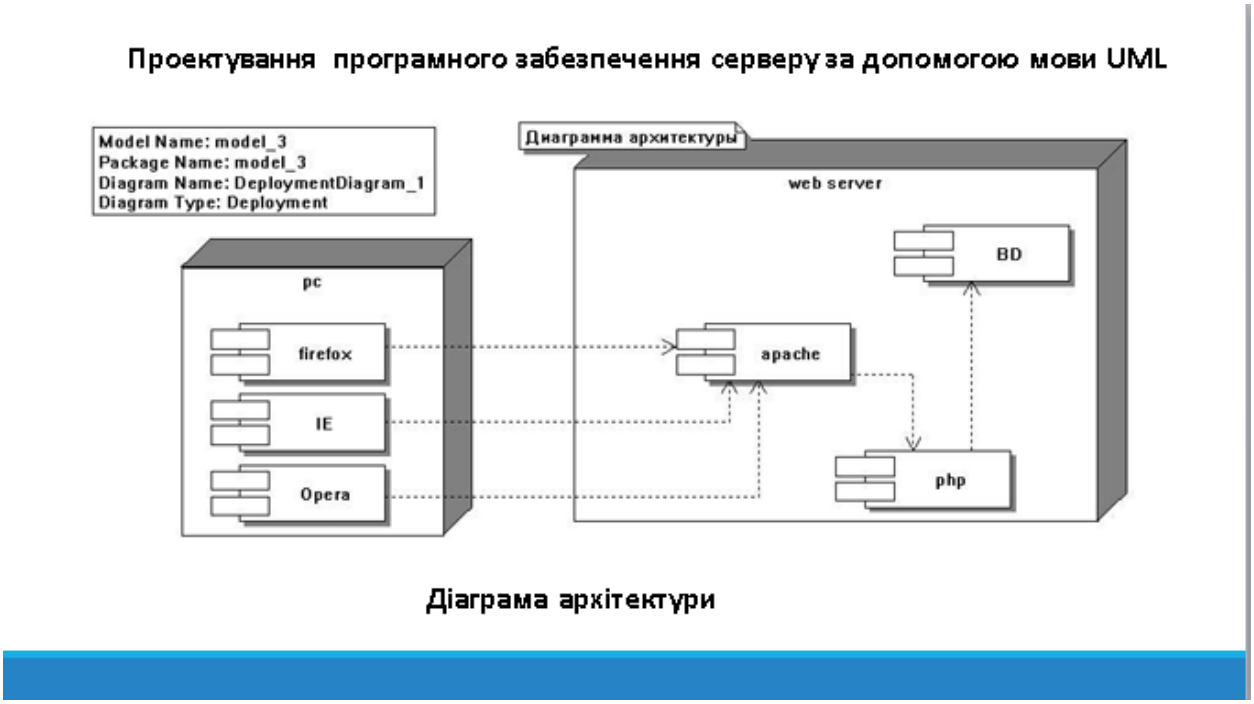


Рисунок В.1- Слайд №12

Рисунок В.1- Слайд №13

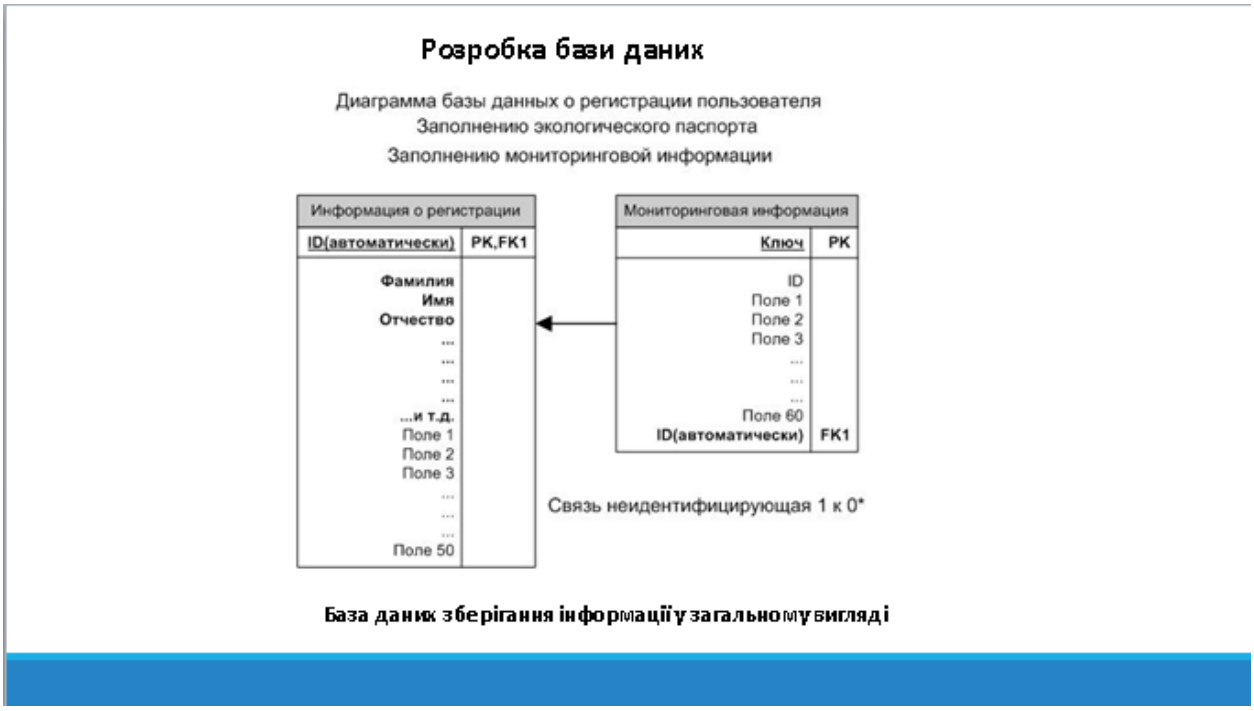
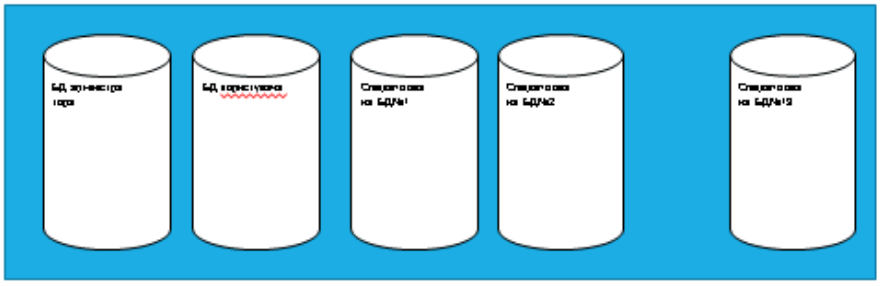


Рисунок В.1- Слайд №14

Розробка бази даних

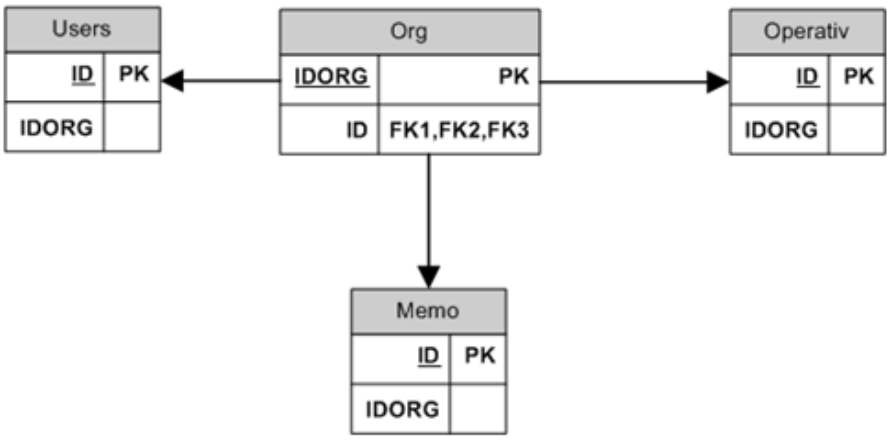


Структура бази даних програмного забезпечення серверу

Кожному постачальникові інформації виділено свою базу даних (згідно завдання маємо 13 постачальників (носіїв) інформації) та ще дві бази даних - база даних користувачів сайту (для реєстрації та доступу до ресурсів всіх користувачів, які не є носіями моніторинговій інформації), база даних адміністратора.

Рисунок В.1- Слайд №15

Розробка бази даних



Структура спеціалізованої бази даних у загальному вигляді

Рисунок В.1- Слайд №16

Рисунок В.1- Слайд №17

Розробка програмного забезпечення

В роботі проведений аналіз існуючих технологій розробки програмних систем і розробка по декількох з них, де враховувалися різні параметри: тип інформації, безпека, особливості розробки програмного забезпечення, надійність і здатність відповідати необхідним характеристикам.

Розроблено ряд алгоритмів для забезпечення безпечної передачі інформації в системі, алгоритми статистичного та екологічного аналізу, побудови графіків, підсистеми ухвалення рішень.

Для реалізації програмного забезпечення використано Microsoft Visual Studio 2010 та бібліотеки ATL і STL.

Рисунок В.1- Слайд №18

Рисунок В.1- Слайд №19

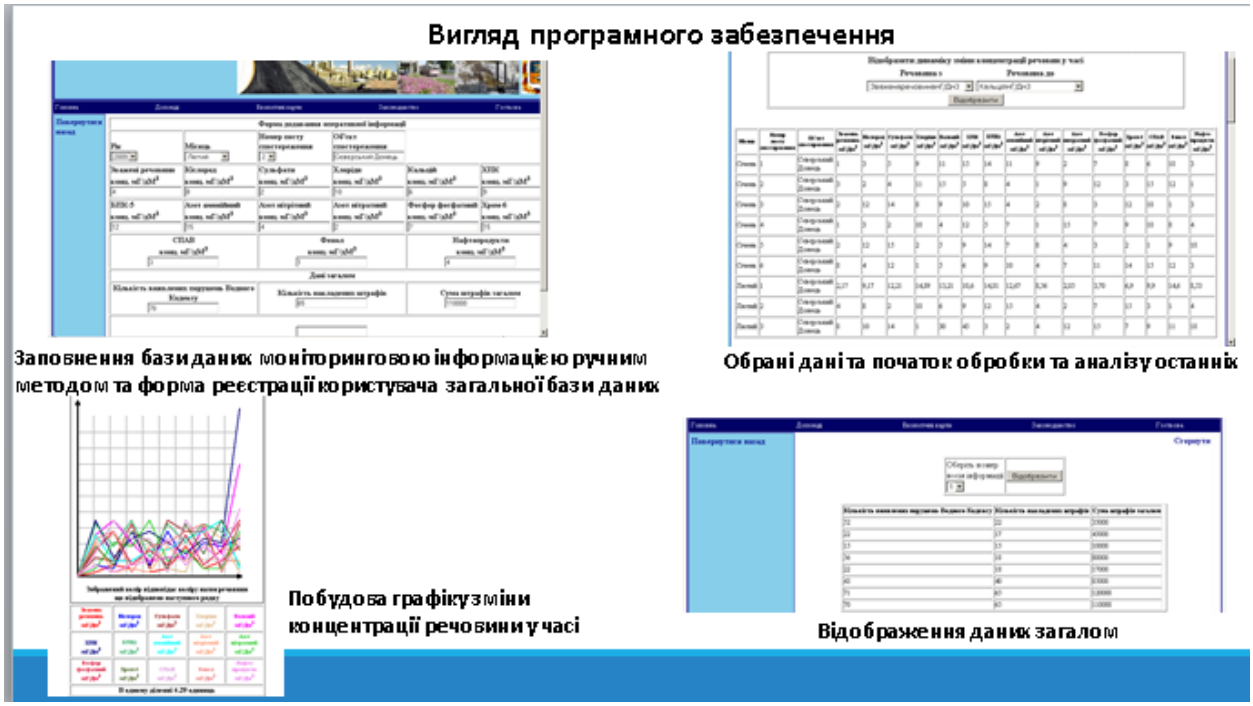


Рисунок В.1- Слайд №20

Рисунок В.1- Слайд №21

Розроблене програмне забезпечення сервера регіональної системи екологічного моніторингу відповідає вимогам, що ставилися, та реалізує наступні функції:

- стежить, щоб моніторингова інформація була коректно і вчасно отримана від заданого постачальника інформації;
- стежить за правами доступу до бази даних користувачів сервера;
- обробляє і аналізує наявну інформацію в базі даних і вчасно повідомляє про виникнення виняткових ситуацій;
- блокує небезпечних користувачів, шпигунські програми, віруси, що намагаються дістати доступ до бази даних;
- перевіряє аутентифікацію кожного користувача і чітко розмежовує його права доступу в системі;
- перевіряє сервери на аварійні завершення;
- забезпечує адміністративний режим;
- підтримкує автоматичну підсистему ухвалення рішень на основі наявної моніторингової інформації з генерацією звіту, що містить прогнозі результати обробки, аналізу.

Результатом роботи є працездатний тестовий сервер регіональної системи екологічного моніторингу, що забезпечує реалізацію вимог завдання.

Рисунок В.1- Слайд №22