

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки _____
(шифр і назва)
Спеціальність Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
І.С. Скарга-Бандурова
« _____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Маланчак Данилу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмні засоби обробки інформації про субсидії
керівник роботи Шумова Л. О., к.т.н.
затверджені наказом вищого навчального закладу від "13" 05. 2019 р. №83/15.15
2. Термін подання студентом роботи 10.06.2019
3. Вихідні дані до роботи матеріали переддипломної практики
4. Зміст розрахунково-пояснювальної записки
Аналіз предметної області та постановка задачі, Вибір засобів для розробки,
Реалізація програмних засобів обробки інформації про субсидії,
Охорона праці та безпека в надзвичайних ситуаціях.
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 та безпека в надзвичайних ситуаціях	ст.викл. Критська Я.О.		

7. Дата видачі завдання 16.05.2019

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Ознайомлення з предметною областю	16.05 – 18.05	
2	Аналіз існуючих аналогів	19.05 – 21.05	
3	Вибір засобів для розробки	22.05 – 25.05	
4	Розробка основних модулів	26.05 – 03.06	
5	Розробка розділу «Охорона праці та безпека в надзвичайних ситуаціях»	04.06 – 08.06	
6	Оформлення пояснювальної записки	09.06 – 10.06	

Студент

_____ (підпис)

Маланчак Д.О.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Шумова Л.О.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка дипломного проекту складається з чотирьох розділів, містить 117 сторінки, 11 рисунків, 12 таблиць, 25 джерел.

Метою виконання даної дипломної роботи є створення інформаційно-комп'ютерної системи для автоматизації засобів обробки інформації про субсидії. Досягнення поставленої мети дозволить уникнути дублювання даних, прискорить пошук потрібної інформації, підвищить ефективність обробки інформації про субсидії.

У роботі поставлені та вирішені задачі:

1. Опис предметного середовища - Особливості роботи відділу субсидій.
2. Аналіз системи управління і організації баз даних.
3. Проектування інформаційної системи контролю та обміну даними по субсидіях.
4. Розробка підсистеми "Контроль доходів", що призначена для автоматизації контролю та обміну інформацією з податковою інспекцією.

ІНФОРМАЦІЙНА СИСТЕМА, ЖИТЛОВІ СУБСИДІЇ, БАЗА ДАНИХ, СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ, SQL.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Інформаційні системи в сучасній економіці	10
1.2 Особливості роботи відділу субсидій	11
1.3 Системи управління і організації баз даних.....	12
1.4 Функції і структура СУБД	15
1.4.1 Функція управління буферами оперативної пам'яті.....	17
1.4.2 Функція управління транзакціями	17
1.4.3 Функція журналізації.....	17
1.4.4 Функція підтримки мов БД	20
1.5 Вибір засобів для роботи з БД.....	21
1.6 Вимоги до технічного завдання.....	24
2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ І ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ.....	25
2.1 Огляд основних завдань	25
2.2 Архітектура баз даних	25
2.3 Опис структури моделі підсистеми.....	28
2.4 Визначення функціональної структури системи	29
2.5 Вибір засобів і методів для реалізації системи	30
2.6 Опис структури таблиць підсистеми	30
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
3.1 Розробка інтерфейсу підсистеми "Контроль доходів"	34
3.1.1 Вибір інструментального засобу програмування	34
3.1.2 Візуальне середовище розробки.....	35
3.1.3 Швидкість роботи компілятора і швидкодія програм.....	36
3.2 Опис програми.....	37
3.2.1 Підпрограма відкриття таблиці	37

3.2.2 Підпрограма створення файлу звіту.....	39
3.2.3 Підпрограма формування звіту по субсидіях.....	40
3.2.4 Підпрограма відкриття довідника	41
3.2.5 Підпрограма формування вибірки за умовою.....	42
4 НОРМИ І ВИМОГИ ОХОРОНИ ПРАЦІ НА РОБОЧОМУ МІСЦІ	48
4.2 Загальні положення	48
4.3 Аналіз стану умов праці.....	49
4.3.1 Вимоги до приміщень	50
4.3.2 Вимоги до організації місця праці	51
4.4 Аналіз потенційних небезпечних і шкідливих виробничих факторів при роботі з персональним комп'ютером.....	53
4.5 Заходи з охорони праці	54
4.5.1 Організація робочого місця з ПК.....	54
4.5.2 Електробезпека	54
4.5.3 Розрахунок захисного заземлення	55
4.6 Заходи, що забезпечують виробничу санітарію та гігієну праці.....	57
4.6.1 Мікроклімат.....	57
4.6.2 Освітлення	58
4.7 Рекомендації щодо пожежної безпеки	60
ПЕРЕЛІК ПОСИЛАНЬ	65
ДОДАТОК А Лістинг програми АРМ "Субсидії"	68
ДОДАТОК В Презентація	113

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

- AIC - автоматизована інформаційна система
- БД - бази даних
- ДПІ - Державна податкова інспекція
- ЕОМ - електронна обчислювальна машина
- ЖКП - Житлово-комунальні послуги
- ІС - інформаційна система
- ПЕОМ - персональна електронна обчислювальна машина
- ПЗ - програмне забезпечення
- СУБД - системи управління базами даних
- SQL - Structured Query Language (мова структурованих запитів)

ВСТУП

За останні кілька років інформація, така необхідна в різних сферах нашої діяльності, перемістилася з паперу в персональні комп'ютери. Життя людини настільки сильно насичена різного роду інформацією, що для управління нею потрібне створення величезної кількості баз і банків даних

Використання баз даних та інформаційних систем стає невід'ємною складовою ділової діяльності сучасної людини і функціонування будь-якої організації, будь-то ціла корпорація, урядове відомство, університет, банк або підприємство, що складається з однієї особи - всі вони мають набір даних, пов'язаний з їх діяльністю. Це дані про продукцію, про студентів, про рахунки, дані планування. Вся ця сукупність спеціальним чином організованих даних, що зберігаються в пам'яті обчислювальної системи і відображають стан об'єктів та їх взаємозв'язку в аналізованій предметній області, являє собою базу даних (БД) системи. Між базою даних і безпосередніми користувачами системи розташовується рівень програмного забезпечення. Для створення, ведення і сумісного використання БД багатьма користувачами існує комплекс мовних і програмних засобів - система управління базами даних (СУБД)

Оперативний доступ до поділюваних інформаційних ресурсів дозволяє отримати інтерактивний доступ до віддалених баз даних. Даний режим доступу ON-LINE дозволяє протягом секунд здійснити передачу необхідних даних.

Всі ці переваги СУБД дозволяють багаторазово збільшити ефективність роботи державних служб і установ, за родом своєї діяльності обробляють великі обсяги інформації і документації. Одним із напрямів державної діяльності, що вимагає автоматизації.

Програма житлових субсидій стала однією з найважливіших програм соціального захисту. Вона є першою в Україні адресною програмою соціального захисту, спрямована найменш захищеним верствам населення, яка зняла соціальну напругу в суспільстві в умовах підвищення цін і тарифів на житло, комунальні послуги та енергоносії. Програма житлових субсидій була заснована в

1995 році рішенням Кабінету Міністрів України для допомоги малозабезпеченим сім'ям з оплати житлово-комунальних послуг за новими - підвищеними тарифами.

Оскільки критерієм надання адресної допомоги є рівень сімейних доходів, то для правильного нарахування субсидії і щоб уникнути розтрати державних коштів необхідно контролювати правильність і достовірність наданої інформації про доходи громадян. Для цього було запропоновано здійснювати обмін інформацією з податковою інспекцією, і створення спеціалізованого стандартизованого програмного забезпечення для реалізації цих функцій.

Система автоматизації діяльності міських відділів субсидій призначена для автоматизованої обробки документів, які здаються громадянами до відділів субсидій, а також формування довідкових і звітних документів.

Метою даної роботи є розробка, впровадження та застосування системи контролю та обміну даними по субсидіях. Підсистема, що розробляється "Контроль доходів" призначена для автоматизації контролю та обміну інформацією з податковою інспекцією.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформаційні системи в сучасній економіці

В основі вирішення багатьох завдань лежить обробка інформації. Для полегшення обробки інформації створюються ІС – сукупність апаратно-програмних засобів, задіяних для вирішення деякої прикладної задачі.

У широкому розумінні під визначення інформаційної системи потрапляє будь-яка система обробки інформації.

По області застосування інформаційні системи можна розділити на системи, які використовуються у виробництві, освіті, охороні здоров'я, науці, інформаційно-довідковій, підтримки прийняття рішень. В цьому випадку до інформації може відноситися все, що заслуговує на увагу окремого користувача або підприємства, що використовує систему. Будь-яке підприємство чи організація неминує має велику кількість даних, пов'язаних з його діяльністю. Деякий набір постійних даних, які використовуються прикладними системами для будь-якого підприємства, є БД цього підприємства.

При спрощеному розгляді схеми системи баз даних можна виділити 4 основних компоненти системи:

- дані;
- апаратне забезпечення (накопичувачі для зберігання інформації, разом з приєднаними пристроями введення-виведення, контролерами пристроїв, каналами введення-виведення, а також процесор разом з основною пам'яттю, яка використовується для підтримки роботи програмного забезпечення системи);
- програмне забезпечення, (система управління базами даних, утиліти, засоби розробки додатків, засоби проектування, генератори звітів і т.д.);
- користувачі.

У зв'язку з тим, що обсяги інформації підлягають зберігання, ростуть високими темпами, необхідно розробляти нові гнучкі підходи до управління великими обсягами даних.

Забезпечення високої продуктивності функціонування інформаційної системи - одна з головних задач, і необхідно визначити ситуації, в яких доцільно переходити до використання нових програмних технологій.

В першу чергу, це ситуація зростання використання старої ІС. У цій ситуації при тій же потужності використовуваних програмних і апаратних засобів, відбувається істотне зростання прямих виробничих витрат і втрат. Зазвичай, ця ситуація супроводжується також зростанням непрямих непродуктивних витрат. На практиці це має різні прояви: від псування до втрати важливих даних, від занадто повільного обслуговування до появи відмов в їх обслуговуванні.

Зростання використання ІС викликається як мінімум одним з таких чинників:

- збільшення потоку інформації;
- зростання числа операцій;
- появанових завдань, що вимагають автоматизації;
- зростання складності вирішуваних завдань (вимога переходу до інтерактивної графіку в геоінформсистемі, вимога переходу до середовища мультимедіа в ІС, і ін.).

Важливим фактором є моральне старіння експлуатованої ІС.

Корисно якомога раніше прогнозувати такі ситуації, кваліфікувати причини їх виникнення та одразу переходити до розробки нової ІС.

1.2 Особливості роботи відділу субсидій

Швидка і безперебійна обробка значних потоків інформації і формування документації є однією з головних задач в роботі відділу субсидій. Основні етапи роботи відділу субсидій, досить легко піддаються автоматизації.

Фактично відділом субсидій здійснюються такі види діяльності:

- прийом заяв на надання субсидій від громадян;
- занесення даних відділом субсидій;

- обробка отриманої інформації;
- призначення субсидії або відмову в ній;
- внесення отриманої інформації в комп'ютер.

Відповідальність за повноту введення в ЕОМ документації відділу субсидій лежить на фахівця, який оформляє житлову субсидію.

1.3 Системи управління і організації баз даних

Традиційно обсяги інформації, з якими доводиться мати справу ІС, досить великі, а самі БД мають складну структуру, тому необхідні повномасштабні СУБД, для управління, організації та маніпуляції такими обсягами інформації.

Повномасштабна СУБД - це три інтегрованих комплексно-програмних продуктів, які забезпечують практично будь-яку потребу замовника, а так само підтримує їх комплекс послуг. Ці комплекси такі:

- комплекс БД, куди входить ядро СУБД, що забезпечує будь-які види контролю та захисту даних, і продукти, що підтримують роботу з розподіленою базою даних в різноманітній за типами комп'ютерів і мережевих засобів середовищі. Останнім часом в ядро, як правило, входять засоби паралельної роботи: прискорення роботи за рахунок автоматичного розпаралелювання дій різних видів;

- комплекс інструментів, куди входять від 10 до 20 програмних продуктів при одночасній підтримці кількох поколінь інструментальних засобів. У число інструментів входять програми, орієнтовані на різні класи користувачів: програмістів, аналітиків зі стратегічного планування автоматизації підприємства, аналітиків- "постановників завдань", користувачів, працівників офісів. Виконуються вимоги наступності: основу зазвичай складають сучасні версії тих інструментів, які отримали визнання раніше. До них додаються інструменти, що відповідають вимогам новітніх напрямків у автоматизації (об'єктно-орієнтоване проектування та ін.);

– комплекс програм, куди фірми-постачальники включають кілька десятків потужних пакетів прикладних програм, розроблених самою фірмою, і ще більше число додатків інших розробників. Банківська діяльність, фінансова діяльність, планування виробництва і управління виробництвом, геоінформаційні системи і системи навчання в середовищі мультимедіа - приклади областей, для яких повинен існувати набір готових додатків;

– комплекс послуг, куди входять послуги передпродажного консультування, консультування з питань системної інтеграції (тобто збалансованому комплексированію всіх компонент ІС починаючи з програмно-апаратної платформи), технічний супровід системи, навчання фахівців і ін.

ІС, що представляє собою програмний комплекс, завдання якого полягають у підтримці надійного зберігання БД в комп'ютері, виконанні перетворень інформації і відповідних обчислень, поданні користувачам данні зручного і легко освоюються, інтерфейс уявляє собою комп'ютеризовану ІС. Якщо говорити про переваги комп'ютеризованої системи баз даних в порівнянні з традиційним паперовим методом змісту записів, то це:

– компактність, тому що немає необхідності в багатотомних паперових картотеках;

– швидкість. Комп'ютер може вести пошук і змінювати дані набагато швидше людини;

– низькі трудовитрати. Немає необхідності в стомлюючої ручній роботі над картотекою. Механічну роботу машини завжди виконують краще;

– придатність. Точна, свіжа інформація завжди під рукою.

Ці переваги набувають ще більшого значення в багатокористувацькій середовищі, де БД більше і складніше одного користувача. Крім того, багатокористувацьке середовище має додаткову перевагу: система БД надає централізоване управління його даними (а таке управління є найбільш цінною властивістю БД). Передбачається, що при централізованому управлінні на підприємстві, яке використовує систему БД, є людина, яка несе основну відповідальність за дані підприємства. Це адміністратор даних. В його обов'язки

входить: приймати рішення, які дані необхідно вносити в базу даних в першу чергу, а також забезпечувати підтримання порядку при обслуговуванні даних і використанні їх після занесення в БД. Іншими словами, він повинен забезпечувати безпеку даних.

Переваги централізованого підходу в управлінні даними:

- можливість скорочення надмірності. У системах не використовують БД, кожен додаток має свої файли. Це часто призводить до надмірності даних, що зберігаються, тобто до марнотратства пам'яті;

- можливість усунення (до деякої міри) суперечливості;

- можливість спільного доступу до даних. Це означає можливість доступу до даних декількох існуючих додатків БД і розробки нових додатків для роботи з цими даними. Іншими словами, нові додатки можуть отримати доступ до тих же даних, і при цьому немає необхідності в створенні нових даних;

- можливість дотримання стандартів. Завдяки централізованому управлінню адміністратор БД може забезпечувати подання даних в певних стандартах. Стандарти можуть бути корпоративними, установочними, відомчими, промисловими і інтернаціональними. Стандартизація представлення даних найбільш важлива для обміну і перенесення даних між системами. Крім того, стандарти ідентифікації даних і документації важливі і для спільного використання даних, і для їх розуміння;

- можливість забезпечення цілісності даних.

Завдання цілісності полягає в забезпеченні правильності і точності даних в БД. Протиріччя між двома записами, що представляють один «факт», є прикладом нестачі цілісності (ця проблема може виникнути лише при наявності надмірності даних). Централізоване управління дозволяє уникнути подібних проблем, наскільки їх взагалі можна уникнути. Для цього адміністратор даних визначає правила цілісності, які застосовуються при будь-якій спробі виконати будь-яку операцію оновлення.

1.4 Функції і структура СУБД

Поняття архітектури і структури є одним з найважливіших в теорії БД і служить основою для розуміння можливостей сучасних СУБД. Структура БД не повинна модифікуватися до тих пір, поки зміни в реальному світі не зажадають відповідних коректив до неї для того, щоб ця модель продовжувала відповідати предметної області БД. Природно, що проектування БД починають з аналізу предметної області та виявлення вимог до неї кінцевих користувачів. Проектування, як правило, доручається адміністратору бази даних - людині добре знайомому з машинної обробкою даних. Об'єднуючи вимоги майбутніх користувачів про вміст БД, адміністратор бази даних спочатку створює узагальнений опис створюваної БД з використанням природної мови, математичних формул, графіків і інших засобів. Тільки після формування такого узагальненого уявлення адміністратор приступає безпосередньо до формалізації представлення даних в БД.

Логічно в сучасній реляційній СУБД можна виділити найбільш внутрішню частину - ядро СУБД (часто його називають DataBaseEngine), компілятор мови БД (зазвичай SQL), підсистему підтримки часу виконання, набір утиліт. У деяких системах ці частини виділяються явно, в інших - ні, але логічно такий поділ можна провести у всіх СУБД.

Ядро СУБД відповідає за управління даними у зовнішній пам'яті, управління буферами оперативної пам'яті, управління транзакціями і журналізацію. Відповідно, можна виділити такі компоненти ядра (принаймні, логічно, хоча в деяких системах ці компоненти виділяються явно), як менеджер даних, менеджер буферів, менеджер транзакцій і менеджер журналу. Функції цих компонентів між собою, і для забезпечення коректної роботи СУБД всі ці компоненти повинні взаємодіяти з ретельно продуманим і перевіреном протоколом.

Основною функцією компілятора мови БД є компіляція операторів мови БД в деяку виконувану програму. Основною проблемою реляційних СУБД є те, що

мови цих систем (а це, як правило, SQL) є непроцедурною, тобто в операторі такої мови специфікується деяку дію над БД; але ця специфікація не є процедурою, а лише описує в деякій формі умови здійснення бажаної дії. Тому компілятор повинен вирішити, яким чином виконувати оператор мови перш, ніж зробити програму. Результатом компіляції є виконувана програма, яка надається в деяких системах в машинних кодах, але більш часто в виконуваному внутрішньому машинно-незалежнім коду. В останньому випадку реальне виконання оператора проводиться із залученням підсистеми підтримки часу виконання, що представляє собою, по суті справи, інтерпретатором цієї внутрішньої мови.

Нарешті, в окремі утиліти БД звичайно виділяють такі процедури, які занадто накладно виконувати з використанням мови БД, наприклад, завантаження і вивантаження БД, збір статистики, глобальна перевірка цілісності БД і т.д. Утиліти програмуються з використанням інтерфейсу ядра СУБД, а іноді навіть з проникненням всередину ядра.

В даний час, кількість СУБД обчислюється тисячами, але їх організація і склад компонентів відповідають певному набору функцій. Якщо говорити більш детально, то до функцій СУБД відносять наступні:

- управління даними у зовнішній пам'яті;
- управління буферами оперативної пам'яті;
- управління транзакціями;
- журналізація і відновлення БД після збоїв;
- підтримка мов БД.

Перша функція включає забезпечення необхідних структур зовнішньої пам'яті як для зберігання даних, які безпосередньо входять в БД, так і для службових цілей, наприклад, для прискорення доступу до даних в деяких випадках (зазвичай для цього використовуються індекси). У деяких реалізаціях СУБД активно використовуються можливості існуючих файлових систем, в інших робота проводиться аж до рівня пристроїв зовнішньої пам'яті. Але підкреслимо, що в розвинених СУБД користувачів будь-якому разі не зобов'язані знати, чи

використовує СУБД файловою системою, і якщо використовує, то як організовані файли. Зокрема, СУБД підтримує власну систему іменування об'єктів БД.

1.4.1 Функція управління буферами оперативної пам'яті

СУБД зазвичай працюють з БД значного розміру; принаймні цей розмір зазвичай істотно більше доступного обсягу оперативної пам'яті. Зрозуміло, що якщо при зверненні до будь-якого елемента даних буде здійснюватися обмін із зовнішньою пам'яттю, то вся система буде працювати зі швидкістю пристрою зовнішньої пам'яті. Практично єдиним способом реального збільшення цієї швидкості є буферизація даних в оперативній пам'яті.

1.4.2 Функція управління транзакціями

Транзакція – це послідовність операцій над БД, розглянутих СУБД як єдине ціле. Або транзакція успішно виконується, і СУБД фіксує зміни БД, вироблені цією транзакцією, у зовнішній пам'яті, або жодне з цих змін ніяк не відбивається на стані БД. Поняття транзакції необхідне для підтримки логічної цілісності БД. Підтримка механізму транзакцій є обов'язковою умовою як для одного користувача так і для багатокористувацьких СУБД. Властивість, що кожна транзакція починається при цілісному стані БД і залишає цей стан цілісним після свого завершення, робить дуже зручним використання поняття транзакції як одиниці активності користувача по відношенню до БД. При відповідному управлінні паралельно виконуються транзакціями з боку СУБД кожен з користувачів може в принципі відчувати себе єдиним користувачем СУБД.

1.4.3 Функція журналізації

Одним з основних вимог до СУБД є надійність зберігання даних у зовнішній пам'яті. Під надійністю зберігання розуміється те, що СУБД повинна

бути в змозі відновити останній узгоджений стан БД після будь-якого апаратного або програмного збою. Зазвичай розглядаються два можливі види апаратних збоїв, які можна трактувати як раптову зупинку роботи комп'ютера (наприклад, аварійне вимкнення живлення), і жорсткі збої, що характеризуються втратою інформації на носіях зовнішньої пам'яті. Прикладами програмних збоїв можуть бути: аварійне завершення роботи СУБД (через помилки в програмі або в результаті деякого апаратного збою) чи аварійне завершення користувальницької програми, в результаті чого деяка транзакція залишається незавершеною. Першу ситуацію можна розглядати як особливий вид м'якого апаратного збою; при виникненні останній потрібно ліквідувати наслідки тільки однієї транзакції.

Зрозуміло, що в будь-якому випадку для відновлення БД потрібно мати у своєму розпорядженні деякою додатковою інформацією. Іншими словами, підтримку надійності зберігання даних в БД вимагає надмірності зберігання даних, причому та частина даних, яка використовується для відновлення, повинна зберігатися особливо надійно. Найбільш поширеним методом підтримання такої надмірної інформації є ведення журналу змін БД.

Журнал – це особлива частина БД, недоступна користувачам СУБД і підтримується з особливою ретельністю (іноді підтримуються дві копії журналу, наявні на різних фізичних дисках), в яку надходять записи про всі зміни основної частини БД. У різних СУБД зміни БД журналізуються на різних рівнях: іноді запис в журналі відповідає деякій логічній операції зміни БД (наприклад, операції видалення рядка з таблиці реляційної БД), іноді – мінімальної внутрішньої операції модифікації сторінки зовнішньої пам'яті; в деяких системах одночасно використовуються обидва підходи.

У всіх випадках дотримуються стратегії "опереджувальним" запису в журнал (так званого протоколу WriteAheadLog - WAL). Грубо кажучи, ця стратегія полягає в тому, що запис про зміну будь-якого об'єкта БД повинна потрапити в зовнішню пам'ять журналу раніше, ніж змінений об'єкт потрапить у зовнішню пам'ять основної частини БД. Відомо, що якщо в СУБД коректно дотримується

протокол WAL, то за допомогою журналу можна вирішити всі проблеми відновлення БД після будь-якого збою.

Найпростіша ситуація відновлення – індивідуальний відкат транзакції. Строго кажучи, для цього не потрібно загальносистемний журнал змін БД. Досить для кожної транзакції підтримувати локальний журнал операцій модифікації БД, виконаних в цій транзакції, і виробляти відкат транзакції шляхом виконання зворотних операцій, слідуючи від кінця локального журналу. У деяких СУБД так і роблять, але в більшості систем локальні журнали не підтримують, а індивідуальний відкат транзакції виконують по загальносистемному журналу, для чого всі записи від однієї транзакції пов'язують зворотним списком (від кінця до початку).

При м'якому збої у зовнішній пам'яті основної частини БД можуть перебувати об'єкти, модифіковані транзакціями, що не закінчилися до моменту збою, і можуть бути відсутніми об'єкти, модифіковані транзакціями, які до моменту збою успішно завершилися (через використання буферів оперативної пам'яті, вміст яких при м'якому збої пропадає). При дотриманні протоколу WAL у зовнішній пам'яті журналу повинні гарантовано перебувати записи, які відносяться до операцій модифікації обох видів об'єктів. Метою процесу відновлення після м'якого збою є стан зовнішньої пам'яті основної частини БД, яке виникло б при фіксації в зовнішній пам'яті змін всіх завершених транзакцій і яке не містило б ніяких слідів незакінчених транзакцій. Для того, щоб цього домогтися, спочатку виробляють відкат незавершених транзакцій (undo), а потім повторно відтворюють (redo) ті операції завершених транзакцій, результати яких не відображені у зовнішній пам'яті. Цей процес містить багато тонкощів, пов'язаних із загальною організацією управління буферами і журналом.

Для відновлення БД після жорсткого збою використовують журнал і архівну копію БД. Грубо кажучи, архівна копія – це повна копія БД до моменту початку заповнення журналу (мається багато варіантів більш гнучкого трактування сенсу архівної копії). Звичайно, для нормального відновлення БД після жорсткого збою необхідно, щоб журнал не пропав. Як вже зазначалося, до

збереження журналу в зовнішній пам'яті в СУБД пред'являються особливо підвищені вимоги. Тоді відновлення БД полягає в тому, що виходячи з архівної копії за журналом відтворюється робота всіх транзакцій, які закінчилися до моменту збою. В принципі, можна навіть відтворити роботу незавершених транзакцій і продовжити їх роботу після завершення відновлення. Однак в реальних системах це зазвичай не робиться, оскільки процес відновлення після жорсткого збою є досить тривалим.

1.4.4 Функція підтримки мов БД

В сучасних СУБД зазвичай підтримується єдина інтегрована мова, що містить всі необхідні засоби для роботи з БД, починаючи від її створення, і забезпечує базовий призначений для користувача інтерфейс з базами даних. Стандартною мовою найбільш поширених в даний час реляційних СУБД є мова SQL (StructuredQueryLanguage). Перелічимо основні функції реляційної СУБД, підтримувані на "мовному" рівні (тобто функції, підтримувані при реалізації інтерфейсу SQL).

Перш за все, мова SQL поєднує засоби SDL (SchemaDefinitionLanguage – мова визначення схеми) і DML (DataManipulationLanguage – мова маніпулювання даними), тобто дозволяє визначати схему реляційної БД і маніпулювати даними. При цьому іменування об'єктів БД (для реляційної БД – іменування таблиць і їх стовпців) підтримується на мовному рівні в тому сенсі, що компілятор мови SQL виробляє перетворення імен об'єктів в їх внутрішні ідентифікатори на підставі спеціально підтримуваних службових таблиць-каталогів. Внутрішня частина СУБД (ядро) взагалі не працює з іменами таблиць і їх стовпців.

Мова SQL містить спеціальні засоби визначення обмежень цілісності БД. Знову ж, обмеження цілісності зберігаються в спеціальних таблицях-каталогах, і забезпечення контролю цілісності БД проводиться на мовному рівні, тобто при компіляції операторів модифікації БД компілятор SQL на підставі наявних у БД обмежень цілісності генерує відповідний програмний код. Спеціальні оператори

мови SQL дозволяють визначати так звані представлення БД, що фактично є збереженими в БД запитами (результатом будь-якого запиту до реляційної БД є таблиця) з іменованими стовпцями. Для користувача подання є такою ж таблицею, як будь-яка базова таблиця, яка зберігається в БД, але за допомогою уявлень можна обмежити або навпаки розширити видимість БД для конкретного користувача. Підтримка уявлень проводиться також на мовному рівні. Нарешті, авторизація доступу до об'єктів БД виробляється також на основі спеціального набору операторів SQL. Ідея полягає в тому, що для виконання ператорів SQL різного виду користувач повинен володіти різними повноваженнями. Користувач, який створив таблицю БД, володіє повним набором повноважень для роботи з цією таблицею. У число цих повноважень входить повноваження на передачу всіх або частини повноважень іншим користувачам, включаючи повноваження на передачу повноважень. Повноваження користувачів описуються в спеціальних таблицях-каталогах, контроль повноважень підтримується на мовному рівні.

1.5 Вибір засобів для роботи з БД

При виборі програмних продуктів слід віддавати перевагу программам більш широкого призначення. Багато популярних повнофункціональних СУБД мають засоби як для користувачів і адміністраторів, так і для розробників. Хоча сучасні СУБД і дозволяють вирішувати широке коло завдань по роботі з БД без розробки додатків, але є випадки, коли доцільно розробити додаток. Наприклад, якщо потрібно автоматизація маніпуляцій з даними, термінальний інтерфейс СУБД недостатньо розвинений, або наявні в СУБД стандартні функції по обробці інформації не влаштовують користувача. Для розробки додатків СУБД повинна мати програмний інтерфейс, основу якого складають функції та/або процедури відповідної мови програмування. Так, СУБД Access дозволяє для програмування додатків використовувати VisualBasic .

Існуючі СУБД підтримують такі технології (і їх комбінації) розробки додатків:

- ручне кодування програм (Clipper, FoxPro, Paradox);
- створення текстів програм за допомогою генераторів (FoxApp в FoxPro, PersonalProgrammer в Paradox);
- автоматична генерація готового додатка методами візуального програмування (Delphi, Access, ParadoxforWindows).

При ручному кодуванні програмісти вручну набирають текст програм додатків, після чого виконують їх налагодження.

Використання генераторів спрощує розробку додатків, тому що при цьому можна отримувати програмний код без ручного набору. Генератори додатків полегшують розробку основних елементів додатків (меню, екранних форм, запитів), але часто не можуть повністю виключити ручне кодування.

При візуальному програмуванні додаток будується з готових «будівельних блоків» за допомогою зручного інтегрованого середовища.

Якщо основним файлом додатків є виконуваний файл (наприклад, ехе-файл), то цей додаток є незалежним додатком, який виконується автономно від середовища СУБД (FoxPro, Delphi).

У багатьох випадках додаток не може виконуватися без середовища СУБД. Виконання програми полягає в тому, що СУБД аналізує вміст файлів програми і автоматично будує необхідні виконувані машинні команди. Іншими словами, додаток виконується методом інтерпретації. Режим інтерпретації реалізований в багатьох сучасних СУБД: Access, Paradox, VisualFoxPro.

Перевагою застосування незалежних додатків є те, що час виконання машинної програми зазвичай менше, ніж при інтерпретації. Такі додатки доцільно використовувати на слабких машинах і в разі установки систем «під ключ», коли необхідно закрити програму від доробок з боку користувача.

Важливою перевагою застосування інтерпретованих додатків є легкість їх модифікації. Якщо готова програма піддається частим змінам, то для їх внесення потрібна інструментальне середовище, тобто СУБД або аналогічна середа. Для інтерпретованих додатків такий інструмент завжди під рукою, що дуже зручно. Іншою серйозною перевагою систем з інтерпретацією є те, що хороші СУБД

зазвичай мають потужні засоби контролю цілісності даних і захисту від несанкціонованого доступу, чого не скажеш про системи компілювання типу. В останніх згадані функції доводиться програмувати вручну, або залишати на совісті адміністраторів.

Різниця у виконанні незалежного додатка і виконання програми в режимі інтерпретації коливається в межах мілісекунд на користь незалежного додатка. У той же час різниця в часі підготовки додатки до його використання зазвичай становить величини порядку хвилини - годинник на користь систем з інтерпретацією.

При аналізі можливостей і експлуатаційних характеристик сучасних СУБД особливу увагу було приділено розгляду СУБД Microsoft Access 2000.

Програмний комплекс MS Office є найпоширенішим пакетом автоматизації роботи в офісі. Тому СУБД Access, що входить в комплект професійної версії комплексу, займає лідируюче положення серед СУБД для настільних систем, що функціонують під управлінням операційної системи Windows . СУБД вкрай корисна для організації документообігу, так як дозволяє виключити дублювання і прискорити обробку документів. БД в MS Access являє собою сукупність інструментів для введення, зберігання, перегляду, вибірки та управління інформацією. До цих засобів відносяться таблиці, форми, звіти, запити. У MS Access підтримуються два способи створення БД. По-перше, можна створити порожню БД, а потім додати в неї таблиці, форми, звіти та інші об'єкти. Такий спосіб є найбільш гнучким, але вимагає окремого визначення кожного елемента БД. Крім цього є можливість створити за допомогою майстра БД визначеного типу з усіма необхідними таблицями, формами і звітами. Так як MS Access містить великий вибір підготовлених баз даних, другий спосіб у багатьох випадках може виявитися кращим. В обох випадках є можливість у будь-який час змінити і розширити створену БД.

СУБД Access надає необхідні засоби для роботи з БД початківцю, дозволяючи йому легко і просто створювати, при необхідності, бази даних, вводити в них інформацію, обробляти запити і формувати звіти.

1.6 Вимоги до технічного завдання

В даний час для управління різного роду інформацією потрібне створення величезної кількості баз і банків даних різного призначення. Зараз практично будь-яке завдання пов'язана з маніпуляцією інформацією та даними.

Швидка і безперебійна обробка значних потоків інформації є однією з головних задач в роботі відділу субсидій. Програма, яка використовується в даний час для роботи відділу субсидій, морально застаріла і не відповідає існуючим вимогам. Однак прагнення до зменшення часу на виконання вимагає використання нового програмного забезпечення. Необхідно розробити тимчасову програму "Контроль доходів".

2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ І ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ

2.1 Огляд основних завдань

Основні завдання системи "Контроль доходів" обумовлюють властивості і функції реалізованої системи. Система повинна забезпечувати роботу з БД і формування документів відділу субсидій, яка може бути реалізована за допомогою різних інформаційних засобів. Завдання вибору зводиться до підбору найбільш оптимального варіанту для більш ефективного та економного рішення задачі, для цього потрібно розробити структуру системи, яка б найбільш точно відображала рішення поставленого завдання.

2.2 Архітектура баз даних

Ефективність функціонування інформаційної системи багато в чому залежить від її архітектури. У дипломному проекті необхідно створити локальний додаток, але в даний час перспективною є архітектура клієнт-сервер, тому створення додаток для цієї архітектури - може бути одним з напрямків для подальших розробок. У досить поширеному варіанті, дана архітектура передбачає наявність комп'ютерної мережі і розподіленої бази даних, що включає корпоративну БД і персональної БД. КБД розміщується на комп'ютері-сервері, ПБД розміщується на комп'ютерах співробітників підрозділів, які є клієнтами корпоративної БД.

Сервером певного ресурсу в комп'ютерній мережі називається комп'ютер (програма), керуючий цим ресурсом, клієнтом - комп'ютер (програма), яка використовує цей ресурс. В якості ресурсу комп'ютерної мережі можуть виступати, наприклад, бази даних, файлові системи, служби друку, поштові служби. Тип сервера визначається ресурсом, яким він керує.

Перевагою організації інформаційної системи по архітектурі клієнт-сервер є вдале поєднання централізованого зберігання, обслуговування і колективного доступу до загальної інформації з індивідуальною роботою користувача над персональною інформацією. Архітектура клієнт-сервер допускає різні варіанти реалізації.

Історично першими з'явилися розподілені ІВ із застосуванням файл-сервера. У таких ІС за запитами користувачів файли БД передаються на персональні комп'ютери, де і проводиться їх обробка. Недоліком такого варіанту архітектури є висока інтенсивність передачі оброблюваних даних і дуже часто передаються надлишкові дані: незалежно від того скільки записів з бази даних потрібно користувачеві, файли БД передаються цілком.

При архітектурі клієнт-сервер з використанням сервера баз даних, сервер забезпечує виконання основного обсягу обробки даних. Формуються користувачем або додатком запити надходять до сервера БД у вигляді інструкцій мови SQL. Сервер бази даних виконує пошук і витяг корисних даних, які потім передаються на комп'ютер користувача. Перевагою такого підходу в порівнянні з попереднім є помітно менший обсяг переданих даних.

Для створення і управління персональними БД і додатків, що працюють з ними, використовуються СУБД, такі як Access і Visual FoxPro фірми Microsoft, Paradox фірми Borland. Корпоративна БД створюється, підтримується і функціонує під управлінням сервера БД, наприклад, Microsoft SQL Server або Oracle Server.

Основні переваги клієнт-серверних в порівнянні з аналогічними інформаційними системами полягає в наступному. По-перше, це зниження кількості переданої по комп'ютерній мережі інформації. По-друге, перевагою архітектури є можливість зберігання правил доступу та обробки на сервері, що дозволяє уникнути дублювання коду в різних додатках, що використовують загальну БД. Крім того, будь-яка

маніпуляція з даними може бути проведена тільки в рамках цих правил. Частина коду, пов'язаного з обробкою даних, як правило, реалізується у вигляді збережених процедур сервера, що дозволяє ще більше прискорити роботу клієнтського додатку за рахунок зменшення його розмірів, а це в свою чергу означає, що вимоги до робочих станцій можуть бути не такими високими. Це в кінцевому підсумку знижує загальну вартість інформаційної системи навіть при використанні дорогої СУБД і потужного сервера БД.

По-третє, сучасні СУБД, реалізовані на платформі клієнт-сервер, мають потужні можливості управління доступу до елементів БД, резервного копіювання, архівації і паралельної обробки даних, що значно покращує роботу.

Використовуючи безліч комп'ютерів, системи на платформі клієнт-сервер розподіляють прикладну задачу по різних робочих станцій і серверів. Кожен елемент при цьому бере на себе свою частину обчислювального навантаження, використовуючи інформацію спільно з іншими комп'ютерами мережі, при цьому потужність системи підвищується без нарощування продуктивності одного окремого комп'ютера, а виходить як результат підсумовування можливостей багатьох.

Традиційно в клієнт-серверних системах використовуються дві ланки - клієнт і сервер. При такій побудові велике програмне навантаження несе на собі клієнтська частина, а сервер в основному використовують для обслуговування даних. Дволанкові системи виходять досить простими в побудові і обслуговуванні, але таку модель неможливо наростити для спільної роботи тисяч користувачів. Для забезпечення роботи при великих обчислювальних навантаженнях використовується трьох- або багатоланкова архітектура, що тягне за собою розподіл системи по додатковим ланкам, що забезпечує додаткову обчислювальну потужність.

Залежно від розмірів організації і особливостей вирішуваних завдань, ІС може мати одну з наступних конфігурацій:

- комп'ютер - сервер, що містить корпоративну і персональні бази;
- комп'ютер - сервер і персональні комп'ютери з ПБД;
- кілька комп'ютерів - серверів і персональних комп'ютерів з ПБД.

Використання архітектури клієнт-сервер дає можливість нарощування інформаційної системи підприємства, по-перше, у міру розвитку підприємства; по-друге, у міру розвитку самої інформаційної системи.

Поділ загальної БД на корпоративну БД і персональні дозволяє зменшити складність проектування БД в порівнянні з централізованим варіантом, а значить знизити ймовірність помилок при проектуванні і вартість проектування.

Найважливішим достоїнством застосування БД в інформаційних системах є забезпечення незалежності даних від прикладних програм. Це дає можливість користувачам не займатися проблемами представлення даних на фізичному рівні: розміщення даних в пам'яті, методів доступу до них і т.д.

Така незалежність досягається підтримуваним СУБД багаторівневим поданням даних в БД на логічному (призначеному для користувача) і фізичному рівні. Завдяки СУБД і наявності логічного рівня представлення даних забезпечується відділення концептуальної, понятійної моделі БД від її фізичного представлення в пам'яті ЕОМ.

2.3 Опис структури моделі підсистеми

Основу системи складають таблиці БД (рис. 2.1).

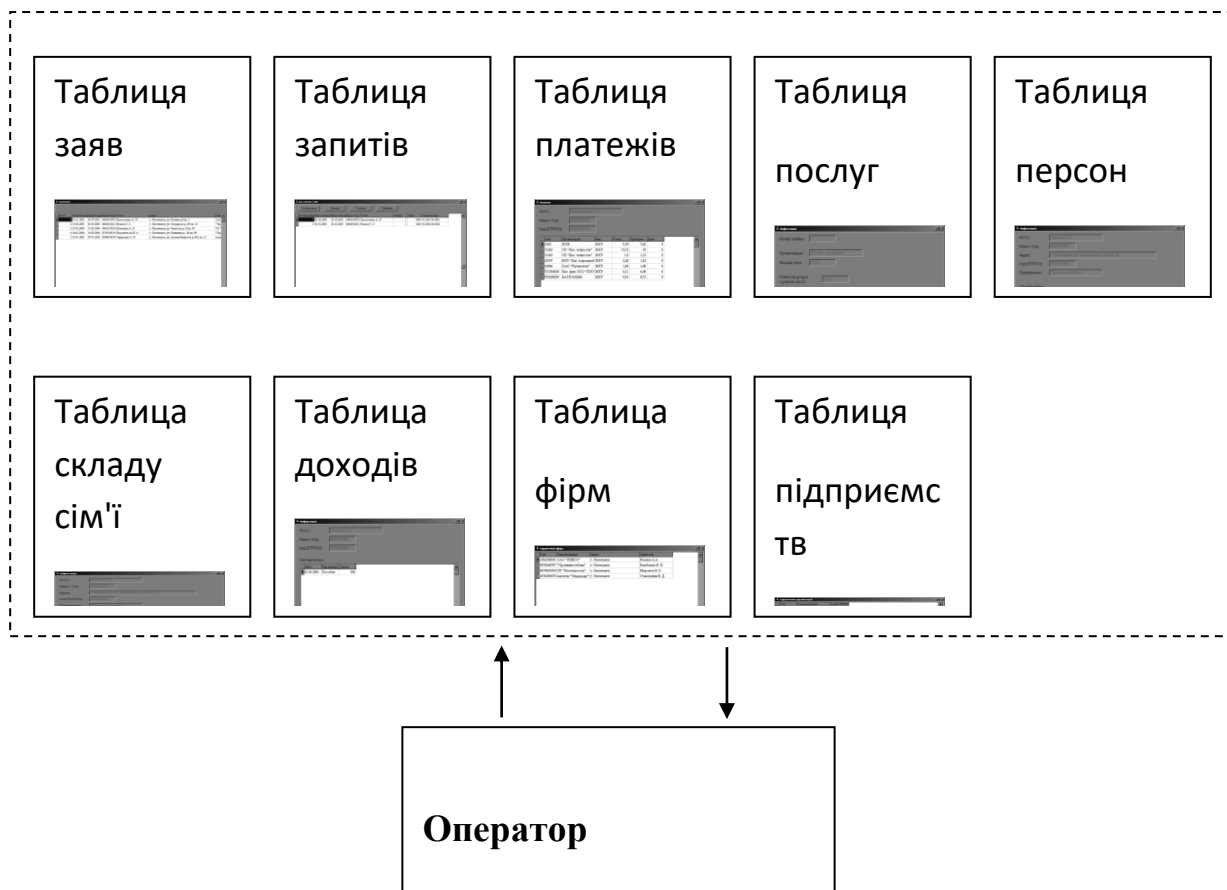


Рисунок 2.1 - Структура системи "Контроль доходів"

БД складається з таблиці персон, таблиці запитів, таблиці заяв, таблиці доходів, таблиці платежів, таблиці послуг, таблиці фірм, таблиці організацій і таблиці складу сім'ї.

2.4 Визначення функціональної структури системи

Згідно з вимогами поставленим до даної системи, вона повинна підтримувати реалізацію наступних функцій:

- відображення таблиць і довідників;
- формування звітів;
- відображення таблиць і довідників;
- формування звітів.

2.5 Вибір засобів і методів для реалізації системи

Для розроблюваної системи "Контроль доходів" оптимальним варіантом є вибір операційної системи сімейства Windows.

2.6 Опис структури таблиць підсистеми

На підставі отриманих даних можна виділити структуру даних системи "Контролю доходів". Основу системи складають робота з БД, яка складається з таблиці персон, таблиці запитів, таблиці заяв, таблиці доходів, таблиці платежів, таблиці послуг, таблиці фірм, таблиці організацій і таблиці складу сім'ї.

Структура таблиці запитів представлена в таблиці 2.1.

Таблиця 2.1 - Структура таблиці запитів

Найменування поля	Тип даних поля
код запиту	числовий
номер запиту	строковий
код персони	числовий
дата початку періоду доходу	дата
запит	логічний
відповідь	логічний
сума запиту	числовий
сума відповіді	числовий
дата запиту	дата
дата відповіді	дата

Структура таблиці заяв представлена в таблиці 2.2.

Таблиця 2.2 - Структура таблиці заяв

Найменування поля	Тип даних поля
код заяви	числовий
дата початку	дата
дата кінця	дата
код персони	числовий
код ЄДРПОУ	строковий
ЖКП	логічний
СГТТ	логічний
сума ЖКП	числовий
сума СГТТ	числовий

Структура таблиці доходів представлена в таблиці 2.3:

Таблиця 2.3 - Структура таблиці доходів

Найменування поля	Тип даних поля
код доходу	числовий
дата початку	дата
дата кінця	дата
код персони	числовий
тип доходу	дата
сума доходу	строковий

Структура таблиці платежів представлена в таблиці 2.4:

Таблиця 2.4 - Структура таблиці платежів

Найменування поля	Тип даних поля
код платежу	числовий
код персони	числовий
код організації	числовий
код послуги	числовий
сума боргу	числовий
дата платежу	дата

Структура таблиці послуг представлена в таблиці 2.5:

Таблиця 2.5 - Структура таблиці послуг

Найменування поля	Тип даних поля
код послуги	числовий
скорочене найменування	строковий
найменування	строковий
сума платежу	числовий
сума субсидії	числовий

Структура таблиці фірм представлена в таблиці 2.6.

Таблиця 2.6 - Структура таблиці фірм

Найменування поля	Тип даних поля
код фірми	числовий
особистий рахунок	строковий
найменування	строковий
адреса	строковий
директор	строковий

Структура таблиці організацій представлена в таблиці 2.7

Таблиця 2.7 - Структура таблиці організацій

Найменування поля	Тип даних поля
код організації	числовий
особистий рахунок	строковий
найменування	строковий
код ЄДРПОУ	строковий

Структура таблиці складу сім'ї представлена в таблиці 2.8

Таблиця 2.8 - Структура таблиці складу сім'ї

Найменування поля	Тип даних поля
код складу сім'ї	числовий
код персони	числовий
найменування	строковий
вид відносин	строковий
вид доходу	строковий
сума доходу	числовий

Структура таблиці персон представлена в таблиці 2.9:

Таблиця 2.9 - Структура таблиці персон представлена

Найменування поля	Тип даних поля
код персони	числовий
ідентифікаційний код	строковий
П.І.Б.	строковий
адреса	строковий
код ЄДРПОУ	строковий
код фірми	числовий

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка інтерфейсу підсистеми "Контроль доходів"

Лістинг розробленої програми представлений в додатку А.

3.1.1 Вибір інструментального засобу програмування

В якості мови програмування використовується при розробці інтерфейсу викладача, можна запропонувати кілька варіантів. Це найбільш популярні засоби розробки додатків в середовищі Microsoft Windows. такими мовами є Visual C ++, Delphi, Visual Basic for Applications і інші. У кожного з них є свої переваги і недоліки.

Як інструментальний засіб програмування для реалізації поставленого завдання було обрано засіб розробки, надане компанією Borland Inprise - Delphi. Це середовище підтримує багато сучасні технології і є практично повністю відкритою для розробника.

Вибір був обумовлений "продуктивністю" середовища - на сьогоднішній день робота в Delphi є одним з найбільш продуктивних методів створення додатків.

Загальна продуктивність будь-яких інструментів створення програмного забезпечення визначається такими найважливішими аспектами:

- якістю візуального середовища розробки;
- швидкістю роботи компілятора і швидкодією програм;
- потужністю мови програмування і його складністю;

Зупинимось докладніше на трьох перших аспектах вищевказаної схеми, які найбільше вплинули на вибір кошти програмування, і оцінимо відповідальні показники Delphi в порівнянні з іншими інструментами розробки додатків.

3.1.2 Візуальне середовище розробки

Візуальне середовище розробки, в загальному випадку, складається з трьох взаємопов'язаних компонентів: редактора, відладчика і конструктора форм. У будь-якому з сучасних інструментів прискореної розробки додатків (Rapid Application Development - RAD) повинні гармонійно взаємодіяти один з одним в процесі створення додатку. При роботі в Delphi, конструктор форм неявно генерує програмний код всіх компонентів, які містяться, і обробляються в формах. У вікні редактора в текст автоматично створеної програми можна ввести необхідні доповнення, що визначають специфічну поведінку цього додатка. Тут же, у вікні редактора, програми можуть регламентуватиме за допомогою внесення точок зупину, визначення контрольованих змінних.

У Delphi, яке було вибрано для реалізації поставленого завдання, відладчик підтримує вже весь набір функціональних можливостей, присутніх відкладчику пакета Visual Studio. До знову доданих функцій відносяться засоби вилученого налагодження, підключення процесів, налагодження пакетів і бібліотек DLL, засоби контролю значень автоматичних локальних змінних і підтримки вікна ЦП. Крім того, Delphi надає зручні засоби управління графічної середовищем налагодження. Вони дозволяють в ході налагодження розміщувати вікна у будь-якому зручному місці, а також запам'ятовувати відомості про отриманий конфігурації у вигляді параметрів настройки робочого столу. Одна з дуже зручних функцій відладчиків, яка широко поширена в середовищі інтерпретаторів (таких як Visual Basic або Java), полягає в можливості змінювати програмний текст і, отже, поведінку додатків безпосередньо в процесі його налагодження. На жаль, в середовищі компіляторів реалізація подібних функцій пов'язана з дуже великими труднощами, і тому вона відсутня в Delphi.

Конструктор форм є обов'язковою приналежністю всіх RAD - Інструменти, включаючи Delphi, Visual Basic, C ++ Builder і PowerBuilder. Класичний варіант середовища розроблення (наприклад, Visual C ++ і Borland C ++) зазвичай включає редактори діалогів, однак ці інструменти менш зручні для інтеграції в робочий потік створення додатка, ніж конструктори форм. Відсутність конструктора форм справляє помітний негативний вплив на загальні показники продуктивності розробки додатків.

3.1.3 Швидкість роботи компілятора і швидкодія програм

Швидкий компілятор дозволяє розробляти програмне забезпечення поетапно, оскільки допускає багаторазове внесення в вихідну програму невеликих змін, з наступною перекомпіляцією і тестуванням. В результаті виникає досить ефективний цикл розробки. Більш повільний компілятор змушує розробника одночасно вносити більший обсяг змін, комбінуючи кілька окремих доробок в одному циклі компіляції і налагодження. Це знижує ефективність окремих циклів розробки, що досягаються за рахунок підвищеної ефективності роботи відкомпільованих програм. У будь-якому випадку, чим швидше працює програма, і чим менше її об'єктний код, тим краще. Фактично, це - найшвидший компілятор мови високого рівня з усіх, що існують в середовищі Windows. В останні роки відзначаються помітні поліпшення в роботі компіляторів мови C ++, який традиційно вважався нешвидким в сенсі швидкості компіляції. Успіхи були досягнуті за рахунок покрокового зв'язування і різних стратегій кешування, використовуваних, зокрема, в пакетах Visual C ++ і C ++ Builder. Проте, навіть ці поліпшені компілятори мови C ++ працюють в кілька разів повільніше, ніж компілятор Delphi. Створений в Delphi об'єктний код має

ті ж показники ефективності, що і об'єктний код, створений транслятором C ++ Builder.

3.2 Опис програми

Основна програма включає в себе функції відображення таблиць БД, створення і друк звітів. Функція вибору, вироблених оператором дій покладається на форми.

У кодї програми можна виділити основний набір підпрограм:

- підпрограма відкриття таблиці;
- підпрограма створення файлу TTL;
- підпрограма формування звіту по субсидіях;
- підпрограма відкриття довідників.

3.2.1 Підпрограма відкриття таблиці

Підпрограма виробляє відкриття та відображення таблиць БД. Підпрограма реалізується рядками 344-367, 640-674, 1029-1048 в додатку А. Схема алгоритму функціонування даної підпрограми зображена на рисунку 3.1

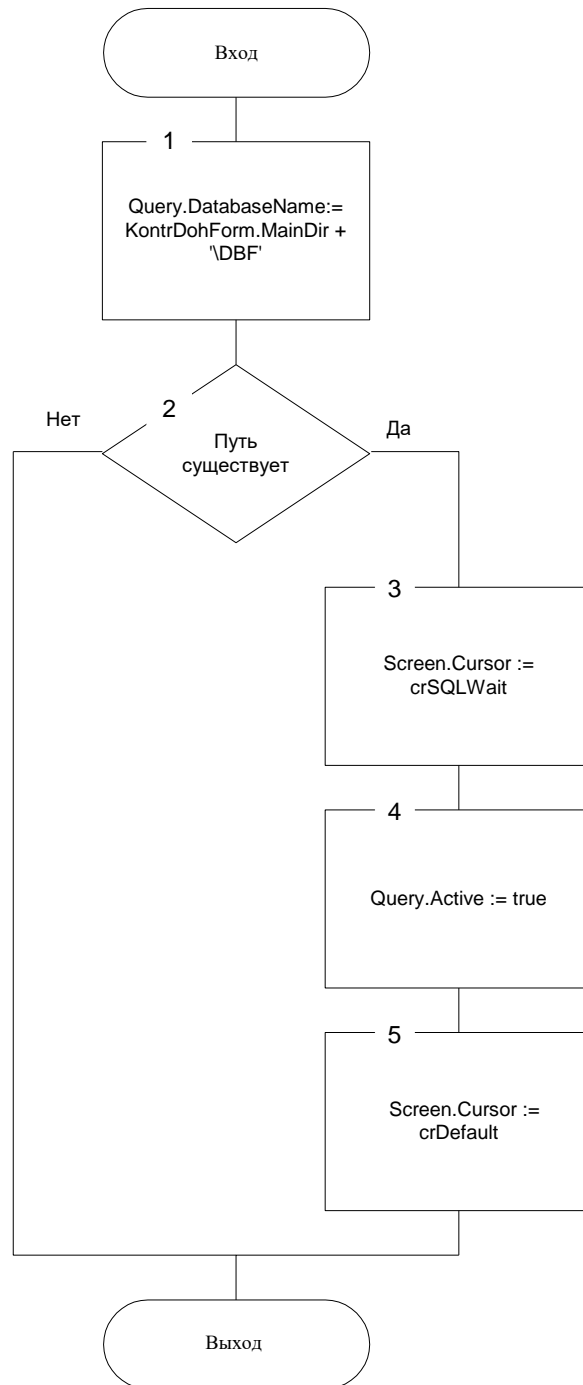


Рисунок 3.1 - Схема алгоритму функціонування підпрограми відкриття таблиці

3.2.2 Підпрограма створення файлу звіту

Підпрограма формує текстовий файл і реалізується рядками 394-472, 555-638 в додатку А. Схема алгоритму функціонування даної підпрограми зображена на рисунку 3.2.

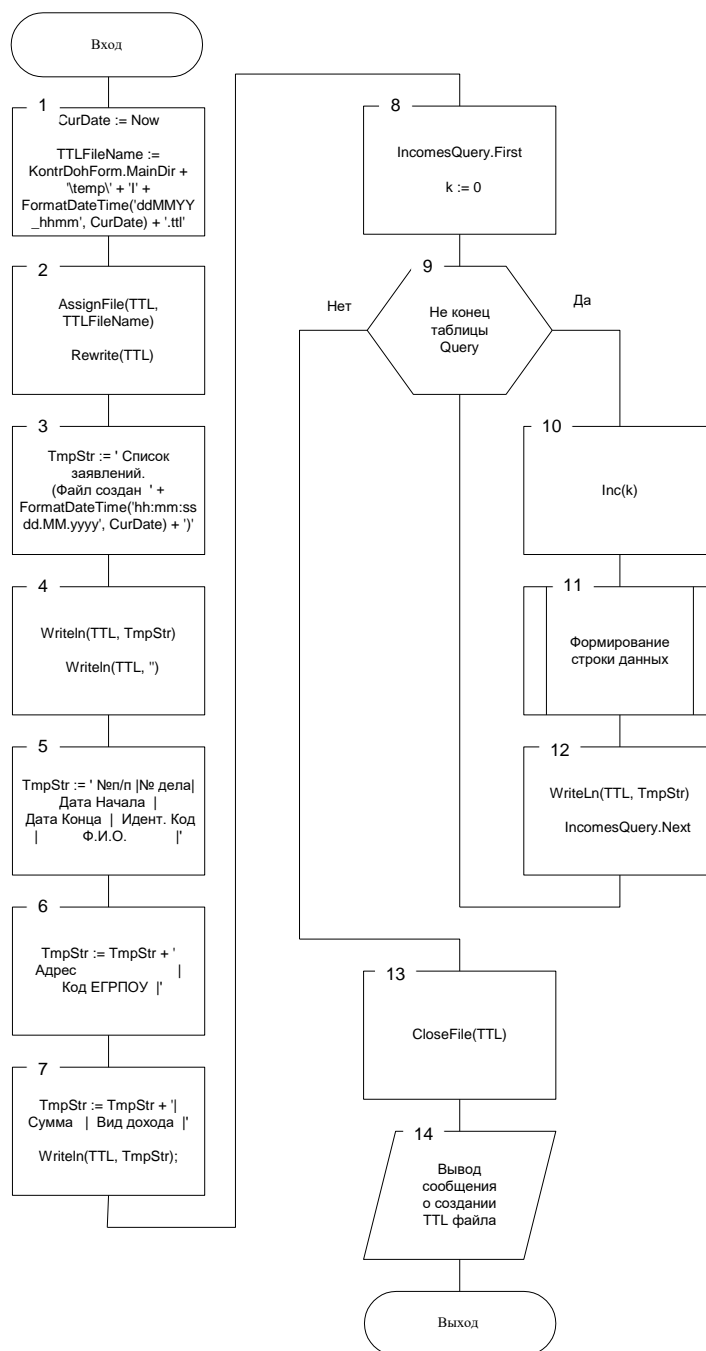


Рисунок 3.2 - Схема алгоритму функціонування підпрограми створення файлу звіту

3.2.3 Підпрограма формування звіту по субсидіях

Підпрограма виробляє формування звіту по субсидіях. Підпрограма реалізується рядками 2324-2351 в додатку А. Схема алгоритму функціонування даної підпрограми зображена на рисунку 3.3.

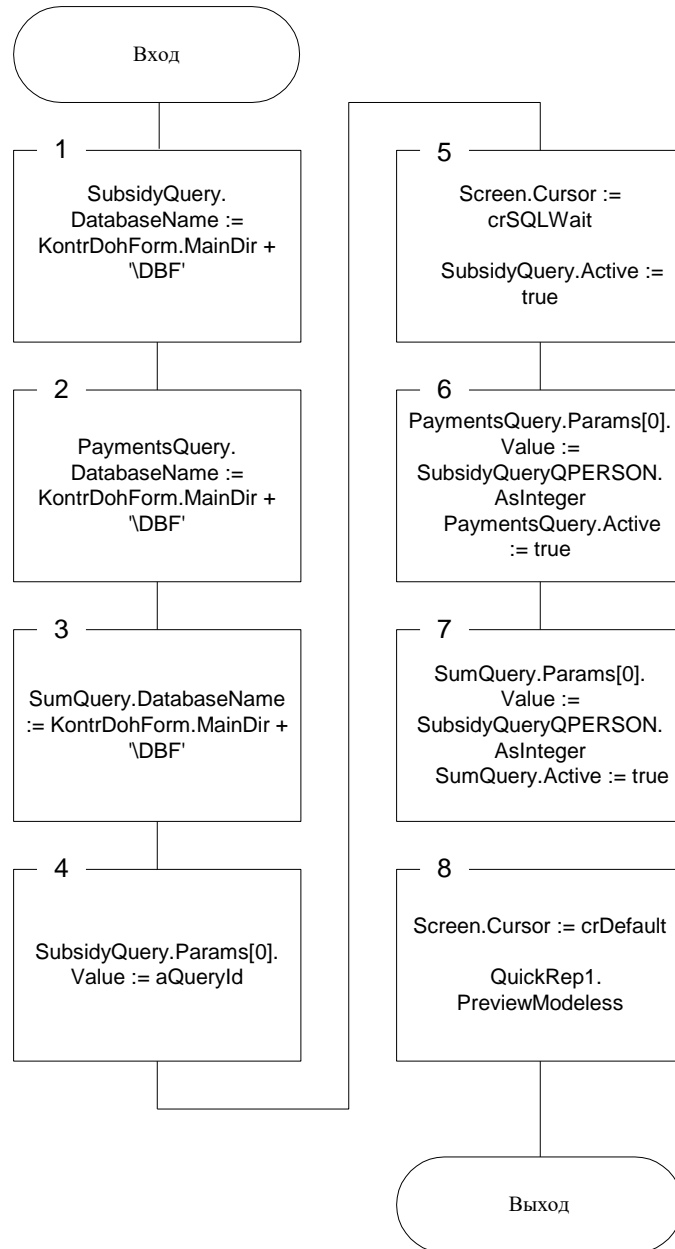


Рисунок 3.3 - Схема алгоритму формування звіту по субсидіях;

3.2.4 Підпрограма відкриття довідника

Підпрограма виробляє відкриття та відображення довідника. Підпрограма реалізується рядками 1999-2014, 2065-2082 в додатку А. Схема алгоритму функціонування даної підпрограми зображена на рисунку 3.4.

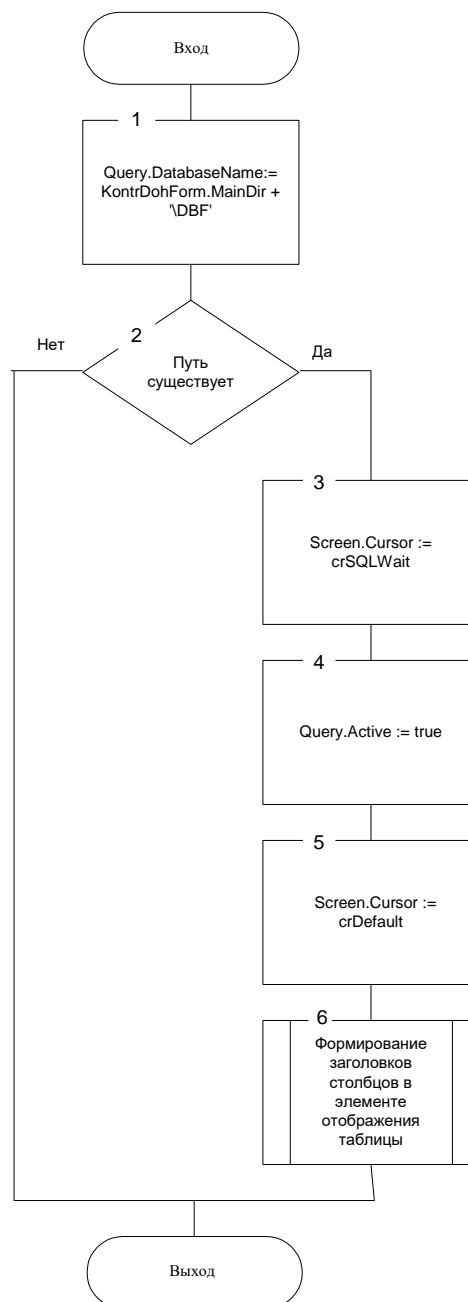


Рисунок 3.4 - Схема алгоритму функціонування підпрограми відкриття довідника

3.2.5 Підпрограма формування вибірки за умовою

Підпрограма виробляє формування вибірки за умовою. Підпрограма реалізується рядками 1217-1268 у додатку А. Схема алгоритму функціонування даної підпрограми зображена на малюнку 3.5, рисунку 3.7, рисунку 3.8, рисунку 3.9, рисунку 3.10.

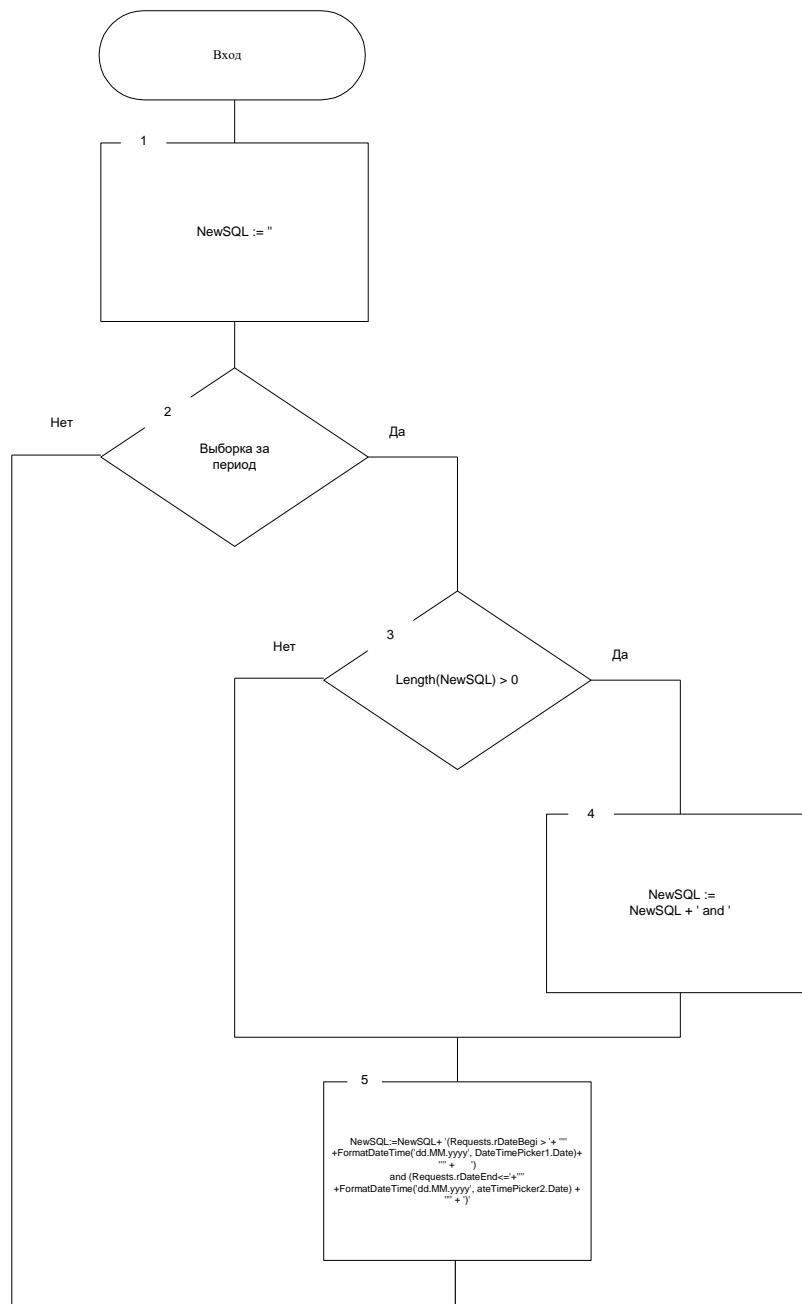


Рисунок 3.5 - Схема алгоритму формування вибірки за умовою

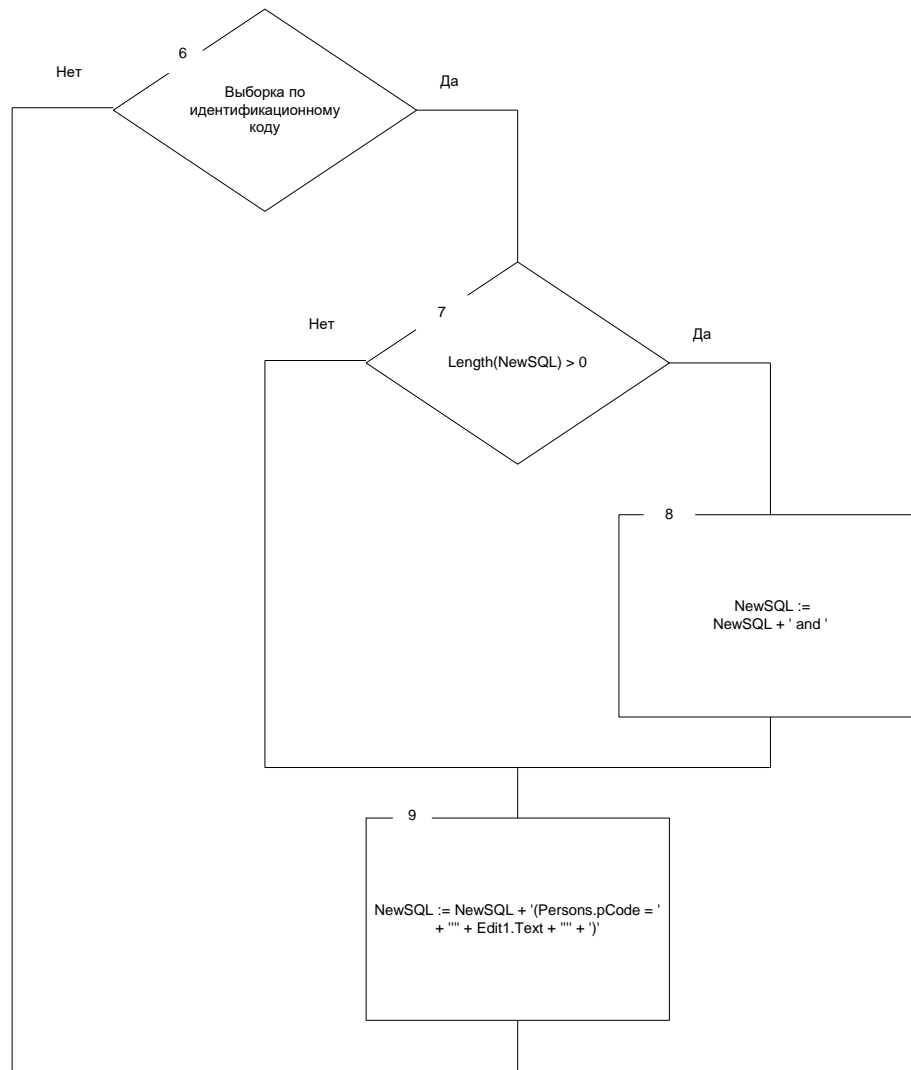


Рисунок 3.6 - Схема алгоритму формування вибірки за умовою

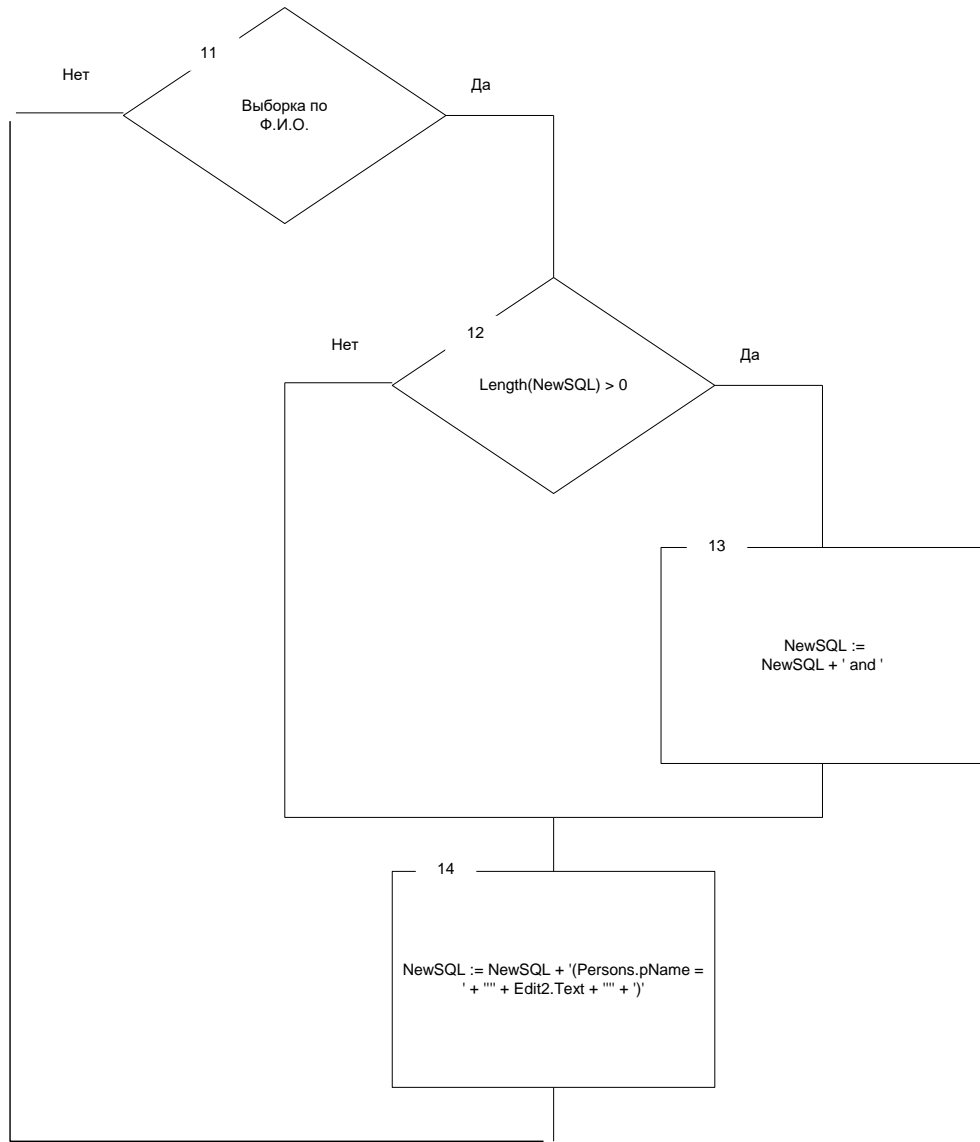


Рисунок 3.7 - Схема алгоритму формування вибірки за умовою

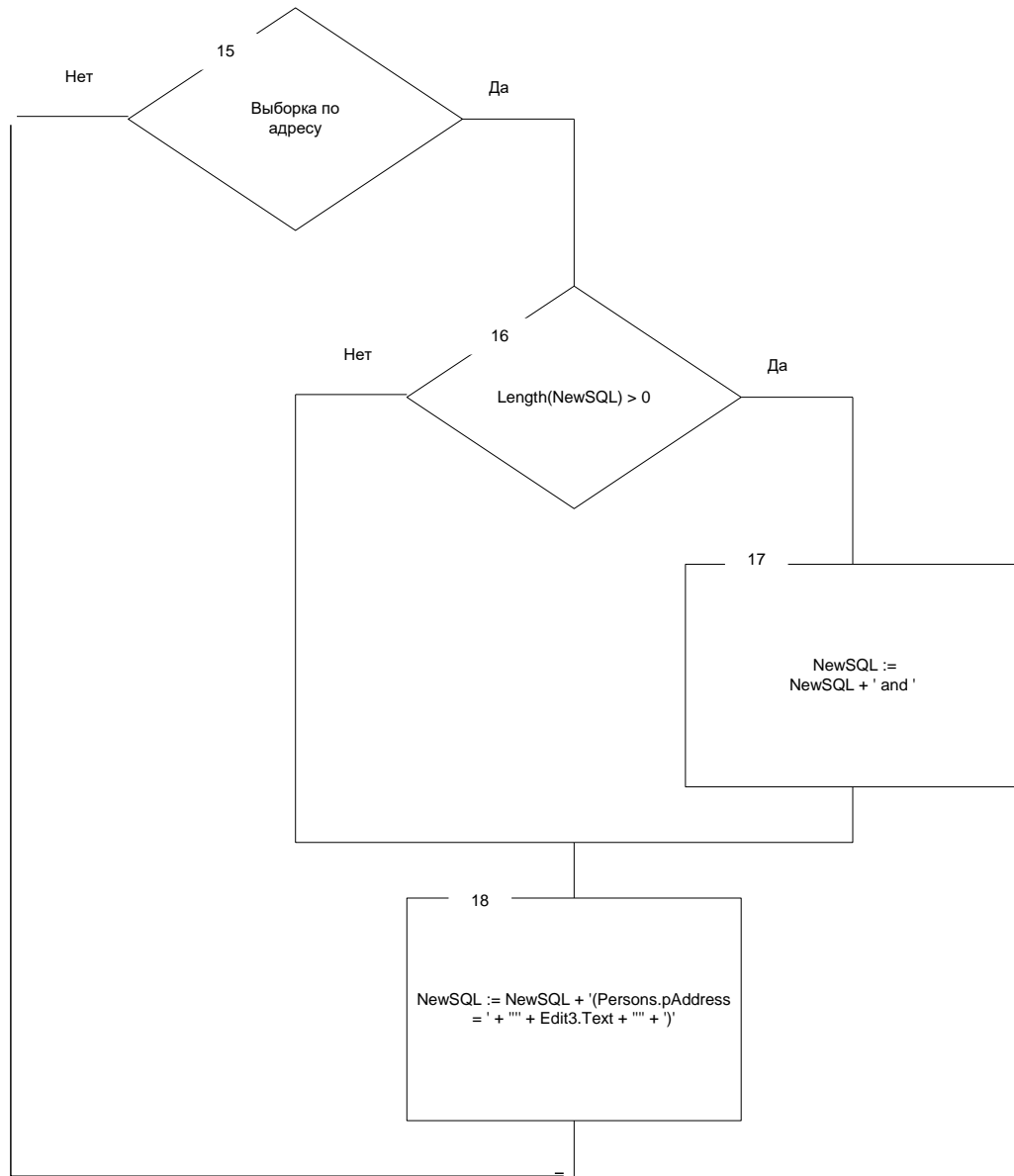


Рисунок 3.8 - Схема алгоритму формування вибірки за умовою

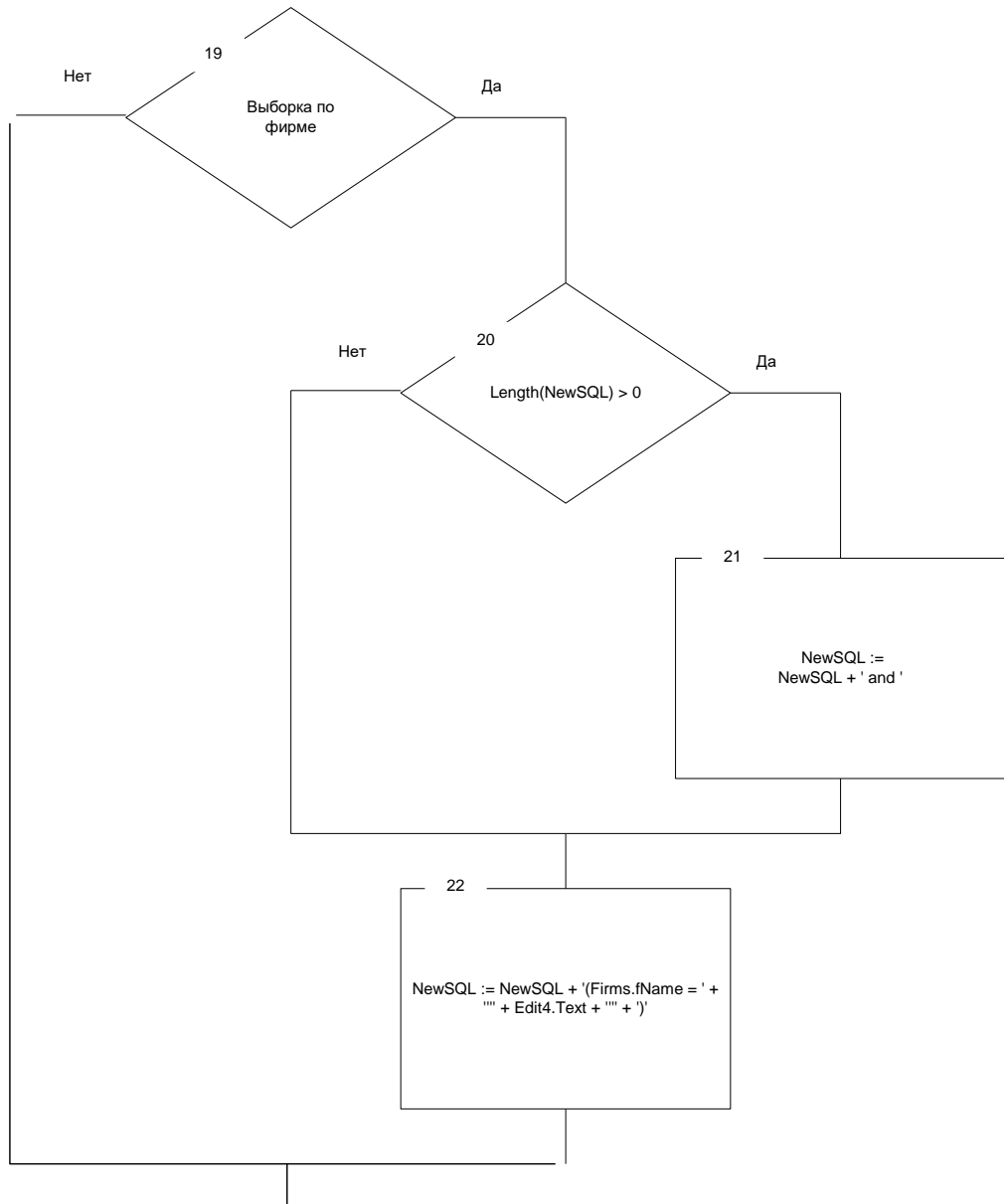


Рисунок 3.9 - Схема алгоритму формування вибірки за умовою

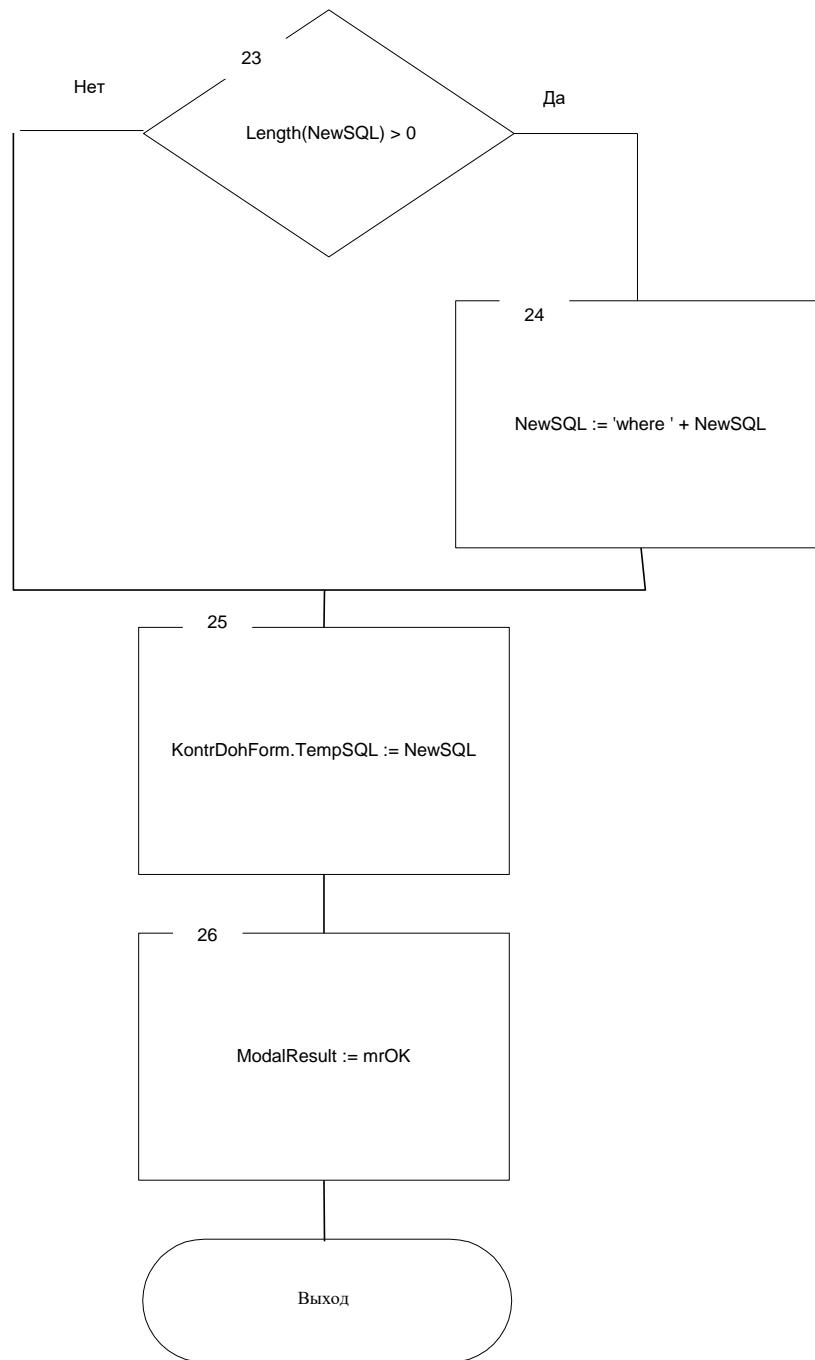


Рисунок 3.10 - Схема алгоритму формування вибірки за умовою

У розділі розроблено необхідне програмне забезпечення. Зроблено вибір інструментального засобу програмування. Розглянута швидкість роботи компілятора і швидкодія програм, та самі підпрограми.

4 НОРМИ І ВИМОГИ ОХОРОНИ ПРАЦІ НА РОБОЧОМУ МІСЦІ

4.1 Загальні положення

Зростання ефективності праці та збереження здоров'я працівників під час професійної діяльності залишаються актуальною проблемою і одним із шляхів її вирішення є підвищення, як соціальної відповідальності держави, так і особистої відповідальності працівників за безпеку праці.

Основним завданням розділу "Охорона праці" [1] є розробка технічних, санітарно-гігієнічних і організаційних заходів, спрямованих на усунення причин виробничого травматизму, професійної захворюваності, підвищення продуктивності праці.

До роботи з персональними ЕОМ і зовнішніми пристроями ПК допускаються особи, які пройшли спеціальне навчання, медичний огляд, перевірку знань вимог даної інструкції і питань пожежної безпеки, а також інструктажі в установленому на підприємстві порядку.

Заборонено доступ до роботи осіб, які не пройшли навчання, інструктаж і перевірку знань з охорони праці, медичний огляд (ст. 17, ст. 18 Закону України «Про охорону праці»)

Оператори і користувачі ПК зобов'язані:

- знати пристрої і правила безпечної експлуатації комп'ютерної техніки;
- проходити в установленому порядку періодичні медичні огляди;
- виконувати роботу відповідно до завдання безпосереднього керівника;
- не допускати в робочу зону сторонніх осіб;
- при роботі з Інтернетом користуватися тільки ресурсами мережі, необхідними для виконання поставленого виробничого завдання;
- відкривати вкладення в листи від невідомих користувачів
- не робити завантаження файлів, не перевірявши їх на наявність вірусів;
- знати і виконувати вимоги Закону України "про авторські та суміжні права";

- знати місця розташування первинних засобів пожежогасіння та вміти ними користуватися;
- виконувати правила внутрішнього трудового розпорядку, які регулюються Законом про охорону праці;
- дотримуватися правил особистої гігієни.

4.2 Аналіз стану умов праці

Для створення системи автоматизації збору інформації достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

Оформлення дипломного проекту з розробки системи автоматизації збору інформації за фізичним навантаженням відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово-скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявні психофізіологічні небезпечні та шкідливі фактори:

- фізичного перевантаження:

- 1) статичного;
 - 2) динамічного;
- нервово-психічного перевантаження:
- 1) розумового перенапруження;
 - 2) монотонності праці;
 - 3) перенапруження аналізаторів;
 - 4) емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи [5]. Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви тривалістю 15 хвилин через кожен годину роботи.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в таблиці 4.1.

Найменування	Значення
Довжина, м	6
Ширина, м	3
Висота, м	2,5
Площа, м ²	18
Об'єм, м ³	45

Таблиця 4.1 – Розміри приміщення

Згідно з [5] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основним вимогам до організації робочого місця за [19] і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 -800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 -500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 -800

Приміщення кабінету має об'єм 45 м³, площу – 18 м².

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість

— близько 50%. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.

За ступенем пожежної безпеки приміщення належить до категорії В.

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

Основними характеристиками персонального комп'ютера є наступні:

- робоча напруга $U = + 220 + -5\%$;
- робочий струм $I = 2A$;
- споживана потужність $P = 350 \text{ Вт}$.

Роботу користувача розробленої підсистеми слід віднести до категорії Іа (легкі фізичні роботи). До даної категорії відносяться всі види діяльності, які виконуються сидячи, з періодом ходінням, і не потребують фізичного напруження [23].

Згідно з при експлуатації даного програмного продукту існують такі небезпечні і шкідливі виробничі фактори:

а) фізичні:

- 1) підвищений рівень напруги електричної мережі;
- 2) підвищена або знижена рухомість та вологість повітря;
- 3) підвищений рівень статичної електрики;
- 4) відсутність або нестача природного світла;
- 5) підвищений рівень електромагнітного випромінювання;

б) психофізіологічні:

в) фізичні перевантаження:

- статичні;
- динамічні;

г) нервово-психічні перевантаження:

- 1) розумове перенапруження;
- 2) монотонність праці;
- 3) емоційні перевантаження.

4.4 Заходи з охорони праці

4.4.1 Організація робочого місця з ПК

Площа на одне робоче місце з ВДТ або ПК не менше 6,0 кв. м, об'єм - не менше 20,0 куб. м. Приміщення має природне і штучне освітлення. При розміщенні робочих місць включена можливість прямих відблисків від джерел природного та штучного освітлення. Робочі місця розташовувані так, щоб природне світло падало з лівого боку.

Екран відео монітора розташований на оптимальній відстані від очей користувача, близько 600 мм. Клавіатура розташована на поверхні столу на відстані 100-300 мм від краю, зверненого до користувача.

Забороняється палити в приміщенні, в якому розташовані ПК і її зовнішні пристрої.

4.4.2 Електробезпека

Основним небезпечним фактором при роботі з ЕОМ є небезпека ураження людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані виявити наявності електричної напруги на обладнанні[19].

Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання струму в тілі;
- типу і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Приміщення для ЕОМ відноситься до приміщень без підвищеної небезпеки, тобто в приміщення, в яких відсутні умови, що створюють підвищену або особливу небезпеку.

Електробезпека забезпечується:

- відповідною конструкцією електроустановок;
- застосуванням технічних способів і засобів захисту;
- організаційними і технічними заходами.

Основними технічними способами і засобами захисту від ураження електричним струмом, що використовуються окремо або в поєднанні один з одним, є:

- захисне заземлення;
- занулення;
- мала напруга;
- захисне відключення;
- ізоляція струмоведучих частин;
- попереджувальна сигналізація, блокування, знаки безпеки;

4.4.3 Розрахунок захисного заземлення

Основними технічними способами і засобами захисту від ураження електричним струмом, що передбачаються в даному дипломному проєкті, є:

- захисне заземлення,
- занулення,
- захисне відключення,
- ізоляція струмоведучих частин.

Завдання захисного заземлення - усунення небезпеки ураження струмом у випадку дотику до корпусу та інших струмоведучих металевих частин електроустановок, які опинилися під напругою.

Розрахунок заземлюючого контуру виконується виходячи з умови (4.1).

$$R_{зв} = \frac{R_3 \cdot R_{л}}{R_{л} \cdot n \cdot \eta_3 + R_3 \cdot \eta_{л}} \leq 4 \text{ Ом} \quad (4.1)$$

де R_3 - опір заземлювача (стержня, труби, куточка і т.д.), Ом;

$R_{л}$ - Опір лінії, що з'єднує заземлювачі, Ом;

n - кількість заземлювачів;

η_3 і $\eta_{л}$ - Коефіцієнти екранування відповідно заземлювача і з'єднує смуги ($\eta_3 = 0,2 \div 0,9$; $\eta_{л} = 0,1 \div 0,7$).

Опір заземлювача розраховується за формулою 4.2

$$R_3 = \frac{\rho}{2 \cdot \pi \cdot l} \left(\ln \frac{2 \cdot l}{d} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right), \quad (4.2)$$

де ρ - питомий опір ґрунту (взяти з довідкової літератури);

l - довжина заземлювача (для труб 2-3 м, для стрижнів до 10 м), м;

d - діаметр заземлювача (для стрижнів 0,01-0,03 м, для труб 0,03-0,05 м);

t - відстань від середини забитого в ґрунт заземлювача до рівня землі, м.

Розрахуємо опір заземлювача (4.3).

$$R_3 = \frac{60}{2 \cdot \pi \cdot 3} \left(\ln \frac{2 \cdot 3}{0,03} + \frac{1}{2} \cdot \ln \frac{4 \cdot 1 + 3}{4 \cdot 1 - 3} \right) = 19,96 \quad (4.3)$$

Опір лінії, що з'єднує заземлювачі розраховується за формулою 4.4

$$R_{л} = \frac{\rho}{2 \cdot \pi \cdot l} \cdot \ln \frac{2 \cdot L^2}{b \cdot t}, \quad (4.4)$$

де L - довжина лінії, що з'єднує заземлювачі (при контурному заземленні вона приблизно дорівнює периметру виробничої будівлі), м;

b - ширина смуги (0,03 - при прокладанні всередині будівлі і 0,05 - при прокладанні поза будівлею), м;

t - глибина заземлення від рівня землі (0,5 м.).

Розрахуємо опір лінії, що з'єднує заземлювачі

$$R_{\pi} = \frac{60}{2 \cdot \pi \cdot 3} \cdot \ln \frac{2 \cdot 50^2}{0.03 \cdot 0.5} = 14.37 \quad (4.5)$$

Необхідна кількість заземлювачів, розраховується за формулою 4.6

$$n = \frac{2 \cdot R_3}{4 \cdot \eta_3} \quad (4.6)$$

де 4 - допустимий загальний опір;

2 - коефіцієнт сезонності.

Розрахуємо необхідну кількість заземлювачів,

$$n = \frac{2 \cdot 19.9}{4 \cdot 0.5} = 19.9 \approx 20 \quad (4.7)$$

Округлимо результат в більшу сторону і отримуємо необхідну кількість заземлювачів - 20. Маючи всі необхідні дані розрахуємо опір заземлюючого контуру.

$$R_{зз} = \frac{19.96 \cdot 14.37}{14.37 \cdot 20 \cdot 0.5 + 19.96 \cdot 0.4} = 1.89 \leq 4 \text{ Ом} \quad (4.8)$$

Опір заземлюючого контуру 1,89 Ом, що відповідає умові $R_{зз} < 4 \text{ Ом}$.

4.5 Заходи, що забезпечують виробничу санітарію та гігієну праці

4.5.1 Мікроклімат

Трудова діяльність людини завжди протікає в певних метеорологічних умовах, які визначаються поєднанням температури повітря, швидкості його руху і відносної вологості, тиском і тепловим випромінюванням від нагрітих поверхонь. Оскільки експлуатація проектного програмного засобу відбувається в приміщенні, то ці показники в сукупності (за винятком тиску) називаються мікрокліматом виробничого приміщення. На даний час основним нормативним документом, щодо нормалізації мікроклімату є [25].

Важкість праці характеризує сукупну дію всіх елементів, складових умови праці, на працездатність людини, його здоров'я, життєдіяльність і відновлення робочої сили. У такому представленні поняття тяжкості праці однаково застосовні як до розумової, так і до фізичної праці. Згідно [25] тяжкість роботи персоналу, який обслуговує ЕОМ, відноситься до легкої категорії 1б (роботи, виконувані сидячи, не вимагаючи систематичного фізичного напруження і перенесення важких предметів). Загальні санітарно-гігієнічні вимоги до повітря робочої зони. Оптимальні норми мікроклімату в робочій зоні, що забезпечуються для робіт легкої категорії 1б приведені в таблиці 4.3.

Таблиця 4.3 - Оптимальні норми мікроклімату

Період року	Температура, °С	Відносна вологість, %	Швидкість вітру, м / с
Холодний	21 - 23	60 - 40	0,1
Теплий	22 - 24	60 - 40	0,2

4.5.2 Освітлення

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

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

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла у світильниках загального освітлення, оскільки люмінесцентні лампи мають високу

потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний склад випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IV в розряд точності (середня точність). При цьому нормована освітленість на робочому місці (E_n) дорівнює 200 лк. Джерелом природного освітлення є сонячне світло. У приміщенні, де розташовані ЕОМ передбачається природне освітлення, рівень якого відповідає стандартам [26].

Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовільний і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для будівель виробництв світловий коефіцієнт приймається в межах 1/6 – 1/10 (4.9).

$$\sqrt{a^2 + b^2} \cdot S_b = (1/8 \div 1/10) \cdot S_n \quad (4.9)$$

де S_b - площа віконних прорізів, m^2 ;

S_n - площа підлоги, m^2 .

$$S_n = a \cdot b = 6 \cdot 3 = 18 \text{ м}^2$$

$$S_{\text{вік}} = 1/8 \cdot 18 = 2,25 \text{ м}^2$$

Приймаємо 2 вікна площею $S = 1,13 \text{ м}^2$ кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.10):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.10)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 18m^2$;

Z – поправочний коефіцієнт світильника ($Z=1,15$ для ламп розжарювання та ДРЛ; $Z=1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400 лм (для ЛБ – 80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 18 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 1.$$

Приймаємо освітлювальну установку, яка складається з одного світильника, який складається з 2-х люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.6 Рекомендації щодо пожежної безпеки

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

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

– поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання $420^\circ C$;

– пластикат кабельний - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;

– деревина - будівельний і оздоблювальний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Згідно [20] таке приміщення належить до категорії "В" (пожежонебезпечної).

Причинами можливого загоряння і пожежі можуть бути:

- несправність електроустановки;
- конструктивні недоліки обладнання;
- коротке замикання в електричних мережах;
- запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згоряння, що виділяються під час пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор та ін.

При горінні пластмас, крім звичних продуктів згоряння, виділяються різні продукти термічного розкладання: хлорангідридні кислоти; формальдегіди; хлористий водень; фосген; синильна кислота; аміак; фенол; ацетон; стирол [27].

Для захисту персоналу від впливу небезпечних і шкідливих факторів пожежі проектом передбачається застосування промислового протигазу фільтруючого з коробкою марки В (жовтий).

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється розвиненою системою електроживлення ЕОМ. Небезпека загорання в ЕОМ пов'язана з великою кількістю щільно розташованих на платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів.

Пожежна безпека при застосуванні ЕОМ забезпечується:

- системою запобігання пожежі;
- системою протипожежного захисту;
- організаційно-технічними заходами.

Запобігти утворенню горючого середовища не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворенню (або внесення) в горюче середовище джерел запалювання, таких як:

- застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонами відповідно до ПУЕ;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалювання;
- виключення можливості появи іскрового розряду в займистою середовищі з енергією, яка дорівнює і вище мінімальної енергії запалювання.

В четвертому розділі була проведена робота, в результаті зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також наведена інформація щодо пожежної та електробезпеки. Описані розміри приміщення та значення температури, вологості і рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ВИСНОВКИ

Метою даної роботи була розробка, впровадження та застосування підсистеми «Контроль доходів». Тема актуальна і має практичне значення для організації спілкування між відділом субсидій та податковою адміністрацією, за допомогою сучасних інформаційних технологій.

У проекті поставлені і вирішені такі основні завдання: проаналізовано існуючі програмні засоби аналогічного призначення, зроблені відповідні висновки про областях їх застосування, а також про необхідність і доцільність розробки нового програмного засобу. Складено технічне завдання на розробку програмного забезпечення підсистеми "Контроль доходів", що включає в себе основні технічні вимоги до розробляється програмного забезпечення.

У першому розділі виконаний аналіз аналогічних засобів та проаналізовано їх переваги та недоліки. А саме: важливість забезпечення високої продуктивності функціонування інформаційної системи. Розглянуті особливості роботи відділу субсидій. Виявлені переваги комп'ютерних систем на паперовим методом збереження даних. А також описані функції та структура СУБД, мова БД та інше.

У другому розділі проведений аналіз технічного завдання та постановка задачі проектування. Розглянута архітектура БД, структура моделі підсистем. Визначено функціональну структуру системи та структури таблиць підсистем.

У третьому розділі було розроблено необхідне програмне забезпечення. Зроблено вибір інструментального засобу програмування. Розглянута швидкість роботи компілятора і швидкодія програм, та самі підпрограми.

У розділі норми і вимоги охорони праці на робочому місці розглянуті загальні положення; зроблений аналіз стану приміщення та аналіз шкідливих факторів при роботі з персональним комп'ютером. Описані основні заходи з

охорони праці та зроблені відповідні розрахунки. Дані рекомендації щодо пожежної безпеки.

ПЕРЕЛІК ПОСИЛАНЬ

1. НПАОП 40.1-1.01-97 Правила безпечної експлуатації електроустановок.
<https://zakon.rada.gov.ua/laws/show/z0011-98>
2. Розробка і оформлення конструкторської документації радіоелектронної апаратури: Довідник / Е.Т. Романичева, А.К. Іванова, А.С. Куликов та ін .; Під ред. Е.Т. Романичева. - 2-е изд., Перераб. і доп. - М .: Радио и связь, 1989. - 448 с .: іл.
3. В.В.Шураков. "Автоматизоване робоче місце для статичної обробки даних", 1990р.
4. Посібник з надання житлових субсидій. Міністерство праці та соціальної політики України, Київ - 2000.
5. Хомоненко А.Д., Циганков В.М., Мальцев М.Г. Бази даних: підручник для вищих навчальних закладів - СПб .: Корона, 2000., - 416с.
6. І.Л.Кантарь. "Автоматизовані робочі місця управлінського апарату", 1990р.
7. Д. А. Недзельський «Методичні вказівки щодо виконання розділу дипломного проекту« Надійність »»: - Сєверодонецьк: Изд-во СТІ Восточноукр. Нац. Ун-ту, 2003. - 44с.
8. Методичні вказівки з економічного обґрунтування дипломних проектів для студентів спеціальностей 22.01 і 22.05. Северод про нецке 1996 р
9. Техніко-економічне обґрунтування дипломних проектів: Учеб ве посібник для втузів / Л. А. Астреїна, В. В. Балдесов, В. К. Беклешов і ін .; Під ред. В. К. Беклешова. - М .: Вища. шк., 1991.-176 с .: іл.
- 10.Методичні вказівки по техніко-економічного обґрунтування дипломних проектів для студентів спеціальностей 501,502, 001, 003. Сєверодонецьк 2002р.
- 11.Delphi 5: навчальний курс - СПб .: Питер, 2000 - 640 с .: іл.

12. Коутс Р., Влеймінк І. Інтерфейс "людина - комп'ютер". - М.: Світ, 1990. - 501 с.
13. Лиуду К.А. Надійність, контроль і діагностика обчислювальних машин і систем. - М.: Вища. Шк., 1989, 216 с.
14. Пожаровзривопасність речовин і матеріалів і кошти їх тушення: Довідник під ред. Баратова А. Н. - М: Хімія, 1990. - Ср.
15. НАПБ А.01.001-2014 Правила пожежної безпеки України.
<https://zakon.rada.gov.ua/laws/show/z0252-15#n14>
16. ДСТУ 2273-93. ССБП. Пожежна техніка.
https://dnaop.com/html/41014/doc-ДСТУ_2273-93
17. ДБН В.2.5-28:2018 Природне і штучне освітлення
https://dbn.co.ua/load/normativy/dbn/dbn_v_2_5_28/1-1-0-1188
18. ДСТУ-НБА 3.2-1:2007. Настанова щодо визначення небезпечних і шкідливих факторів та захисту від їх впливу.
https://dnaop.com/html/29999/doc-ДСТУ-Н_Б_А_3.2-1_2007
19. НАПБ Б.03.002-2007. Норми визначення категорій приміщень.
https://dnaop.com/html/32980/doc-НАПБ_Б.03.002-2007
20. НПАОП 0.00-4.15-98. Положення про розробку інструкцій з охорони праці
https://dnaop.com/html/64/doc-НПАОП_0.00-4.15-98
21. НПАОП 0.00-4.12.05. Положення про порядок проведення навчання і перевірки знань з питань охорони праці
https://dnaop.com/html/32368/doc-НПАОП_0.00-4.12.05
22. ДСТУ 7237:2011. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту
http://online.budstandart.com/ru/catalog/doc-page?id_doc=30045
23. Санітарні норми мікроклімату ДСН 3.3.6.042-99
<https://zakon.rada.gov.ua/rada/show/va042282-99>
24. ДБН В.2.5-67:2013. Опалення, вентиляція та кондиціонування.
https://dnaop.com/html/32609/doc-ДБН_В.2.5-67_2013

25. ДСанПіН 3.3.2-007-98. Державні санітарні правила та норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

https://dnaop.com/html/40949/doc-ДСанПіН_3.3.2-007-98

ДОДАТОК А Лістинг програми АРМ "Субсидії"

```

1  program Subsidii;
2
3  uses
4    Forms,
5    frmMain in 'frmMain.pas' {MainForm},
6    frmKontrDoh in 'frmKontrDoh.pas' {KontrDohForm},
7    frmRequests in 'frmRequests.pas' {RequestsForm},
8    frmIncomes in 'frmIncomes.pas' {IncomesForm},
9    repRequests in 'repRequests.pas' {ReqRepForm},
10   frmSelReq in 'frmSelReq.pas' {SelReqForm},
11   frmPersInfo in 'frmPersInfo.pas' {PersInfoForm},
12   repIncomes in 'repIncomes.pas' {IncRepForm},
13   repDifferences in 'repDifferences.pas' {DifRepForm},
14   frmQueries in 'frmQueries.pas' {QueriesForm},
15   frmSelQuery in 'frmSelQuery.pas' {SelQueryForm},
16   frmQueryInfo in 'frmQueryInfo.pas' {QueryInfoForm},
17   repQueries in 'repQueries.pas' {QueriesRepForm},
18   frmIncInfo in 'frmIncInfo.pas' {IncInfoForm},
19   frmPerIncInfo in 'frmPerIncInfo.pas' {PerIncInfoForm},
20   frmPayments in 'frmPayments.pas' {PaymentsForm},
21   frmServiceInfo in 'frmServiceInfo.pas' {ServiceInfoForm},
22   frmFirms in 'frmFirms.pas' {FirmsForm},
23   frmOrganizations in 'frmOrganizations.pas' {OrganizationsForm},
24   repSubsidy in 'repSubsidy.pas' {SubsRepForm};
25
26  {$R *.res}
27
28  begin
29    Application.Initialize;
30    Application.Title := 'АРМ СУБСИДИИ';
31    Application.CreateForm(TMainForm, MainForm);
32    Application.Run;
33  end.
34
35
36
37  unit frmMain;
38
39  interface
40
41  uses
42    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
43    Forms,
44    Dialogs, StdCtrls, ExtCtrls;
45
46  type
47    TMainForm = class(TForm)
48      Panel1: TPanel;
49      Button1: TButton;
50      Button2: TButton;
51      Button3: TButton;
52      Button4: TButton;
53      Button5: TButton;
54      Button6: TButton;
55      btnKontrDoh: TButton;
56      btnClose: TButton;
57      procedure btnCloseClick(Sender: TObject);
58      procedure FormCloseQuery(Sender: TObject; var CanClose:

```

```

59 Boolean);
60   procedure Button1Click(Sender: TObject);
61   procedure btnKontrDohClick(Sender: TObject);
62 private
63   { Private declarations }
64 public
65   { Public declarations }
66 end;
67
68 var
69   MainForm: TMainForm;
70
71 implementation
72
73 Uses
74   frmKontrDoh;
75
76 {$R *.dfm}
77
78 procedure TMainForm.btnCloseClick(Sender: TObject); //Выход
79 begin
80   Close;
81 end;
82
83 procedure TMainForm.FormCloseQuery(Sender: TObject; var
84   CanClose: Boolean);
85 begin
86   if MessageDlg('Вы действительно хотите выйти из программы?',
87     mtConfirmation, [mbYes, mbNo], 0) = mrNo then
88     CanClose := false;
89 end;
90
91 procedure TMainForm.Button1Click(Sender: TObject);
92 begin
93   MessageDlg('Этот раздел в данном проекте не реализован',
94     mtInformation, [mbOK], 0);
95 end;
96
97 procedure TMainForm.btnKontrDohClick(Sender: TObject);
98 begin
99   KontrDohForm := TKontrDohForm.Create(Self);
100   KontrDohForm.ShowModal;
101   MainForm.Show;
102 end;
103
104 end.
105
106
107
108 unit frmKontrDoh;
109
110 interface
111
112 uses
113   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
114   Forms,
115   Dialogs, ExtCtrls, Menus, frmMain;
116
117 type
118   TKontrDohForm = class(TForm)
119     MainMenu1: TMainMenu;
120     N1: TMenuItem;

```

```

121   N2: TMenuItem;
122   N3: TMenuItem;
123   N4: TMenuItem;
124   Panel1: TPanel;
125   N5: TMenuItem;
126   N6: TMenuItem;
127   N7: TMenuItem;
128   N9: TMenuItem;
129   N12: TMenuItem;
130   N13: TMenuItem;
131   N8: TMenuItem;
132   N10: TMenuItem;
133   N11: TMenuItem;
134   N14: TMenuItem;
135   N15: TMenuItem;
136   N16: TMenuItem;
137   N17: TMenuItem;
138   procedure FormShow(Sender: TObject);
139   procedure N13Click(Sender: TObject);
140   procedure N6Click(Sender: TObject);
141   procedure N5Click(Sender: TObject);
142   procedure N7Click(Sender: TObject);
143   procedure N8Click(Sender: TObject);
144   procedure N10Click(Sender: TObject);
145   procedure N16Click(Sender: TObject);
146   procedure N17Click(Sender: TObject);
147   procedure N14Click(Sender: TObject);
148   procedure N15Click(Sender: TObject);
149   private
150     { Private declarations }
151   public
152     { Public declarations }
153     MainDir, TempSQL: string;
154   end;
155
156   var
157     KontrDohForm: TKontrDohForm;
158
159   implementation
160
161   Uses
162     frmRedTabl, frmRequests, frmSelReq, frmIncomes, repDifferences,
163     frmQueries,
164     frmSelQuery, frmFirms, frmOrganizations;
165
166   {$R *.dfm}
167
168   procedure TKontrDohForm.FormShow(Sender: TObject);
169   begin
170     WindowState := wsMaximized;
171     MainDir := GetCurrentDir;
172   end;
173
174   procedure TKontrDohForm.N13Click(Sender: TObject); //Редактор таблиц
175
176   begin
177     RedTablForm := TRedTablForm.Create(Self);
178     RedTablForm.ShowModal;
179     RedTablForm.Free;
180   end;
181
182   procedure TKontrDohForm.N6Click(Sender: TObject); //Вызов формы заявлений

```

```
183
184 begin
185   RequestsForm := TRequestsForm.Create(self);
186   RequestsForm.ShowModal;
187   RequestsForm.Free;
188 end;
189
190 procedure TKontrDohForm.N5Click(Sender: TObject); //Вызов формы условий отбора заявлений
191
192 begin
193   SelReqForm := TSelReqForm.Create(self);
194   if SelReqForm.ShowModal = mrOK then begin
195     RequestsForm := TRequestsForm.Create(self);
196     RequestsForm.RequestsQuery.Active := false;
197     RequestsForm.RequestsQuery.SQL.Add(TempSQL);
198     RequestsForm.RequestsQuery.Active := true;
199     RequestsForm.ShowModal;
200     RequestsForm.Free;
201   end
202   else
203     SelReqForm.Free;
204 end;
205
206 procedure TKontrDohForm.N7Click(Sender: TObject);
207 //Формирование отчета о доходах
208 begin
209   IncomesForm := TINcomesForm.Create(self);
210   IncomesForm.Inbox := false;
211   IncomesForm.ShowModal;
212   IncomesForm.Free;
213 end;
214
215 procedure TKontrDohForm.N8Click(Sender: TObject); //Ввод информации о доходах
216
217 begin
218   if MessageDlg('Получить данные из ГНИ?', mtConfirmation,
219 [mbYes, mbNo], 0) = mrYes then begin
220     MessageDlg('Данные по доходам из ГНИ получены.',
221 mtInformation, [mbOK], 0);
222
223     IncomesForm := TINcomesForm.Create(self);
224     IncomesForm.Caption := 'Данные по доходам из ГНИ';
225     IncomesForm.InBox := true;
226     IncomesForm.ShowModal;
227     IncomesForm.Free;
228   end;
229 end;
230
231 procedure TKontrDohForm.N10Click(Sender: TObject); //Печать реестра расхождений
232
233 begin
234   DifRepForm := TDifRepForm.Create(self);
235   DifRepForm.ShowModal;
236   DifRepForm.Free;
237 end;
238
239 procedure TKontrDohForm.N16Click(Sender: TObject); //Вызов формы условий отбора базы обмена с
ГНИ
240
241 begin
242   SelQueryForm := TSelQueryForm.Create(self);
243   if SelQueryForm.ShowModal = mrOK then begin
```

```

244   QueriesForm := TQueriesForm.Create(self);
245   QueriesForm.QueriesQuery.Active := false;
246   QueriesForm.QueriesQuery.SQL.Add(TempSQL);
247   QueriesForm.QueriesQuery.Active := true;
248   QueriesForm.ShowModal;
249   QueriesForm.Free;
250 end
251 else
252   SelQueryForm.Free;
253 end;
254
255 procedure TKontrDohForm.N17Click(Sender: TObject); //Просмотр базы обмена с ГНИ
256
257 begin
258   QueriesForm := TQueriesForm.Create(self);
259   QueriesForm.ShowModal;
260   QueriesForm.Free;
261 end;
262
263 procedure TKontrDohForm.N14Click(Sender: TObject);
264 //Справочник фирм
265 begin
266   FirmsForm := TFirmsForm.Create(self);
267   FirmsForm.ShowModal;
268   FirmsForm.Free;
269 end;
270
271 procedure TKontrDohForm.N15Click(Sender: TObject);
272 //Справочник организаций
273 begin
274   OrganizationsForm := TOrganizationsForm.Create(self);
275   OrganizationsForm.ShowModal;
276   OrganizationsForm.Free;
277 end;
278
279 end.
280
281
282
283 unit frmRequests;
284
285 interface
286
287 uses
288   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
289   Forms,
290   Dialogs, DB, DBTables, Grids, DBGrids, ExtCtrls, StdCtrls, DBCtrls;
291
292 type
293   TRequestsForm = class(TForm)
294     Panel1: TPanel;
295     Panel2: TPanel;
296     RequestsGrid: TDBGrid;
297     Panel3: TPanel;
298     RequestsSource: TDataSource;
299     RequestsQuery: TQuery;
300     btnClose: TButton;
301     btnReport: TButton;
302     btnToFile: TButton;
303     btnView: TButton;
304     SaveDialog1: TSaveDialog;
305     RequestsQueryN: TIntegerField;

```



```

306   RequestsQueryRDATEBEGIN: TDateField;
307   RequestsQueryRDATEEND: TDateField;
308   RequestsQueryRGKU: TFloatField;
309   RequestsQueryRSGTT: TFloatField;
310   RequestsQueryRGKUSUMM: TFloatField;
311   RequestsQueryRSGTTSUMM: TFloatField;
312   RequestsQueryPCODE: TStringField;
313   RequestsQueryPNAME: TStringField;
314   RequestsQueryPADDRESS: TStringField;
315   RequestsQueryFNAME: TStringField;
316   RequestsQueryRPERSON: TFloatField;
317   DBNavigator1: TDBNavigator;
318   procedure FormCreate(Sender: TObject);
319   procedure FormClose(Sender: TObject; var Action: TCloseAction);
320   procedure RequestsQueryNGetText(Sender: TField; var Text:
321   String;
322     DisplayText: Boolean);
323   procedure btnCloseClick(Sender: TObject);
324   procedure btnReportClick(Sender: TObject);
325   procedure btnToFileClick(Sender: TObject);
326   procedure btnViewClick(Sender: TObject);
327   procedure RequestsGridDbClick(Sender: TObject);
328   private
329     { Private declarations }
330   public
331     { Public declarations }
332   end;
333
334   var
335     RequestsForm: TRequestsForm;
336
337   implementation
338
339   Uses
340     frmKontrDoh, repRequests, frmPersInfo;
341
342   {$R *.dfm}
343
344   procedure TRequestsForm.FormCreate(Sender: TObject); //Вывод завлений
345
346   begin
347     RequestsQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
348
349     try
350       Screen.Cursor := crSQLWait;
351       RequestsQuery.Active := true;
352       Screen.Cursor := crDefault;
353
354       RequestsGrid.Columns.Items[0].Title.Caption := '№ п/п';
355       RequestsGrid.Columns.Items[1].Title.Caption := 'Дата начала';
356       RequestsGrid.Columns.Items[2].Title.Caption := 'Дата конца';
357       RequestsGrid.Columns.Items[3].Title.Caption := 'Идент. Код';
358       RequestsGrid.Columns.Items[4].Title.Caption := 'Ф.И.О.';
359       RequestsGrid.Columns.Items[5].Title.Caption := 'Адрес';
360       RequestsGrid.Columns.Items[6].Title.Caption := 'Предприятие';
361       RequestsGrid.Columns.Items[7].Title.Caption := 'ЖКУ';
362       RequestsGrid.Columns.Items[8].Title.Caption := 'СГТТ';
363       RequestsGrid.Columns.Items[9].Title.Caption := 'Сумма ЖКУ';
364       RequestsGrid.Columns.Items[10].Title.Caption := 'Сумма СГТТ';
365     finally
366     end;
367   end;

```

```

368
369 procedure TRequestsForm.FormClose(Sender: TObject;
370   var Action: TCloseAction);
371 begin
372   RequestsQuery.Active := false;
373 end;
374
375 procedure TRequestsForm.RequestsQueryNGetText(Sender: TField;
376   var Text: String; DisplayText: Boolean);
377 begin
378   Text := IntToStr(Sender.DataSet.RecNo);
379 end;
380
381 procedure TRequestsForm.btnCloseClick(Sender: TObject); //Выход
382 begin
383   Close;
384 end;
385
386 procedure TRequestsForm.btnReportClick(Sender: TObject);
387 //Печать отчета
388 begin
389   ReqRepForm := TReqRepForm.Create(self);
390   ReqRepForm.ShowModal;
391   ReqRepForm.Free;
392 end;
393
394 procedure TRequestsForm.btnToFileClick(Sender: TObject);
395 //Сохранение в файл
396 var
397   TTL: TextFile;
398   TTLFileName, TmpStr: string;
399   k: integer;
400   CurDate: TDateTime;
401
402 procedure Tab(N, L: integer; var S: string);
403 var
404   i: integer;
405 begin
406   for i := 1 to N - L do
407     S := S + ' ';
408 end;
409
410 begin
411   CurDate := Now;
412   TTLFileName := KontrDohForm.MainDir + '\temp\' +
413   FormatDateTime('ddMMYY_hhmm', CurDate) + '.ttl';
414
415   AssignFile(TTL, TTLFileName);
416   Rewrite(TTL);
417   TmpStr := ' Список завлений (Файл создан ' + FormatDateTime
418   ('hh:mm:ss dd.MM.yyyy', CurDate) + ')';
419   Writeln(TTL, TmpStr);
420   Writeln(TTL, '');
421   TmpStr := ' № п/п | Дата начала | Дата конца | Идент. код |
422   Ф.И.О.      |';
423   TmpStr := TmpStr + '                Адрес                |
424   Предприятие |';
425   TmpStr := TmpStr + 'КУ|СГТТ|Сумма ЖКУ |Сумма СГТТ|';
426   Writeln(TTL, TmpStr);
427
428   RequestsQuery.First;
429   k := 0;

```

```

430 while not RequestsQuery.Eof do begin
431   Inc(k);
432
433   TmpStr := "";
434   Tab(4, Length(IntToStr(k)), TmpStr);
435   TmpStr := TmpStr + IntToStr(k) + ' | ';
436   TmpStr := TmpStr + RequestsQueryRDATEBEGIN.AsString;
437   Tab(10, Length(RequestsQueryRDATEBEGIN.AsString), TmpStr);
438   TmpStr := TmpStr + ' | ';
439   TmpStr := TmpStr + RequestsQueryRDATEEND.AsString;
440   Tab(10, Length(RequestsQueryRDATEEND.AsString), TmpStr);
441   TmpStr := TmpStr + ' | ';
442   TmpStr := TmpStr + RequestsQueryPCODE.AsString;
443   Tab(10, Length(RequestsQueryPCODE.AsString), TmpStr);
444   TmpStr := TmpStr + ' | ';
445   TmpStr := TmpStr + RequestsQueryPNAME.AsString;
446   Tab(25, Length(RequestsQueryPNAME.AsString), TmpStr);
447   TmpStr := TmpStr + ' | ';
448   TmpStr := TmpStr + RequestsQueryPADDRESS.AsString;
449   Tab(50, Length(RequestsQueryPADDRESS.AsString), TmpStr);
450   TmpStr := TmpStr + ' | ';
451   TmpStr := TmpStr + RequestsQueryFNAME.AsString;
452   Tab(20, Length(RequestsQueryFNAME.AsString), TmpStr);
453   TmpStr := TmpStr + ' | ';
454   TmpStr := TmpStr + RequestsQueryRGKU.AsString;
455   Tab(1, Length(RequestsQueryRGKU.AsString), TmpStr);
456   TmpStr := TmpStr + ' | ';
457   TmpStr := TmpStr + RequestsQueryRSGTT.AsString;
458   Tab(1, Length(RequestsQueryRSGTT.AsString), TmpStr);
459   TmpStr := TmpStr + ' | ';
460   Tab(6, Length(RequestsQueryRGKUSUMM.AsString), TmpStr);
461   TmpStr := TmpStr + RequestsQueryRGKUSUMM.AsString + ' | ';
462   Tab(6, Length(RequestsQueryRSGTTSUMM.AsString), TmpStr);
463   TmpStr := TmpStr + RequestsQueryRSGTTSUMM.AsString;
464
465   WriteLn(TTL, TmpStr);
466   RequestsQuery.Next;
467 end;
468 CloseFile(TTL);
469
470 MessageDlg('Создан TTL-файл: "' + 'R' + TTLFileName + "'",
471 mtInformation, [mbOK], 0);
472 end;
473
474 procedure TRequestsForm.btnViewClick(Sender: TObject);
475 //Информация по заявлению
476 begin
477   PersInfoForm := TPersInfoForm.Create(self);
478   PersInfoForm.PersonId := RequestsQueryRPERSON.AsInteger;
479   PersInfoForm.ShowModal;
480   PersInfoForm.Free;
481 end;
482
483 procedure TRequestsForm.RequestsGridDbClick(Sender: TObject);
484 //Информация по заявлению
485 begin
486   btnViewClick(Sender);
487 end;
488
489 end.
490
491

```

```

492
493 unit frmIncomes;
494
495 interface
496
497 uses
498   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
499   Forms,
500   Dialogs, ExtCtrls, StdCtrls, Grids, DBGrids, DB, DBTables, DBCtrls;
501
502 type
503   TIncomesForm = class(TForm)
504     Panel1: TPanel;
505     Panel2: TPanel;
506     Panel3: TPanel;
507     btnReport: TButton;
508     btnToFile: TButton;
509     btnClose: TButton;
510     IncomesGrid: TDBGrid;
511     IncomesQuery: TQuery;
512     IncomesSource: TDataSource;
513     IncomesQueryIDATEBEGIN: TDateField;
514     IncomesQueryIDATEEND: TDateField;
515     IncomesQueryITYPE: TStringField;
516     IncomesQueryISUMM: TFloatField;
517     IncomesQueryPCODE: TStringField;
518     IncomesQueryPNAME: TStringField;
519     IncomesQueryPADDRESS: TStringField;
520     IncomesQueryPEGRPOU: TStringField;
521     IncomesQueryIID: TFloatField;
522     IncomesQueryN: TIntegerField;
523     DBNavigator1: TDBNavigator;
524     btnView: TButton;
525     IncomesQueryIPERSON: TFloatField;
526     procedure SaveTTL;
527     procedure FormShow(Sender: TObject);
528     procedure FormClose(Sender: TObject; var Action: TCloseAction);
529     procedure IncomesQueryNGetText(Sender: TField; var Text:
530     String;
531     DisplayText: Boolean);
532     procedure btnReportClick(Sender: TObject);
533     procedure btnToFileClick(Sender: TObject);
534     procedure btnCloseClick(Sender: TObject);
535     procedure btnViewClick(Sender: TObject);
536     procedure IncomesGridDbClick(Sender: TObject);
537     private
538     { Private declarations }
539     ToFile: boolean;
540     public
541     { Public declarations }
542     InBox: boolean;
543     end;
544
545 var
546   IncomesForm: TIncomesForm;
547
548 implementation
549
550 Uses
551   frmKontrDoh, repIncomes, frmIncInfo;
552
553 {$R *.dfm}

```

```

554
555 procedure TIncomesForm.SaveTTL;
556 var
557   TTL: TextFile;
558   TTLFileName, TmpStr: string;
559   k: integer;
560   CurDate: TDateTime;
561
562 procedure Tab(N, L: integer; var S: string);
563 var
564   i: integer;
565 begin
566   for i := 1 to N - L do
567     S := S + ' ';
568 end;
569
570 begin
571   CurDate := Now;
572   TTLFileName := KontrDohForm.MainDir + '\temp\' + T +
573   FormatDateTime('ddMMYY_hhmm', CurDate) + '.ttl';
574
575   AssignFile(TTL, TTLFileName);
576   Rewrite(TTL);
577   TmpStr := ' Список заявлений. (Файл создан ' + FormatDateTime
578 ('hh:mm:ss dd.MM.yyyy', CurDate) + ')';
579   Writeln(TTL, TmpStr);
580   Writeln(TTL, '');
581   TmpStr := ' №п/п |№ дела|Дата Начала | Дата Конца | Идент.
582 код |      Ф.И.О.      |';
583   TmpStr := TmpStr + '                Адрес                | Код
584 ЕГРПОУ |';
585   TmpStr := TmpStr + ' | Сумма | Вид дохода |';
586   Writeln(TTL, TmpStr);
587
588   IncomesQuery.First;
589   k := 0;
590   while not IncomesQuery.Eof do begin
591     Inc(k);
592
593     TmpStr := '';
594     Tab(4, Length(IntToStr(k)), TmpStr);
595     TmpStr := TmpStr + IntToStr(k) + ' | ';
596
597     Tab(4, Length(IncomesQuery.IID.AsString), TmpStr);
598     TmpStr := TmpStr + IntToStr(k) + ' | ';
599
600     TmpStr := TmpStr + IncomesQuery.IDATEBEGIN.AsString;
601     Tab(10, Length(IncomesQuery.IDATEBEGIN.AsString), TmpStr);
602     TmpStr := TmpStr + ' | ';
603
604     TmpStr := TmpStr + IncomesQuery.IDATEEND.AsString;
605     Tab(10, Length(IncomesQuery.IDATEEND.AsString), TmpStr);
606     TmpStr := TmpStr + ' | ';
607
608     TmpStr := TmpStr + IncomesQuery.PCODE.AsString;
609     Tab(10, Length(IncomesQuery.PCODE.AsString), TmpStr);
610     TmpStr := TmpStr + ' | ';
611
612     TmpStr := TmpStr + IncomesQuery.PNAME.AsString;
613     Tab(25, Length(IncomesQuery.PNAME.AsString), TmpStr);
614     TmpStr := TmpStr + ' | ';
615

```

```

616   TmpStr := TmpStr + IncomesQueryPADDRESS.AsString;
617   Tab(50, Length(IncomesQueryPADDRESS.AsString), TmpStr);
618   TmpStr := TmpStr + ' | ';
619
620   TmpStr := TmpStr + IncomesQueryPEGRPOU.AsString;
621   Tab(10, Length(IncomesQueryPEGRPOU.AsString), TmpStr);
622   TmpStr := TmpStr + ' | ';
623
624   TmpStr := TmpStr + IncomesQueryISUMM.AsString;
625   Tab(6, Length(IncomesQueryISUMM.AsString), TmpStr);
626   TmpStr := TmpStr + ' | ';
627
628   Tab(10, Length(IncomesQueryITYPE.AsString), TmpStr);
629   TmpStr := TmpStr + IncomesQueryITYPE.AsString;
630
631   WriteLn(TTL, TmpStr);
632   IncomesQuery.Next;
633 end;
634 CloseFile(TTL);
635
636   MessageDlg('Создан TTL-файл: "' + TTLFileName + "',
637 mtInformation, [mbOK], 0);
638 end;
639
640 procedure TIncomesForm.FormShow(Sender: TObject);
641 begin
642   ToFile := false;
643   if not InBox then
644     if MessageDlg('Создавать файл для ГНИ?', mtConfirmation,
645 [mbYes, mbNo], 0) = mrYes then
646       ToFile := true;
647
648   if not InBox then
649     IncomesQuery.DatabaseName := KontrDohForm.MainDir + '\DBF'
650   else
651     IncomesQuery.DatabaseName := KontrDohForm.MainDir +
652 '\InBox';
653
654   try
655     Screen.Cursor := crSQLWait;
656     IncomesQuery.Active := true;
657     Screen.Cursor := crDefault;
658
659     IncomesGrid.Columns.Items[0].Title.Caption := '№ п/п';
660     IncomesGrid.Columns.Items[1].Title.Caption := 'Номер дела';
661     IncomesGrid.Columns.Items[2].Title.Caption := 'Дата начала';
662     IncomesGrid.Columns.Items[3].Title.Caption := 'Дата конца';
663     IncomesGrid.Columns.Items[4].Title.Caption := 'Идент. Код';
664     IncomesGrid.Columns.Items[5].Title.Caption := 'Ф.И.О.';
665     IncomesGrid.Columns.Items[6].Title.Caption := 'Адрес';
666     IncomesGrid.Columns.Items[7].Title.Caption := 'Код ЕГРПОУ';
667     IncomesGrid.Columns.Items[8].Title.Caption := 'Сумма дохода';
668     IncomesGrid.Columns.Items[9].Title.Caption := 'Вид дохода';
669
670     if ToFile then
671       SaveTTL;
672     finally
673     end;
674 end;
675
676 procedure TIncomesForm.FormClose(Sender: TObject;
677 var Action: TCloseAction);

```

```

678 begin
679   IncomesQuery.Active := false;
680 end;
681
682 procedure TIncomesForm.IncomesQueryNGetText(Sender: TField;
683   var Text: String; DisplayText: Boolean);
684 begin
685   Text := IntToStr(Sender.DataSet.RecNo);
686 end;
687
688 procedure TIncomesForm.btnReportClick(Sender: TObject); //Отчет
689 по доходам
690 begin
691   IncRepForm := TIncRepForm.Create(self);
692   IncRepForm.ShowModal;
693   IncRepForm.Free;
694 end;
695
696 procedure TIncomesForm.btnToFileClick(Sender: TObject);
697 //Сохранить в файл
698 begin
699   SaveTTL;
700 end;
701
702 procedure TIncomesForm.btnCloseClick(Sender: TObject);
703 begin
704   Close;
705 end;
706
707 procedure TIncomesForm.btnViewClick(Sender: TObject);
708 //Информация по доходам
709 begin
710   IncInfoForm := TIncInfoForm.Create(self);
711   IncInfoForm.PersonId := IncomesQuery.IPERSON.AsInteger;
712   IncInfoForm.ShowModal;
713   IncInfoForm.Free;
714 end;
715
716 procedure TIncomesForm.IncomesGridDbClick(Sender: TObject);
717 begin
718   btnViewClick(Sender);
719 end;
720
721 end.
722
723
724
725 unit frmSelReq;
726
727 interface
728
729 uses
730   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
731   Forms,
732   Dialogs, StdCtrls, ComCtrls, ExtCtrls;
733
734 type
735   TSelReqForm = class(TForm)
736     Panel1: TPanel;
737     Button2: TButton;
738     btnDone: TButton;
739     Edit4: TEdit;

```

```

740   CheckBox5: TCheckBox;
741   CheckBox4: TCheckBox;
742   Edit3: TEdit;
743   Edit2: TEdit;
744   CheckBox3: TCheckBox;
745   CheckBox2: TCheckBox;
746   Edit1: TEdit;
747   DateTimePicker1: TDateTimePicker;
748   Label1: TLabel;
749   DateTimePicker2: TDateTimePicker;
750   CheckBox1: TCheckBox;
751   procedure Edit1KeyPress(Sender: TObject; var Key: Char);
752   procedure btnDoneClick(Sender: TObject);
753   procedure CheckBox1Click(Sender: TObject);
754   procedure CheckBox2Click(Sender: TObject);
755   procedure CheckBox3Click(Sender: TObject);
756   procedure CheckBox4Click(Sender: TObject);
757   procedure CheckBox5Click(Sender: TObject);
758   private
759     { Private declarations }
760     NewSQL: string;
761   public
762     { Public declarations }
763   end;
764
765   var
766     SelReqForm: TSelReqForm;
767
768   implementation
769
770   Uses
771     frmKontrDoh;
772
773   {$R *.dfm}
774
775   procedure TSelReqForm.Edit1KeyPress(Sender: TObject; var Key:
776   Char);
777   begin
778     if not (Key in ['0'..'9', #8]) then
779       Key := #0;
780   end;
781
782   procedure TSelReqForm.btnDoneClick(Sender: TObject);
783   begin
784     if (DateTimePicker1.Date > DateTimePicker2.Date) or
785       (DateTimePicker2.Date < DateTimePicker1.Date) then begin
786       MessageDlg('Нарушение хронологического порядка дат!',
787 mtWarning, [mbOK], 0);
788       Exit;
789     end;
790
791     NewSQL := '';
792     if CheckBox1.Checked then begin
793       if Length(NewSQL) > 0 then
794         NewSQL := NewSQL + ' and ';
795       NewSQL := NewSQL + '(Requests.rDateBegin >= ' + '''' +
796 FormatDateTime('dd.MM.yyyy', DateTimePicker1.Date) + '''' +
797       ') and (Requests.rDateEnd <= ' + '''' + FormatDateTime
798 ('dd.MM.yyyy', DateTimePicker2.Date) + '''' + ')';
799     end;
800
801     if CheckBox2.Checked then begin

```



```

802   if Length(NewSQL) > 0 then
803     NewSQL := NewSQL + ' and ';
804   NewSQL := NewSQL + '(Persons.pCode = ' + '' + Edit1.Text + '' +
805   ');
806   end;
807   if CheckBox3.Checked then begin
808     if Length(NewSQL) > 0 then
809       NewSQL := NewSQL + ' and ';
810     NewSQL := NewSQL + '(Persons.pName = ' + '' + Edit2.Text + ''
811   + ')';
812   end;
813   if CheckBox4.Checked then begin
814     if Length(NewSQL) > 0 then
815       NewSQL := NewSQL + ' and ';
816     NewSQL := NewSQL + '(Persons.pAddress = ' + '' + Edit3.Text +
817   '' + ')';
818   end;
819   if CheckBox5.Checked then begin
820     if Length(NewSQL) > 0 then
821       NewSQL := NewSQL + ' and ';
822     NewSQL := NewSQL + '(Firms.fName = ' + '' + Edit4.Text + '' +
823   ')';
824   end;
825
826   if Length(NewSQL) > 0 then
827     NewSQL := 'where ' + NewSQL;
828   KontrDohForm.TempSQL := NewSQL;
829   ModalResult := mrOK;
830 end;
831
832 procedure TSelReqForm.CheckBox1Click(Sender: TObject);
833 begin
834   DateTimePicker1.Enabled := CheckBox1.Checked;
835   DateTimePicker2.Enabled := CheckBox1.Checked;
836 end;
837
838 procedure TSelReqForm.CheckBox2Click(Sender: TObject);
839 begin
840   Edit1.Enabled := CheckBox2.Checked;
841 end;
842
843 procedure TSelReqForm.CheckBox3Click(Sender: TObject);
844 begin
845   Edit2.Enabled := CheckBox3.Checked;
846 end;
847
848 procedure TSelReqForm.CheckBox4Click(Sender: TObject);
849 begin
850   Edit3.Enabled := CheckBox4.Checked;
851 end;
852
853 procedure TSelReqForm.CheckBox5Click(Sender: TObject);
854 begin
855   Edit4.Enabled := CheckBox5.Checked;
856 end;
857
858 end.
859
860
861
862 unit frmPersInfo;
863

```

```

864 interface
865
866 uses
867   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
868   Forms,
869   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
870
871 type
872   TPersInfoForm = class(TForm)
873     Panel1: TPanel;
874     Label1: TLabel;
875     Edit1: TEdit;
876     Label2: TLabel;
877     Edit2: TEdit;
878     Label3: TLabel;
879     Edit3: TEdit;
880     Label4: TLabel;
881     Edit4: TEdit;
882     Label5: TLabel;
883     Edit5: TEdit;
884     PersonQuery: TQuery;
885     FamilyQuery: TQuery;
886     PersonSource: TDataSource;
887     FamilySource: TDataSource;
888     PersonQueryPID: TFloatField;
889     PersonQueryPCODE: TStringField;
890     PersonQueryPNAME: TStringField;
891     PersonQueryPADDRESS: TStringField;
892     PersonQueryPEGRPOU: TStringField;
893     Panel2: TPanel;
894     FamilyGrid: TDBGrid;
895     Label7: TLabel;
896     btnCancel: TButton;
897     FamilyQueryFNAME: TStringField;
898     FamilyQueryFRELIATION: TStringField;
899     FamilyQueryFINCTYPE: TStringField;
900     FamilyQueryFINCSUMM: TFloatField;
901     PersonQueryFNAME: TStringField;
902     procedure btnCancelClick(Sender: TObject);
903     procedure FormShow(Sender: TObject);
904   private
905     { Private declarations }
906   public
907     { Public declarations }
908     PersonId: integer;
909   end;
910
911 var
912   PersInfoForm: TPersInfoForm;
913
914 implementation
915
916 uses
917   frmKontrDoh;
918
919
920 {$R *.dfm}
921
922 procedure TPersInfoForm.btnCancelClick(Sender: TObject);
923 begin
924   FamilyQuery.Active := false;
925   PersonQuery.Active := false;

```

```

926 end;
927
928 procedure TPersInfoForm.FormShow(Sender: TObject);
929 begin
930   PersonQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
931   FamilyQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
932   PersonQuery.Params[0].Value := PersonId;
933   FamilyQuery.Params[0].Value := PersonId;
934
935   try
936     Screen.Cursor := crSQLWait;
937     PersonQuery.Active := true;
938     FamilyQuery.Prepare;
939     FamilyQuery.Active := true;
940     Screen.Cursor := crDefault;
941
942     Edit1.Text := PersonQueryPNAME.AsString;
943     Edit2.Text := PersonQueryPCODE.AsString;
944     Edit3.Text := PersonQueryPADDRESS.AsString;
945     Edit4.Text := PersonQueryPEGRPOU.AsString;
946     Edit5.Text := PersonQueryFNAME.AsString;
947
948     FamilyGrid.Columns.Items[0].Title.Caption := 'Ф.И.О.';
949     FamilyGrid.Columns.Items[1].Title.Caption := 'Отношение';
950     FamilyGrid.Columns.Items[2].Title.Caption := 'Вид дохода';
951     FamilyGrid.Columns.Items[3].Title.Caption := 'Сумма дохода';
952   finally
953     end;
954 end;
955
956 end.
957
958
959
960 unit frmQueries;
961
962 interface
963
964 uses
965   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
966   Forms,
967   Dialogs, ExtCtrls, StdCtrls, Grids, DBGrids, DB, DBTables, DBCtrls;
968
969 type
970   TQueriesForm = class(TForm)
971     Panel1: TPanel;
972     Panel2: TPanel;
973     Panel3: TPanel;
974     btnReport: TButton;
975     btnClose: TButton;
976     QueriesGrid: TDBGrid;
977     QueriesQuery: TQuery;
978     QueriesSource: TDataSource;
979     DBNavigator1: TDBNavigator;
980     QueriesQueryQNUMBER: TFloatField;
981     QueriesQueryQINCPERBEG: TDateField;
982     QueriesQueryQREPSUMM: TFloatField;
983     QueriesQueryQINQDATE: TDateField;
984     QueriesQueryQREPDATE: TDateField;
985     QueriesQueryPCODE: TStringField;
986     QueriesQueryPNAME: TStringField;
987     QueriesQueryQINQUERY: TFloatField;

```

```

988   QueriesQueryPeriod: TStringField;
989   btnFamily: TButton;
990   btnIncomes: TButton;
991   btnPayments: TButton;
992   btnSubs: TButton;
993   btnView: TButton;
994   QueriesQueryQID: TFloatField;
995   QueriesQueryQPERSON: TFloatField;
996   procedure FormShow(Sender: TObject);
997   procedure FormClose(Sender: TObject; var Action: TCloseAction);
998   procedure QueriesQueryNGetText(Sender: TField; var Text: String;
999     DisplayText: Boolean);
1000  procedure btnCloseClick(Sender: TObject);
1001  procedure QueriesQueryCalcFields(DataSet: TDataSet);
1002  procedure btnViewClick(Sender: TObject);
1003  procedure btnReportClick(Sender: TObject);
1004  procedure QueriesGridDbClick(Sender: TObject);
1005  procedure btnFamilyClick(Sender: TObject);
1006  procedure btnIncomesClick(Sender: TObject);
1007  procedure btnPaymentsClick(Sender: TObject);
1008  procedure btnSubsClick(Sender: TObject);
1009  private
1010    { Private declarations }
1011    ToFile: boolean;
1012  public
1013    { Public declarations }
1014    InBox: boolean;
1015  end;
1016
1017  var
1018    QueriesForm: TQueriesForm;
1019
1020  implementation
1021
1022  Uses
1023    frmKontrDoh, repQueries, frmPersInfo, frmQueryInfo, frmIncInfo,
1024    frmPayments,
1025    repSubsidy;
1026
1027  {$R *.dfm}
1028
1029  procedure TQueriesForm.FormShow(Sender: TObject);
1030  begin
1031    QueriesQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1032
1033  try
1034    Screen.Cursor := crSQLWait;
1035    QueriesQuery.Active := true;
1036    Screen.Cursor := crDefault;
1037
1038    QueriesGrid.Columns.Items[0].Title.Caption := '№ обращения';
1039    QueriesGrid.Columns.Items[1].Title.Caption := 'Дата запроса';
1040    QueriesGrid.Columns.Items[2].Title.Caption := 'Дата ответа';
1041    QueriesGrid.Columns.Items[3].Title.Caption := 'Идент. Код';
1042    QueriesGrid.Columns.Items[4].Title.Caption := 'Ф.И.О.';
1043    QueriesGrid.Columns.Items[5].Title.Caption := 'Запрос';
1044    QueriesGrid.Columns.Items[6].Title.Caption := 'Ответ';
1045    QueriesGrid.Columns.Items[7].Title.Caption := 'Период дохода';
1046  finally
1047    end;
1048  end;
1049

```

```

1050 procedure TQueriesForm.FormClose(Sender: TObject;
1051   var Action: TCloseAction);
1052 begin
1053   QueriesQuery.Active := false;
1054 end;
1055
1056 procedure TQueriesForm.QueriesQueryNGetText(Sender: TField;
1057   var Text: String; DisplayText: Boolean);
1058 begin
1059   Text := IntToStr(Sender.DataSet.RecNo);
1060 end;
1061
1062 procedure TQueriesForm.btnCloseClick(Sender: TObject);
1063 begin
1064   Close;
1065 end;
1066
1067 procedure TQueriesForm.QueriesQueryCalcFields(DataSet:
1068   TDataSet);
1069 begin
1070   QueriesQueryPeriod.Value := FormatDateTime('MM.yyyy',
1071     QueriesQueryQINCPERBEG.AsDateTime) +
1072     '-' + FormatDateTime('MM.yyyy',
1073     QueriesQueryQINCPERBEG.AsDateTime + 180);
1074 end;
1075
1076 procedure TQueriesForm.btnViewClick(Sender: TObject);
1077 //Информация по запросам
1078 begin
1079   QueryInfoForm := TQueryInfoForm.Create(self);
1080   QueryInfoForm.QueryId := QueriesQueryQID.AsInteger;
1081   QueryInfoForm.ShowModal;
1082   QueryInfoForm.Free;
1083 end;
1084
1085
1086
1087 procedure TQueriesForm.btnReportClick(Sender: TObject); //Печать отчета
1088
1089 begin
1090   QueriesRepForm := TQueriesRepForm.Create(self);
1091   QueriesRepForm.ShowModal;
1092   QueriesRepForm.Free;
1093 end;
1094
1095 procedure TQueriesForm.QueriesGridDbClick(Sender: TObject);
1096 //Информация по запросам
1097 begin
1098   btnViewClick(Sender);
1099 end;
1100
1101 procedure TQueriesForm.btnFamilyClick(Sender: TObject);
1102 //Информация о семье
1103 begin
1104   PersInfoForm := TPersInfoForm.Create(self);
1105   PersInfoForm.PersonId := QueriesQueryQPERSON.AsInteger;
1106   PersInfoForm.ShowModal;
1107   PersInfoForm.Free;
1108 end;
1109
1110 procedure TQueriesForm.btnIncomesClick(Sender: TObject);
1111 //Информация о доходах

```

```

1112 begin
1113   IncInfoForm := TIncInfoForm.Create(self);
1114   IncInfoForm.PersonId := QueriesQueryQPERSON.AsInteger;
1115   IncInfoForm.ShowModal;
1116   IncInfoForm.Free;
1117 end;
1118
1119 procedure TQueriesForm.btnPaymentsClick(Sender: TObject);
1120 //Информация о платежах
1121 begin
1122   PaymentsForm := TPaymentsForm.Create(self);
1123   PaymentsForm.PersonId := QueriesQueryQPERSON.AsInteger;
1124   PaymentsForm.QueryId := QueriesQueryQID.AsInteger;
1125   PaymentsForm.ShowModal;
1126   PaymentsForm.Free;
1127 end;
1128
1129 procedure TQueriesForm.btnSubsClick(Sender: TObject); //Субсидии
1130 begin
1131   SubsRepForm := TSubsRepForm.Create(self);
1132   SubsRepForm.QueryId := QueriesQueryQID.AsInteger;
1133   SubsRepForm.ShowModal;
1134   SubsRepForm.Free;
1135 end;
1136
1137 end.
1138
1139
1140
1141 unit frmSelQuery;
1142
1143 interface
1144
1145 uses
1146   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1147   Forms,
1148   Dialogs, StdCtrls, ComCtrls, ExtCtrls;
1149
1150 type
1151   TSelQueryForm = class(TForm)
1152     Panel1: TPanel;
1153     Button2: TButton;
1154     btnDone: TButton;
1155     Edit4: TEdit;
1156     CheckBox5: TCheckBox;
1157     CheckBox4: TCheckBox;
1158     Edit3: TEdit;
1159     Edit2: TEdit;
1160     CheckBox3: TCheckBox;
1161     CheckBox2: TCheckBox;
1162     Edit1: TEdit;
1163     DateTimePicker1: TDateTimePicker;
1164     Label1: TLabel;
1165     DateTimePicker2: TDateTimePicker;
1166     CheckBox1: TCheckBox;
1167     CheckBox6: TCheckBox;
1168     DateTimePicker3: TDateTimePicker;
1169     DateTimePicker4: TDateTimePicker;
1170     Label2: TLabel;
1171     CheckBox7: TCheckBox;
1172     Edit5: TEdit;
1173     procedure Edit1KeyPress(Sender: TObject; var Key: Char);

```

```

1174 procedure btnDoneClick(Sender: TObject);
1175 procedure CheckBox1Click(Sender: TObject);
1176 procedure CheckBox2Click(Sender: TObject);
1177 procedure CheckBox3Click(Sender: TObject);
1178 procedure CheckBox4Click(Sender: TObject);
1179 procedure CheckBox5Click(Sender: TObject);
1180 procedure CheckBox6Click(Sender: TObject);
1181 procedure CheckBox7Click(Sender: TObject);
1182 private
1183   { Private declarations }
1184   NewSQL: string;
1185 public
1186   { Public declarations }
1187 end;
1188
1189 var
1190   SelQueryForm: TSelQueryForm;
1191
1192 implementation
1193
1194 Uses
1195   frmKontrDoh;
1196
1197 {$R *.dfm}
1198
1199 procedure TSelQueryForm.Edit1KeyPress(Sender: TObject; var Key:
1200 Char);
1201 begin
1202   if not (Key in ['0'..'9', #8]) then
1203     Key := #0;
1204 end;
1205
1206 procedure TSelQueryForm.btnDoneClick(Sender: TObject);
1207 begin
1208   if (DateTimePicker1.Date > DateTimePicker2.Date) or
1209     (DateTimePicker2.Date < DateTimePicker1.Date) or
1210     (DateTimePicker1.Date > DateTimePicker3.Date) or
1211     (DateTimePicker2.Date < DateTimePicker4.Date) then begin
1212     MessageDlg('Нарушение хронологического порядка дат!',
1213 mtWarning, [mbOK], 0);
1214     Exit;
1215   end;
1216
1217   NewSQL := '';
1218   if CheckBox1.Checked then begin
1219     if Length(NewSQL) > 0 then
1220       NewSQL := NewSQL + ' and ';
1221     NewSQL := NewSQL + '(Queries.qInqDate >= ' + '''' +
1222 FormatDateTime('dd.MM.yyyy', DateTimePicker1.Date) + '''' +
1223     ') and (Queries.qInqDate <= ' + '''' + FormatDateTime
1224 ('dd.MM.yyyy', DateTimePicker2.Date) + '''' + ')';
1225   end;
1226   if CheckBox6.Checked then begin
1227     if Length(NewSQL) > 0 then
1228       NewSQL := NewSQL + ' and ';
1229     NewSQL := NewSQL + '(Queries.qRepDate >= ' + '''' +
1230 FormatDateTime('dd.MM.yyyy', DateTimePicker3.Date) + '''' +
1231     ') and (Queries.qRepDate <= ' + '''' + FormatDateTime
1232 ('dd.MM.yyyy', DateTimePicker4.Date) + '''' + ')';
1233   end;
1234   if CheckBox7.Checked then begin
1235     if Length(NewSQL) > 0 then

```

```

1236   NewSQL := NewSQL + ' and ';
1237   NewSQL := NewSQL + '(Queries.qNumber = ' + '' + Edit5.Text +
1238   '' + ')';
1239   end;
1240   if CheckBox2.Checked then begin
1241     if Length(NewSQL) > 0 then
1242       NewSQL := NewSQL + ' and ';
1243     NewSQL := NewSQL + '(Persons.pCode = ' + '' + Edit1.Text + '' +
1244     ')';
1245   end;
1246   if CheckBox3.Checked then begin
1247     if Length(NewSQL) > 0 then
1248       NewSQL := NewSQL + ' and ';
1249     NewSQL := NewSQL + '(Persons.pName = ' + '' + Edit2.Text + ''
1250     + ')';
1251   end;
1252   if CheckBox4.Checked then begin
1253     if Length(NewSQL) > 0 then
1254       NewSQL := NewSQL + ' and ';
1255     NewSQL := NewSQL + '(Persons.pAddress = ' + '' + Edit3.Text +
1256     '' + ')';
1257   end;
1258   if CheckBox5.Checked then begin
1259     if Length(NewSQL) > 0 then
1260       NewSQL := NewSQL + ' and ';
1261     NewSQL := NewSQL + '(Firms.fName = ' + '' + Edit4.Text + '' +
1262     ')';
1263   end;
1264
1265   if Length(NewSQL) > 0 then
1266     NewSQL := 'where ' + NewSQL;
1267   KontrDohForm.TempSQL := NewSQL;
1268   ModalResult := mrOK;
1269 end;
1270
1271 procedure TSelQueryForm.CheckBox1Click(Sender: TObject);
1272 begin
1273   DateTimePicker1.Enabled := CheckBox1.Checked;
1274   DateTimePicker2.Enabled := CheckBox1.Checked;
1275 end;
1276
1277 procedure TSelQueryForm.CheckBox2Click(Sender: TObject);
1278 begin
1279   Edit1.Enabled := CheckBox2.Checked;
1280 end;
1281
1282 procedure TSelQueryForm.CheckBox3Click(Sender: TObject);
1283 begin
1284   Edit2.Enabled := CheckBox3.Checked;
1285 end;
1286
1287 procedure TSelQueryForm.CheckBox4Click(Sender: TObject);
1288 begin
1289   Edit3.Enabled := CheckBox4.Checked;
1290 end;
1291
1292 procedure TSelQueryForm.CheckBox5Click(Sender: TObject);
1293 begin
1294   Edit4.Enabled := CheckBox5.Checked;
1295 end;
1296
1297 procedure TSelQueryForm.CheckBox6Click(Sender: TObject);

```



```

1298 begin
1299   DateTimePicker3.Enabled := CheckBox6.Checked;
1300   DateTimePicker4.Enabled := CheckBox6.Checked;
1301 end;
1302
1303 procedure TSelQueryForm.CheckBox7Click(Sender: TObject);
1304 begin
1305   Edit5.Enabled := CheckBox7.Checked;
1306 end;
1307
1308 end.
1309
1310
1311
1312 unit frmQueryInfo;
1313
1314 interface
1315
1316 uses
1317   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1318   Forms,
1319   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1320
1321 type
1322   TQueryInfoForm = class(TForm)
1323     Panel1: TPanel;
1324     Label1: TLabel;
1325     Edit1: TEdit;
1326     Label2: TLabel;
1327     Edit2: TEdit;
1328     Label3: TLabel;
1329     Edit3: TEdit;
1330     Label4: TLabel;
1331     Edit4: TEdit;
1332     Label5: TLabel;
1333     Edit5: TEdit;
1334     btnCancel: TButton;
1335     Label6: TLabel;
1336     Edit6: TEdit;
1337     Label7: TLabel;
1338     Edit7: TEdit;
1339     Label8: TLabel;
1340     Edit8: TEdit;
1341     Label9: TLabel;
1342     Edit9: TEdit;
1343     Label10: TLabel;
1344     Edit10: TEdit;
1345     Label11: TLabel;
1346     Edit11: TEdit;
1347     Label12: TLabel;
1348     Edit12: TEdit;
1349     QueryQuery: TQuery;
1350     Label13: TLabel;
1351     Edit13: TEdit;
1352     QueryQueryQNUMBER: TFloatField;
1353     QueryQueryPCODE: TStringField;
1354     QueryQueryPNAME: TStringField;
1355     QueryQueryPADDRESS: TStringField;
1356     QueryQueryPEGRPOU: TStringField;
1357     QueryQueryFNAME: TStringField;
1358     QueryQueryQINQUERY: TFloatField;
1359     QueryQueryQREPLY: TFloatField;

```

```

1360 QueryQueryQINQSUMM: TFloatField;
1361 QueryQueryQREPSUMM: TFloatField;
1362 QueryQueryQINQDATE: TDateField;
1363 QueryQueryQREPDATE: TDateField;
1364 QueryQueryQINCPERBEG: TDateField;
1365 QueryQueryPeriod: TStringField;
1366 procedure btnCancelClick(Sender: TObject);
1367 procedure FormShow(Sender: TObject);
1368 procedure QueryQueryCalcFields(DataSet: TDataSet);
1369 private
1370   { Private declarations }
1371 public
1372   { Public declarations }
1373   QueryId: integer;
1374 end;
1375
1376 var
1377   QueryInfoForm: TQueryInfoForm;
1378
1379 implementation
1380
1381 uses
1382   frmKontrDoh;
1383
1384 {$R *.dfm}
1385
1386 procedure TQueryInfoForm.btnCancelClick(Sender: TObject);
1387 begin
1388   QueryQuery.Active := false;
1389 end;
1390
1391 procedure TQueryInfoForm.FormShow(Sender: TObject);
1392 begin
1393   QueryQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1394   QueryQuery.Params[0].Value := QueryId;
1395
1396   try
1397     Screen.Cursor := crSQLWait;
1398     QueryQuery.Active := true;
1399     Screen.Cursor := crDefault;
1400     Edit1.Text := QueryQueryPNAME.AsString;
1401     Edit2.Text := QueryQueryPCODE.AsString;
1402     Edit3.Text := QueryQueryPADDRESS.AsString;
1403     Edit4.Text := QueryQueryPEGRPOU.AsString;
1404     Edit5.Text := QueryQueryFNAME.AsString;
1405     Edit6.Text := QueryQueryQNUMBER.AsString;
1406     Edit7.Text := QueryQueryQINQUERY.AsString;
1407     Edit8.Text := QueryQueryQREPLY.AsString;
1408     Edit9.Text := QueryQueryQINQSUMM.AsString;
1409     Edit10.Text := QueryQueryQREPSUMM.AsString;
1410     Edit11.Text := QueryQueryQINQDATE.AsString;
1411     Edit12.Text := QueryQueryQREPDATE.AsString;
1412     Edit13.Text := QueryQueryPeriod.AsString;
1413   finally
1414     end;
1415 end;
1416
1417 procedure TQueryInfoForm.QueryQueryCalcFields(DataSet:
1418 TDataSet);
1419 begin
1420   QueryQueryPeriod.Value := FormatDateTime('MM.yyyy',
1421 QueryQueryQINCPERBEG.AsDateTime) +

```

```

1422     '-' + FormatDateTime('MM.yyyy',
1423 QueryQueryQINCPERBEG.AsDateTime + 180);
1424 end;
1425 end.
1426
1427 unit frmQueryInfo;
1428
1429 interface
1430
1431 uses
1432   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1433   Forms,
1434   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1435
1436 type
1437   TQueryInfoForm = class(TForm)
1438     Panel1: TPanel;
1439     Label1: TLabel;
1440     Edit1: TEdit;
1441     Label2: TLabel;
1442     Edit2: TEdit;
1443     Label3: TLabel;
1444     Edit3: TEdit;
1445     Label4: TLabel;
1446     Edit4: TEdit;
1447     Label5: TLabel;
1448     Edit5: TEdit;
1449     btnCancel: TButton;
1450     Label6: TLabel;
1451     Edit6: TEdit;
1452     Label7: TLabel;
1453     Edit7: TEdit;
1454     Label8: TLabel;
1455     Edit8: TEdit;
1456     Label9: TLabel;
1457     Edit9: TEdit;
1458     Label10: TLabel;
1459     Edit10: TEdit;
1460     Label11: TLabel;
1461     Edit11: TEdit;
1462     Label12: TLabel;
1463     Edit12: TEdit;
1464     QueryQuery: TQuery;
1465     Label13: TLabel;
1466     Edit13: TEdit;
1467     QueryQueryQNUMBER: TFloatField;
1468     QueryQueryPCODE: TStringField;
1469     QueryQueryPNAME: TStringField;
1470     QueryQueryPADDRESS: TStringField;
1471     QueryQueryPEGRPOU: TStringField;
1472     QueryQueryFNAME: TStringField;
1473     QueryQueryQINQUERY: TFloatField;
1474     QueryQueryQREPLY: TFloatField;
1475     QueryQueryQINQSUMM: TFloatField;
1476     QueryQueryQREPSUMM: TFloatField;
1477     QueryQueryQINQDATE: TDateField;
1478     QueryQueryQREPDATE: TDateField;
1479     QueryQueryQINCPERBEG: TDateField;
1480     QueryQueryPeriod: TStringField;
1481     procedure btnCancelClick(Sender: TObject);
1482     procedure FormShow(Sender: TObject);
1483     procedure QueryQueryCalcFields(DataSet: TDataSet);

```

```

1484 private
1485   { Private declarations }
1486 public
1487   { Public declarations }
1488   QueryId: integer;
1489 end;
1490
1491 var
1492   QueryInfoForm: TQueryInfoForm;
1493
1494 implementation
1495
1496 uses
1497   frmKontrDoh;
1498
1499 {$R *.dfm}
1500
1501
1502
1503
1504 procedure TQueryInfoForm.btnCancelClick(Sender: TObject);
1505 begin
1506   QueryQuery.Active := false;
1507 end;
1508
1509 procedure TQueryInfoForm.FormShow(Sender: TObject);
1510 begin
1511   QueryQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1512   QueryQuery.Params[0].Value := QueryId;
1513
1514   try
1515     Screen.Cursor := crSQLWait;
1516     QueryQuery.Active := true;
1517     Screen.Cursor := crDefault;
1518
1519     Edit1.Text := QueryQueryPNAME.AsString;
1520     Edit2.Text := QueryQueryPCODE.AsString;
1521     Edit3.Text := QueryQueryPADDRESS.AsString;
1522     Edit4.Text := QueryQueryPEGRPOU.AsString;
1523     Edit5.Text := QueryQueryFNAME.AsString;
1524     Edit6.Text := QueryQueryQNUMBER.AsString;
1525     Edit7.Text := QueryQueryQINQUERY.AsString;
1526     Edit8.Text := QueryQueryQREPLY.AsString;
1527     Edit9.Text := QueryQueryQINQSUMM.AsString;
1528     Edit10.Text := QueryQueryQREPSUMM.AsString;
1529     Edit11.Text := QueryQueryQINQDATE.AsString;
1530     Edit12.Text := QueryQueryQREPDATE.AsString;
1531     Edit13.Text := QueryQueryPeriod.AsString;
1532   finally
1533     end;
1534 end;
1535
1536 procedure TQueryInfoForm.QueryQueryCalcFields(DataSet:
1537 TDataSet);
1538 begin
1539   QueryQueryPeriod.Value := FormatDateTime('MM.yyyy',
1540 QueryQueryQINCPERBEG.AsDateTime) +
1541   '-' + FormatDateTime('MM.yyyy',
1542 QueryQueryQINCPERBEG.AsDateTime + 180);
1543 end;
1544
1545 end.

```

```

1546
1547
1548
1549 unit frmIncInfo;
1550
1551 interface
1552
1553 uses
1554   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1555   Forms,
1556   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1557
1558 type
1559   TIncInfoForm = class(TForm)
1560     Panel1: TPanel;
1561     Label1: TLabel;
1562     Edit1: TEdit;
1563     Label2: TLabel;
1564     Edit2: TEdit;
1565     Label3: TLabel;
1566     Edit3: TEdit;
1567     IncomeSource: TDataSource;
1568     btnCancel: TButton;
1569     IncomeQuery: TQuery;
1570     Panel2: TPanel;
1571     IncomeGrid: TDBGrid;
1572     Label7: TLabel;
1573     PersonQuery: TQuery;
1574     PersonSource: TDataSource;
1575     PersonQueryPID: TFloatField;
1576     PersonQueryPCODE: TStringField;
1577     PersonQueryPNAME: TStringField;
1578     PersonQueryPEGRPOU: TStringField;
1579     IncomeQueryITYPE: TStringField;
1580     IncomeQueryISUMM: TFloatField;
1581     IncomeQueryIDATEBEGIN: TDateField;
1582     IncomeQueryIID: TFloatField;
1583     procedure btnCancelClick(Sender: TObject);
1584     procedure FormShow(Sender: TObject);
1585     procedure IncomeGridDbClick(Sender: TObject);
1586   private
1587     { Private declarations }
1588   public
1589     { Public declarations }
1590     PersonId: integer;
1591   end;
1592
1593 var
1594   IncInfoForm: TIncInfoForm;
1595
1596 implementation
1597
1598 uses
1599   frmKontrDoh, frmPerIncInfo;
1600
1601 {$R *.dfm}
1602
1603 procedure TIncInfoForm.btnCancelClick(Sender: TObject);
1604 begin
1605   PersonQuery.Active := false;
1606   IncomeQuery.Active := false;

```

```

1608 end;
1609
1610 procedure TIncInfoForm.FormShow(Sender: TObject);
1611 begin
1612   PersonQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1613   IncomeQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1614   PersonQuery.Params[0].Value := PersonId;
1615   IncomeQuery.Params[0].Value := PersonId;
1616
1617   try
1618     Screen.Cursor := crSQLWait;
1619     PersonQuery.Active := true;
1620     IncomeQuery.Prepare;
1621     IncomeQuery.Active := true;
1622     Screen.Cursor := crDefault;
1623
1624     Edit1.Text := PersonQueryPNAME.AsString;
1625     Edit2.Text := PersonQueryPCODE.AsString;
1626     Edit3.Text := PersonQueryPEGRPOU.AsString;
1627
1628     IncomeGrid.Columns.Items[0].Title.Caption := 'Дата';
1629     IncomeGrid.Columns.Items[1].Title.Caption := 'Вид дохода';
1630     IncomeGrid.Columns.Items[2].Title.Caption := 'Сумма';
1631   finally
1632     end;
1633 end;
1634
1635 procedure TIncInfoForm.IncomeGridDbClick(Sender: TObject);
1636 //Просмотр доходов за пол года
1637 begin
1638   PerIncInfoForm := TPerIncInfoForm.Create(self);
1639   PerIncInfoForm.IncomeId := IncomeQueryIID.AsInteger;
1640   PerIncInfoForm.ShowModal;
1641   PerIncInfoForm.Free;
1642 end;
1643
1644 end.
1645
1646
1647
1648 unit frmPerIncInfo;
1649
1650 interface
1651
1652 uses
1653   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1654   Forms,
1655   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1656
1657 type
1658   TPerIncInfoForm = class(TForm)
1659     Panel1: TPanel;
1660     Label1: TLabel;
1661     Edit1: TEdit;
1662     Edit2: TEdit;
1663     Label3: TLabel;
1664     Label4: TLabel;
1665     Edit4: TEdit;
1666     Label5: TLabel;
1667     Edit5: TEdit;
1668     btnCancel: TButton;
1669     Label7: TLabel;

```

```

1670 Edit7: TEdit;
1671 Label8: TLabel;
1672 Edit8: TEdit;
1673 PerIncQuery: TQuery;
1674 Label13: TLabel;
1675 Edit12: TEdit;
1676 Edit3: TEdit;
1677 Edit6: TEdit;
1678 Edit9: TEdit;
1679 Edit10: TEdit;
1680 Edit11: TEdit;
1681 PerIncQueryPNAME: TStringField;
1682 PerIncQueryPCODE: TStringField;
1683 PerIncQueryPEGRPOU: TStringField;
1684 PerIncQueryITYPE: TStringField;
1685 PerIncQueryISUMM: TFloatField;
1686 PerIncQueryIDATEBEGIN: TDateField;
1687 PerIncQueryIDATEEND: TDateField;
1688 procedure btnCancelClick(Sender: TObject);
1689 procedure FormShow(Sender: TObject);
1690 private
1691   { Private declarations }
1692 public
1693   { Public declarations }
1694   IncomeId: integer;
1695 end;
1696
1697 var
1698   PerIncInfoForm: TPerIncInfoForm;
1699
1700 implementation
1701
1702 uses
1703   frmKontrDoh, frmQueries;
1704
1705 {$R *.dfm}
1706
1707 procedure TPerIncInfoForm.btnCancelClick(Sender: TObject);
1708 begin
1709   PerIncQuery.Active := false;
1710 end;
1711
1712 procedure TPerIncInfoForm.FormShow(Sender: TObject);
1713 begin
1714   PerIncQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1715   PerIncQuery.Params[0].Value := IncomeId;
1716
1717   try
1718     Screen.Cursor := crSQLWait;
1719     PerIncQuery.Active := true;
1720     Screen.Cursor := crDefault;
1721
1722     Edit1.Text := PerIncQueryPNAME.AsString;
1723     Edit2.Text := PerIncQueryPCODE.AsString;
1724     Edit3.Text := IntToStr(Round(PerIncQueryISUMM.AsFloat) div
1725 6);
1726     Edit4.Text := PerIncQueryPEGRPOU.AsString;
1727     Edit5.Text := PerIncQueryITYPE.AsString;
1728     Edit6.Text := IntToStr(Round(PerIncQueryISUMM.AsFloat) div
1729 6);
1730     Edit7.Text := PerIncQueryISUMM.AsString + ' (' +
1731     FormatDateTime('MM/yy',

```

```

1732 PerIncQueryIDATEBEGIN.AsDateTime) + ' - ' +
1733   FormatDateTime('MM/yy',
1734 PerIncQueryIDATEEND.AsDateTime) + ');
1735   Edit8.Text := PerIncQueryISUMM.AsString;
1736   Edit9.Text := IntToStr(Round(PerIncQueryISUMM.AsFloat) div
1737 6);
1738   Edit10.Text := IntToStr(Round(PerIncQueryISUMM.AsFloat) div
1739 6);
1740   Edit11.Text := IntToStr(Round(PerIncQueryISUMM.AsFloat) div
1741 6);
1742   Edit12.Text := IntToStr(Round(PerIncQueryISUMM.AsFloat) div
1743 6);
1744   finally
1745   end;
1746 end;
1747
1748 end.
1749
1750
1751
1752
1753
1754
1755 unit frmPayments;
1756
1757 interface
1758
1759 uses
1760   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1761   Forms,
1762   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1763
1764 type
1765   TPaymentsForm = class(TForm)
1766     Panel1: TPanel;
1767     Label1: TLabel;
1768     Edit1: TEdit;
1769     Label2: TLabel;
1770     Edit2: TEdit;
1771     Label3: TLabel;
1772     Edit3: TEdit;
1773     PaymentsSource: TDataSource;
1774     btnCancel: TButton;
1775     PaymentsQuery: TQuery;
1776     Panel2: TPanel;
1777     PaymentsGrid: TDBGrid;
1778     PersonQuery: TQuery;
1779     PersonSource: TDataSource;
1780     PersonQueryPID: TFloatField;
1781     PersonQueryPCODE: TStringField;
1782     PersonQueryPNAME: TStringField;
1783     PersonQueryPEGRPOU: TStringField;
1784     PaymentsQueryOACCOUNT: TStringField;
1785     PaymentsQuerySSHORTNAME: TStringField;
1786     PaymentsQueryONAME: TStringField;
1787     PaymentsQuerySPAY: TFloatField;
1788     PaymentsQuerySSUBSIDY: TFloatField;
1789     PaymentsQueryPDEBT: TFloatField;
1790     PaymentsQueryPID: TFloatField;
1791     procedure btnCancelClick(Sender: TObject);
1792     procedure FormShow(Sender: TObject);
1793     procedure PaymentsGridDbClick(Sender: TObject);

```



```

1794 private
1795   { Private declarations }
1796 public
1797   { Public declarations }
1798   PersonId, QueryId: integer;
1799 end;
1800
1801 var
1802   PaymentsForm: TPaymentsForm;
1803
1804 implementation
1805
1806 uses
1807   frmKontrDoh, frmServiceInfo;
1808
1809
1810 {$R *.dfm}
1811
1812 procedure TPaymentsForm.btnCancelClick(Sender: TObject);
1813 begin
1814   PersonQuery.Active := false;
1815   PaymentsQuery.Active := false;
1816 end;
1817
1818 procedure TPaymentsForm.FormShow(Sender: TObject);
1819 begin
1820   PersonQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1821   PaymentsQuery.DatabaseName := KontrDohForm.MainDir +
1822 '\DBF';
1823   PersonQuery.Params[0].Value := PersonId;
1824   PaymentsQuery.Params[0].Value := PersonId;
1825
1826   try
1827     Screen.Cursor := crSQLWait;
1828     PersonQuery.Active := true;
1829     PaymentsQuery.Active := true;
1830     Screen.Cursor := crDefault;
1831
1832     Edit1.Text := PersonQueryPNAME.AsString;
1833     Edit2.Text := PersonQueryPCODE.AsString;
1834     Edit3.Text := PersonQueryPEGRPOU.AsString;
1835
1836     PaymentsGrid.Columns.Items[0].Title.Caption := 'Счет';
1837     PaymentsGrid.Columns.Items[1].Title.Caption := 'Организация';
1838     PaymentsGrid.Columns.Items[2].Title.Caption := 'Вид';
1839     PaymentsGrid.Columns.Items[3].Title.Caption := 'Плата';
1840     PaymentsGrid.Columns.Items[4].Title.Caption := 'Субсидия';
1841     PaymentsGrid.Columns.Items[5].Title.Caption := 'Долг';
1842   finally
1843     end;
1844 end;
1845
1846 procedure TPaymentsForm.PaymentsGridDbClick(Sender: TObject);
1847 //Информация по услуге
1848 begin
1849   ServiceInfoForm := TServiceInfoForm.Create(self);
1850   ServiceInfoForm.PaymentId := PaymentsQueryPID.AsInteger;
1851   ServiceInfoForm.QueryId := QueryID;
1852   ServiceInfoForm.ShowModal;
1853   ServiceInfoForm.Free;
1854 end;
1855

```

```

1856 end.
1857
1858
1859
1860 unit frmServiceInfo;
1861
1862 interface
1863
1864 uses
1865   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1866   Forms,
1867   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1868
1869 type
1870   TServiceInfoForm = class(TForm)
1871     Panel1: TPanel;
1872     Label1: TLabel;
1873     Edit1: TEdit;
1874     Label2: TLabel;
1875     Edit2: TEdit;
1876     Label3: TLabel;
1877     Edit3: TEdit;
1878     Label4: TLabel;
1879     Edit4: TEdit;
1880     Label5: TLabel;
1881     Edit5: TEdit;
1882     btnCancel: TButton;
1883     Label6: TLabel;
1884     Edit6: TEdit;
1885     Label7: TLabel;
1886     Edit7: TEdit;
1887     Label8: TLabel;
1888     Edit8: TEdit;
1889     ServiceQuery: TQuery;
1890     ServiceQueryONAME: TStringField;
1891     ServiceQueryOACCOUNT: TStringField;
1892     ServiceQuerySPAY: TFloatField;
1893     ServiceQueryPDEBT: TFloatField;
1894     ServiceQuerySNAME: TStringField;
1895     QueriesQuery: TQuery;
1896     QueriesQueryQNUMBER: TFloatField;
1897     procedure btnCancelClick(Sender: TObject);
1898     procedure FormShow(Sender: TObject);
1899   private
1900     { Private declarations }
1901   public
1902     { Public declarations }
1903     PaymentId, QueryId: integer;
1904   end;
1905
1906 var
1907   ServiceInfoForm: TServiceInfoForm;
1908
1909 implementation
1910
1911 uses
1912   frmKontrDoh, frmQueries;
1913
1914 {$R *.dfm}
1915
1916
1917 procedure TServiceInfoForm.btnCancelClick(Sender: TObject);

```

```

1918 begin
1919   ServiceQuery.Active := false;
1920   QueriesQuery.Active := false;
1921 end;
1922
1923 procedure TServiceInfoForm.FormShow(Sender: TObject);
1924 begin
1925   ServiceQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1926   QueriesQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
1927   ServiceQuery.Params[0].Value := PaymentId;
1928   QueriesQuery.Params[0].Value := QueryId;
1929
1930 try
1931   Screen.Cursor := crSQLWait;
1932   ServiceQuery.Active := true;
1933   QueriesQuery.Active := true;
1934   Screen.Cursor := crDefault;
1935
1936   Edit1.Text := ServiceQueryONAME.AsString;
1937   Edit2.Text := ServiceQueryOACCOUNT.AsString;
1938   Edit3.Text := ServiceQuerySPAY.AsString;
1939   Edit4.Text := ServiceQuerySPAY.AsString;
1940   Edit5.Text := ServiceQueryPDEBT.AsString;
1941   Edit6.Text := QueriesQueryQNUMBER.AsString;
1942   Edit7.Text := ServiceQuerySNAME.AsString;
1943   Edit8.Text := ServiceQuerySPAY.AsString;
1944 finally
1945   end;
1946 end;
1947
1948 end.
1949
1950
1951
1952 unit frmFirms;
1953
1954 interface
1955
1956 uses
1957   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
1958   Forms,
1959   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
1960
1961 type
1962   TFirmsForm = class(TForm)
1963     Panel1: TPanel;
1964     btnClose: TButton;
1965     Panel2: TPanel;
1966     FirmsGrid: TDBGrid;
1967     FirmsQuery: TQuery;
1968     FirmsSource: TDataSource;
1969     FirmsQueryFID: TFloatField;
1970     FirmsQueryFACCOUNT: TStringField;
1971     FirmsQueryFNAME: TStringField;
1972     FirmsQueryFADDRESS: TStringField;
1973     FirmsQueryFDIRECTOR: TStringField;
1974     procedure FormClose(Sender: TObject; var Action: TCloseAction);
1975     procedure FormShow(Sender: TObject);
1976   private
1977     { Private declarations }
1978   public
1979     { Public declarations }

```

```

1980 end;
1981
1982 var
1983   FirmsForm: TFirmsForm;
1984
1985 implementation
1986
1987 uses
1988   frmKontrDoh;
1989
1990 {$R *.dfm}
1991
1992
1993 procedure TFirmsForm.FormClose(Sender: TObject; var Action:
1994   TCloseAction);
1995 begin
1996   FirmsQuery.Active := false;
1997 end;
1998
1999 procedure TFirmsForm.FormShow(Sender: TObject);
2000 begin
2001   FirmsQuery.DatabaseName := KontrDohForm.MainDir + '\DBF';
2002
2003   try
2004     Screen.Cursor := crSQLWait;
2005     FirmsQuery.Active := true;
2006     Screen.Cursor := crDefault;
2007
2008     FirmsGrid.Columns.Items[0].Title.Caption := 'Счет';
2009     FirmsGrid.Columns.Items[1].Title.Caption := 'Наименование';
2010     FirmsGrid.Columns.Items[2].Title.Caption := 'Адрес';
2011     FirmsGrid.Columns.Items[3].Title.Caption := 'Директор';
2012   finally
2013     end;
2014 end;
2015
2016 end.
2017
2018
2019
2020 unit frmOrganizations;
2021
2022 interface
2023
2024 uses
2025   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
2026   Forms,
2027   Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, DBGrids;
2028
2029 type
2030   TOrganizationsForm = class(TForm)
2031     Panel1: TPanel;
2032     btnClose: TButton;
2033     Panel2: TPanel;
2034     OrganizationsGrid: TDBGrid;
2035     OrganizationsQuery: TQuery;
2036     OrganizationsSource: TDataSource;
2037     OrganizationsQueryOID: TFloatField;
2038     OrganizationsQueryOACCOUNT: TStringField;
2039     OrganizationsQueryONAME: TStringField;
2040     OrganizationsQueryOEGRPOU: TStringField;
2041     procedure FormClose(Sender: TObject; var Action: TCloseAction);

```

```

2042 procedure FormShow(Sender: TObject);
2043 private
2044   { Private declarations }
2045 public
2046   { Public declarations }
2047 end;
2048
2049 var
2050   OrganizationsForm: TOrganizationsForm;
2051
2052 implementation
2053
2054 uses
2055   frmKontrDoh;
2056
2057 {$R *.dfm}
2058
2059 procedure TOrganizationsForm.FormClose(Sender: TObject; var
2060 Action: TCloseAction);
2061 begin
2062   OrganizationsQuery.Active := false;
2063 end;
2064
2065 procedure TOrganizationsForm.FormShow(Sender: TObject);
2066 begin
2067   OrganizationsQuery.DatabaseName := KontrDohForm.MainDir +
2068   '\DBF';
2069
2070   try
2071     Screen.Cursor := crSQLWait;
2072     OrganizationsQuery.Active := true;
2073     Screen.Cursor := crDefault;
2074
2075     OrganizationsGrid.Columns.Items[0].Title.Caption := 'Счет';
2076     OrganizationsGrid.Columns.Items[1].Title.Caption :=
2077     'Наименование';
2078     OrganizationsGrid.Columns.Items[2].Title.Caption := 'Код
2079     ЕГРПОУ';
2080   finally
2081     end;
2082 end;
2083
2084 end.
2085
2086
2087
2088 unit repRequests;
2089
2090 interface
2091
2092 uses
2093   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
2094   Forms,
2095   Dialogs, QuickRpt, QRCtrls, QRPrntr, ExtCtrls, StdCtrls;
2096
2097 type
2098   TReqRepForm = class(TForm)
2099     Panel1: TPanel;
2100     QRPreview1: TQRPreview;
2101     Panel2: TPanel;
2102     QuickRep1: TQuickRep;
2103     QRBand1: TQRBand;

```

```

2104   QRBand2: TQRBand;
2105   QRBand3: TQRBand;
2106   QRLabel1: TQRLabel;
2107   QRLabel2: TQRLabel;
2108   QRLabel3: TQRLabel;
2109   QRLabel4: TQRLabel;
2110   QRLabel5: TQRLabel;
2111   QRLabel6: TQRLabel;
2112   QRLabel7: TQRLabel;
2113   QRLabel8: TQRLabel;
2114   QRLabel9: TQRLabel;
2115   QRLabel10: TQRLabel;
2116   QRLabel11: TQRLabel;
2117   QRLabel12: TQRLabel;
2118   QRDBText1: TQRDBText;
2119   QRDBText2: TQRDBText;
2120   QRDBText3: TQRDBText;
2121   QRDBText4: TQRDBText;
2122   QRDBText5: TQRDBText;
2123   QRDBText6: TQRDBText;
2124   QRDBText7: TQRDBText;
2125   QRDBText8: TQRDBText;
2126   QRDBText9: TQRDBText;
2127   QRDBText10: TQRDBText;
2128   QRDBText11: TQRDBText;
2129   btnClose: TButton;
2130   btnPrint: TButton;
2131   procedure FormCreate(Sender: TObject);
2132   procedure QRDBText1Print(sender: TObject; var Value: String);
2133   procedure btnCloseClick(Sender: TObject);
2134   procedure QuickRep1Preview(Sender: TObject);
2135   procedure btnPrintClick(Sender: TObject);
2136   procedure QRLabel12Print(sender: TObject; var Value: String);
2137   private
2138     { Private declarations }
2139   public
2140     { Public declarations }
2141   end;
2142
2143   var
2144     ReqRepForm: TReqRepForm;
2145
2146   implementation
2147
2148   Uses
2149     frmRequests;
2150
2151   {$R *.dfm}
2152
2153   procedure TReqRepForm.FormCreate(Sender: TObject);
2154   begin
2155     WindowState := wsMaximized;
2156     QuickRep1.SendToBack;
2157     QuickRep1.PreviewModeless;
2158   end;
2159
2160
2161
2162   procedure TReqRepForm.QRDBText1Print(sender: TObject; var
2163   Value: String);
2164   begin
2165     Value := ' ' + Value;

```

```

2166 end;
2167
2168 procedure TReqRepForm.btnCloseClick(Sender: TObject);
2169 begin
2170   Close;
2171 end;
2172
2173 procedure TReqRepForm.QuickRep1Preview(Sender: TObject);
2174 begin
2175   QRPreview1.QRPrinter := TQRPrinter(Sender);
2176 end;
2177
2178 procedure TReqRepForm.btnPrintClick(Sender: TObject);
2179 begin
2180   QuickRep1.PrinterSetup;
2181   if QuickRep1.Tag = 0 then
2182     QuickRep1.Print;
2183 end;
2184
2185
2186 procedure TReqRepForm.QRLabel12Print(sender: TObject; var
2187 Value: String);
2188 begin
2189   Value := Format(Value, [FormatDateTime('hh:mm dd.MM.yyyy ?',
2190 Now)]);
2191 end;
2192
2193 end.
2194
2195
2196
2197 unit repIncomes;
2198
2199 interface
2200
2201 uses
2202   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
2203   Forms,
2204   Dialogs, QuickRpt, QRCtrls, QRPrntr, ExtCtrls, StdCtrls;
2205
2206 type
2207   TIncRepForm = class(TForm)
2208     Panel1: TPanel;
2209     QRPreview1: TQRPreview;
2210     Panel2: TPanel;
2211     QuickRep1: TQuickRep;
2212     QRBand1: TQRBand;
2213     QRBand2: TQRBand;
2214     QRBand3: TQRBand;
2215     QRLabel1: TQRLabel;
2216     QRLabel2: TQRLabel;
2217     QRLabel3: TQRLabel;
2218     QRLabel4: TQRLabel;
2219     QRLabel5: TQRLabel;
2220     QRLabel6: TQRLabel;
2221     QRLabel7: TQRLabel;
2222     QRLabel10: TQRLabel;
2223     QRLabel11: TQRLabel;
2224     QRLabel12: TQRLabel;
2225     QRDBText1: TQRDBText;
2226     QRDBText2: TQRDBText;
2227     QRDBText3: TQRDBText;

```

```

2228   QRDBText4: TQRDBText;
2229   QRDBText5: TQRDBText;
2230   QRDBText6: TQRDBText;
2231   QRDBText7: TQRDBText;
2232   QRDBText10: TQRDBText;
2233   QRDBText11: TQRDBText;
2234   btnClose: TButton;
2235   btnPrint: TButton;
2236   QRLabel8: TQRLabel;
2237   QRDBText8: TQRDBText;
2238   procedure FormCreate(Sender: TObject);
2239   procedure QRDBText1Print(sender: TObject; var Value: String);
2240   procedure btnCloseClick(Sender: TObject);
2241   procedure QuickRep1Preview(Sender: TObject);
2242   procedure btnPrintClick(Sender: TObject);
2243   procedure QRLabel12Print(sender: TObject; var Value: String);
2244   private
2245     { Private declarations }
2246   public
2247     { Public declarations }
2248   end;
2249
2250   var
2251     IncRepForm: TIncRepForm;
2252
2253   implementation
2254
2255   Uses
2256     frmIncomes;
2257
2258   {$R *.dfm}
2259
2260   procedure TIncRepForm.FormCreate(Sender: TObject);
2261   begin
2262     WindowState := wsMaximized;
2263     QuickRep1.SendToBack;
2264     QuickRep1.PreviewModeless;
2265   end;
2266
2267   procedure TIncRepForm.QRDBText1Print(sender: TObject; var
2268     Value: String);
2269   begin
2270     Value := ' ' + Value;
2271   end;
2272
2273   procedure TIncRepForm.btnCloseClick(Sender: TObject);
2274   begin
2275     Close;
2276   end;
2277
2278   procedure TIncRepForm.QuickRep1Preview(Sender: TObject);
2279   begin
2280     QRPreview1.QRPrinter := TQRPrinter(Sender);
2281   end;
2282
2283   procedure TIncRepForm.btnPrintClick(Sender: TObject);
2284   begin
2285     QuickRep1.PrinterSetup;
2286     if QuickRep1.Tag = 0 then
2287       QuickRep1.Print;
2288   end;
2289

```



```

2290
2291 procedure TIncRepForm.QRLabel12Print(sender: TObject; var Value:
2292 String);
2293 begin
2294   Value := Format(Value, [FormatDateTime('hh:mm dd.MM.yyyy ?',
2295 Now)]);
2296 end;
2297
2298 end.
2299
2300
2301 unit repDifferences;
2302
2303 interface
2304
2305 uses
2306   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
2307   Forms,
2308   Dialogs, QuickRpt, QRCtrls, QRPrntr, ExtCtrls, StdCtrls, DB,
2309   DBTables;
2310
2311 type
2312   TDifRepForm = class(TForm)
2313     Panel1: TPanel;
2314     QRPreview1: TQRPreview;
2315     Panel2: TPanel;
2316     QuickRep1: TQuickRep;
2317     QRBand1: TQRBand;
2318     QRBand2: TQRBand;
2319     QRBand3: TQRBand;
2320     QRLabel4: TQRLabel;
2321     QRLabel5: TQRLabel;
2322     QRLabel7: TQRLabel;
2323     QRLabel10: TQRLabel;
2324     QRLabel11: TQRLabel;
2325     QRDBText4: TQRDBText;
2326     QRDBText5: TQRDBText;
2327     QRDBText7: TQRDBText;
2328     QRDBText10: TQRDBText;
2329     QRDBText11: TQRDBText;
2330     btnClose: TButton;
2331     btnPrint: TButton;
2332     QRLabel8: TQRLabel;
2333     QRDBText1: TQRDBText;
2334     DifferencesQuery: TQuery;
2335     DifferencesSource: TDataSource;
2336     DifferencesQueryQNUMBER: TFloatField;
2337     DifferencesQueryQINCPERBEG: TDateField;
2338     DifferencesQueryQINQSUMM: TFloatField;
2339     DifferencesQueryQREPSUMM: TFloatField;
2340     DifferencesQueryPCODE: TStringField;
2341     DifferencesQueryPNAME: TStringField;
2342     QRLabel1: TQRLabel;
2343     QRLabel2: TQRLabel;
2344     QRLabel3: TQRLabel;
2345     QRLabel6: TQRLabel;
2346     QRLabel15: TQRLabel;
2347     QRLabel16: TQRLabel;
2348     procedure FormCreate(Sender: TObject);
2349     procedure FormClose(Sender: TObject; var Action: TCloseAction);
2350     procedure QRDBText1Print(sender: TObject; var Value: String);
2351     procedure btnCloseClick(Sender: TObject);

```

```
2352 procedure QuickRep1Preview(Sender: TObject);
2353 procedure btnPrintClick(Sender: TObject);
2354 private
2355   { Private declarations }
2356 public
2357   { Public declarations }
2358 end;
2359
2360 var
2361   DifRepForm: TDifRepForm;
2362
2363 implementation
2364
2365 Uses
2366   frmKontrDoh;
2367
2368 {$R *.dfm}
2369
2370 procedure TDifRepForm.FormCreate(Sender: TObject);
2371 begin
2372   DifferencesQuery.DatabaseName := KontrDohForm.MainDir +
2373   'DBF';
2374   DifferencesQuery.Active := true;
2375   WindowState := wsMaximized;
2376   QuickRep1.SendToBack;
2377   QuickRep1.PreviewModeless;
2378 end;
2379
2380 procedure TDifRepForm.FormClose(Sender: TObject; var Action:
2381   TCloseAction);
2382 begin
2383   DifferencesQuery.Active := false;
2384 end;
2385
2386 procedure TDifRepForm.QRDBText1Print(sender: TObject; var
2387   Value: String);
2388 begin
2389   Value := ' ' + Value;
2390 end;
2391
2392 procedure TDifRepForm.btnCloseClick(Sender: TObject);
2393 begin
2394   Close;
2395 end;
2396
2397 procedure TDifRepForm.QuickRep1Preview(Sender: TObject);
2398 begin
2399   QRPreview1.QRPrinter := TQRPrinter(Sender);
2400 end;
2401
2402 procedure TDifRepForm.btnPrintClick(Sender: TObject); //Печать
2403 begin
2404   QuickRep1.PrinterSetup;
2405   if QuickRep1.Tag = 0 then
2406     QuickRep1.Print;
2407 end;
2408
2409
2410 end.
2411
2412
2413
```

```

2414 unit repQueries;
2415
2416 interface
2417
2418 uses
2419   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
2420   Forms,
2421   Dialogs, QuickRpt, QRCtrls, QRPrntr, ExtCtrls, StdCtrls;
2422
2423 type
2424   TQueriesRepForm = class(TForm)
2425     Panel1: TPanel;
2426     QRPreview1: TQRPreview;
2427     Panel2: TPanel;
2428     QuickRep1: TQuickRep;
2429     QRBand1: TQRBand;
2430     QRBand2: TQRBand;
2431     QRBand3: TQRBand;
2432     QRLabel1: TQRLabel;
2433     QRLabel2: TQRLabel;
2434     QRLabel3: TQRLabel;
2435     QRLabel4: TQRLabel;
2436     QRLabel5: TQRLabel;
2437     QRLabel7: TQRLabel;
2438     QRLabel10: TQRLabel;
2439     QRLabel11: TQRLabel;
2440     QRLabel12: TQRLabel;
2441     QRDBText1: TQRDBText;
2442     QRDBText2: TQRDBText;
2443     QRDBText3: TQRDBText;
2444     QRDBText4: TQRDBText;
2445     QRDBText5: TQRDBText;
2446     QRDBText7: TQRDBText;
2447     QRDBText10: TQRDBText;
2448     QRDBText11: TQRDBText;
2449     btnClose: TButton;
2450     btnPrint: TButton;
2451     QRLabel6: TQRLabel;
2452     QRLabel8: TQRLabel;
2453     procedure FormCreate(Sender: TObject);
2454     procedure QRDBText1Print(sender: TObject; var Value: String);
2455     procedure btnCloseClick(Sender: TObject);
2456     procedure QuickRep1Preview(Sender: TObject);
2457     procedure btnPrintClick(Sender: TObject);
2458     procedure QRLabel12Print(sender: TObject; var Value: String);
2459   private
2460     { Private declarations }
2461   public
2462     { Public declarations }
2463   end;
2464
2465 var
2466   QueriesRepForm: TQueriesRepForm;
2467
2468 implementation
2469
2470 Uses
2471   frmQueries;
2472
2473 {$R *.dfm}
2474
2475 procedure TQueriesRepForm.FormCreate(Sender: TObject);

```

```

2476 begin
2477   WindowState := wsMaximized;
2478   QuickRep1.SendToBack;
2479   QuickRep1.PreviewModeless;
2480 end;
2481
2482 procedure TQueriesRepForm.QRDBText1Print(sender: TObject; var
2483 Value: String);
2484 begin
2485   Value := ' ' + Value;
2486 end;
2487
2488 procedure TQueriesRepForm.btnCloseClick(Sender: TObject);
2489 begin
2490   Close;
2491 end;
2492
2493 procedure TQueriesRepForm.QuickRep1Preview(Sender: TObject);
2494 begin
2495   QRPreview1.QRPrinter := TQRPrinter(Sender);
2496 end;
2497
2498 procedure TQueriesRepForm.btnPrintClick(Sender: TObject);
2499 begin
2500   QuickRep1.PrinterSetup;
2501   if QuickRep1.Tag = 0 then
2502     QuickRep1.Print;
2503 end;
2504
2505
2506 procedure TQueriesRepForm.QRLabel12Print(sender: TObject; var
2507 Value: String);
2508 begin
2509   Value := Format(Value, [IntToStr
2510 (QueriesForm.QueriesQuery.RecordCount)]);
2511 end;
2512
2513 end.
2514
2515
2516
2517
2218 unit repSubsidy;
2219
2220 interface
2221
2222 uses
2223   Windows, Messages, SysUtils, Variants, Classes,
2224   Graphics, Controls, Forms,
2225   Dialogs, QuickRpt, QRCtrls, QRPrntr, ExtCtrls,
2226   StdCtrls, DB, DBTables;
2227
2228 type
2229   TSubsRepForm = class(TForm)
2230     Panel1: TPanel;
2231     QRPreview1: TQRPreview;
2232     Panel2: TPanel;
2233     QuickRep1: TQuickRep;
2234     QRBand1: TQRBand;
2235     QRBand3: TQRBand;
2236     QRDBText4: TQRDBText;
2237     QRDBText5: TQRDBText;

```

```

2238   QRDBText7: TQRDBText;
2239   btnClose: TButton;
2240   btnPrint: TButton;
2241   QRDBText1: TQRDBText;
2242   QRLabel6: TQRLabel;
2243   QRLabel15: TQRLabel;
2244   QRLabel16: TQRLabel;
2245   QRLabel9: TQRLabel;
2246   QRLabel12: TQRLabel;
2247   QRLabel13: TQRLabel;
2248   PaymentsQuery: TQuery;
2249   PaymentsSource: TDataSource;
2250   PaymentsQueryONAME: TStringField;
2251   PaymentsQueryOACCOUNT: TStringField;
2252   PaymentsQuerySPAY: TFloatField;
2253   PaymentsQueryTotalPay: TFloatField;
2254   PaymentsQuerySSUBSIDY: TFloatField;
2255   QRBand2: TQRBand;
2256   QRLabel4: TQRLabel;
2257   QRLabel5: TQRLabel;
2258   QRLabel7: TQRLabel;
2259   QRLabel8: TQRLabel;
2260   QRLabel19: TQRLabel;
2261   QRLabel20: TQRLabel;
2262   SubsidyQuery: TQuery;
2263   SubsidyQueryQNUMBER: TFloatField;
2264   SubsidyQueryQPERSON: TFloatField;
2265   SubsidyQueryPNAME: TStringField;
2266   SubsidyQueryPADDRESS: TStringField;
2267   SubsidyQueryQINCPERBEG: TDateField;
2268   SubsidySource: TDataSource;
2269   SumQuery: TQuery;
2270   QRBand4: TQRBand;
2271   QRExpr1: TQRExpr;
2272   QRExpr2: TQRExpr;
2273   QRBand5: TQRBand;
2274   QRLabel1: TQRLabel;
2275   QRLabel2: TQRLabel;
2276   QRLabel3: TQRLabel;
2277   QRLabel11: TQRLabel;
2278   QRLabel10: TQRLabel;
2279   QRLabel14: TQRLabel;
2280   QRLabel17: TQRLabel;
2281   QRLabel18: TQRLabel;
2282   QRLabel21: TQRLabel;
2283   SumQueryPAYSUM: TFloatField;
2284   SumQuerySUBSSUM: TFloatField;
2285   procedure FormClose(Sender: TObject; var
2286 Action: TCloseAction);
2287   procedure QRDBText1Print(sender: TObject; var
2288 Value: String);
2289   procedure btnCloseClick(Sender: TObject);
2290   procedure QuickRep1Preview(Sender: TObject);
2291   procedure btnPrintClick(Sender: TObject);
2292   procedure QRLabel6Print(sender: TObject; var
2293 Value: String);
2294   procedure PaymentsQueryCalcFields(DataSet:
2295 TDataSet);
2296   procedure QRLabel16Print(sender: TObject; var
2297 Value: String);
2298   procedure QRLabel9Print(sender: TObject; var
2299 Value: String);

```

```

2300 procedure QRLabel13Print(sender: TObject; var
2301 Value: String);
2302 procedure QRLabel21Print(sender: TObject; var
2303 Value: String);
2304 procedure QRLabel18Print(sender: TObject; var
2305 Value: String);
2306 private
2307 { Private declarations }
2308 public
2309 { Public declarations }
2310 constructor Create(AOwner: TComponent;
2311 aQueryId: integer); reintroduce; virtual;
2312 end;
2313
2314 var
2315 SubsRepForm: TSubsRepForm;
2316
2317 implementation
2318
2319 Uses
2320 frmKontrDoh;
2321
2322 {$R *.dfm}
2323
2324 constructor TSubsRepForm.Create(AOwner:
2325 TComponent; aQueryId: integer);
2326 begin
2327 inherited Create(AOwner);
2328 WindowState := wsMaximized;
2329 QuickRep1.SendToBack;
2330
2331 SubsidyQuery.DatabaseName :=
2332 KontrDohForm.MainDir + '\DBF';
2333 PaymentsQuery.DatabaseName :=
2334 KontrDohForm.MainDir + '\DBF';
2335 SumQuery.DatabaseName :=
2336 KontrDohForm.MainDir + '\DBF';
2337 SubsidyQuery.Params[0].Value := aQueryId;
2338 try
2339 Screen.Cursor := crSQLWait;
2340 SubsidyQuery.Active := true;
2341 PaymentsQuery.Params[0].Value :=
2342 SubsidyQuery.QPERSON.AsInteger;
2343 PaymentsQuery.Active := true;
2344 SumQuery.Params[0].Value :=
2345 SubsidyQuery.QPERSON.AsInteger;
2346 SumQuery.Active := true;
2347 Screen.Cursor := crDefault;
2348 finally
2349 end;
2350 QuickRep1.PreviewModeless;
2351 end;
2352
2353 procedure TSubsRepForm.FormClose(Sender:
2354 TObject; var Action: TCloseAction);
2355 begin
2356 SubsidyQuery.Active := false;
2357 PaymentsQuery.Active := false;
2358 SumQuery.Active := false;
2359 end;
2360
2361 procedure TSubsRepForm.QRDBText1Print(sender:

```

```

2362 TObject; var Value: String);
2363 begin
2364   Value := ' ' + Value;
2365 end;
2366
2367 procedure TSubsRepForm.btnCloseClick(Sender:
2368 TObject);
2369 begin
2370   Close;
2371 end;
2372
2373 procedure TSubsRepForm.QuickRep1Preview
2374 (Sender: TObject);
2375 begin
2376   QRPreview1.QRPrinter := TQRPrinter(Sender);
2377 end;
2378
2379 procedure TSubsRepForm.btnPrintClick(Sender:
2380 TObject); //Печать
2381 begin
2382   QuickRep1.PrinterSetup;
2383   if QuickRep1.Tag = 0 then
2384     QuickRep1.Print;
2385 end;
2386
2387 procedure TSubsRepForm.QRLabel6Print(sender:
2388 TObject; var Value: String);
2389 begin
2390   Value := Format(Value,
2391 [SubsidyQueryQNUMBER.AsString]);
2392 end;
2393
2394 procedure
2395 TSubsRepForm.PaymentsQueryCalcFields(DataSet:
2396 TDataSet);
2397 begin
2398   PaymentsQueryTotalPay.Value :=
2399 PaymentsQuerySPAY.AsFloat -
2400 PaymentsQuerySSUBSIDY.AsFloat;
2401 end;
2402
2403 procedure TSubsRepForm.QRLabel16Print(sender:
2404 TObject; var Value: String);
2405 begin
2406   Value := Format(Value,
2407 [SubsidyQueryPNAME.AsString]);
2408 end;
2409
2410 procedure TSubsRepForm.QRLabel9Print(sender:
2411 TObject; var Value: String);
2412 begin
2413   Value := Format(Value,
2414 [SubsidyQueryPADDRESS.AsString]);
2415 end;
2416
2417 procedure TSubsRepForm.QRLabel13Print(sender:
2418 TObject; var Value: String);
2419 begin
2420   Value := Format(Value,
2421 [SubsidyQueryQINCPERBEG.AsString,
2422   DateTimeToStr
2423 (SubsidyQueryQINCPERBEG.AsDateTime +

```

```
2424 100));
2425 end;
2426
2427 procedure TSubsRepForm.QRLabel21Print(sender:
2428 TObject; var Value: String);
2429 begin
2430 Value := Format(Value, [FloatToStr
2431 (SumQueryPAYSUM.AsFloat -
2432 SumQuerySUBSSUM.AsFloat)]);
2433 end;
2434
2435 procedure TSubsRepForm.QRLabel18Print(sender:
2436 TObject; var Value: String);
2437 begin
2438 Value := Format(Value, [FormatDateTime
2439 ('dd.MM.yyyy ?.', Int(Now))]);
2440 end;
2441
2442
2443 end.
```


ДОДАТОК В Презентація

**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ імені В. ДАЛЯ
Кафедра комп'ютерних наук та інженерії**

ПРОГРАМНІ ЗАСОБИ ОБРОБКИ ІНФОРМАЦІЇ ПРО СУБСИДІЇ

Освітньо-кваліфікаційний рівень "бакалавр"
Спеціальність 122 – "комп'ютерні науки"

**Здобувач вищої
освіти:**

Маланчак Данило Олександрович

Керівник

дипломної роботи:

Шумова Лариса Олександрівна

АКТУАЛЬНІСТЬ ТЕМИ

Програма житлових субсидій стала однією з найважливіших програм соціального захисту.

Вона є першою в Україні адресною програмою соціального захисту, спрямована найменш захищеним верствам населення, яка зняла соціальну напругу в суспільстві в умовах підвищення цін і тарифів на житло, комунальні послуги та енергоносії.

Актуальною є задача реалізації автоматизованої обробки документів, які здаються громадянами до відділів субсидій, а також формування довідкових і звітних документів.

МЕТА

— створення інформаційно-комп'ютерної системи для автоматизації засобів обробки інформації про субсидії.

Досягнення поставленої мети дозволить уникнути дублювання даних, прискорить пошук потрібної інформації, підвищить ефективність обробки інформації про субсидії.

Постановка задачі

- Аналіз предметної області
- Аналіз системи управління і організації баз даних
- Проектування інформаційної системи контролю та обміну даними по субсидіях
- Розробка підсистеми "Контроль доходів", що призначена для автоматизації контролю та обміну інформацією з податковою інспекцією

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

При спрощеному розгляді схеми системи баз даних можна виділити 4 основних компоненти системи:

- Дані;
- Апаратне забезпечення (накопичувачі для зберігання інформації разом з приєднаними пристроями введення-виведення, контролерами пристроїв, каналами введення-виведення);
- Програмне забезпечення, (система управління базами даних, засоби розробки додатків, засоби проектування, генератори звітів і т.д.);
- Користувачі.

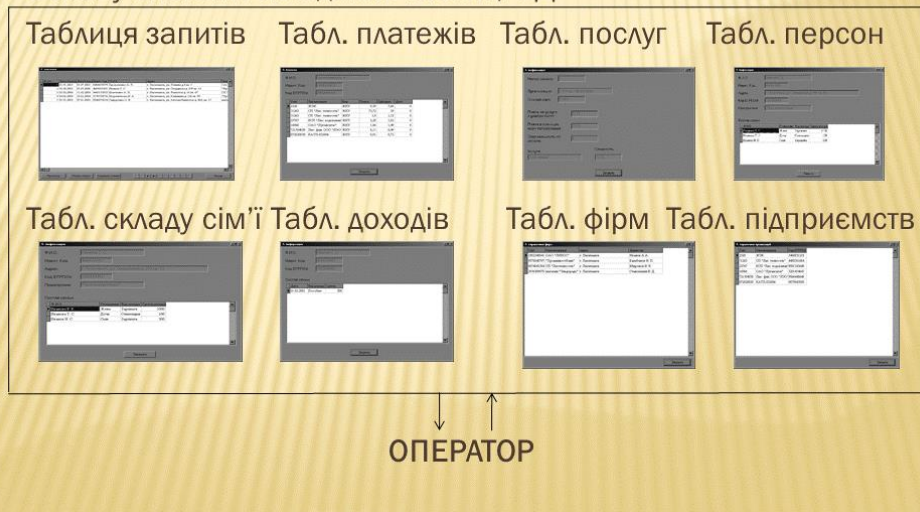
Особливості роботи відділу субсидій

Фактично відділом субсидій здійснюються такі види діяльності:

- ✘ Прийом заяв на надання субсидій від громадян;
- ✘ Занесення даних відділом субсидій;
- ✘ Обробка отриманої інформації;
- ✘ Призначення субсидії або відмову в ній;
- ✘ Внесення отриманої інформації в комп'ютер.

АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

Основу системи складають таблиці БД



Опис структури таблиць підсистеми

Структура таблиці запитів

Найменування поля	Тип даних поля
код запиту	числовий
номер запиту	строковий
код персони	числовий
дата початку періоду доходу	дата
запит	логічний
відповідь	логічний
сума запиту	числовий
сума відповіді	числовий
дата запиту	дата
дата відповіді	дата

Структура таблиці доходів

Найменування поля	Тип даних поля
код доходу	числовий
дата початку	дата
дата кінця	дата
код персони	числовий
тип доходу	дата
сума доходу	строковий

Структура таблиці персон

Найменування поля	Тип даних поля
код персони	числовий
ідентифікаційний код	строковий
П.І.Б.	строковий
адреса	строковий
код ЄДРПОУ	строковий
код фірми	числовий

Структура таблиці платежів

Найменування поля	Тип даних поля
код платежу	числовий
код персони	числовий
код організації	числовий
код послуги	числовий
сума боргу	числовий
дата платежу	дата

Опис структури таблиць підсистеми

Структура таблиці послуг

Найменування поля	Тип даних поля
код послуги	числовий
скорочене найменування	строковий
найменування	строковий
сума платежу	числовий
сума субсидії	числовий

Структура таблиці організацій

Найменування поля	Тип даних поля
код організації	числовий
особистий рахунок	строковий
найменування	строковий
код ЄДРПОУ	строковий

Структура таблиці фірм

Найменування поля	Тип даних поля
код фірми	числовий
особистий рахунок	строковий
найменування	строковий
адреса	строковий
директор	строковий

Структура таблиці складу сім'ї

Найменування поля	Тип даних поля
код складу сім'ї	числовий
код персони	числовий
найменування	строковий
вид відносин	строковий
вид доходу	строковий
сума доходу	числовий

Опис програми

Основна програма включає в себе функції відображення таблиць БД, створення і друк звітів. Функція вибору, вироблених оператором дій покладається на форми.

У кодї програми можна виділити основний набір підпрограм:

- ✘ підпрограма відкриття таблиці;
- ✘ підпрограма створення файлу TTL;
- ✘ підпрограма формування звіту по субсидіях;
- ✘ підпрограма відкриття довідників.

ВИСНОВКИ

У проєкті поставлені і вирішені такі основні завдання:

- ✘ Проаналізовано існуючі програмні засоби аналогічного призначення, зроблені відповідні висновки про області застосування, а також про необхідність і доцільність розробки нового програмного засобу.
- ✘ Складено технічне завдання на розробку програмного забезпечення підсистеми "Контроль доходів", що включає в себе основні технічні вимоги до розробляється програмного забезпечення.
- ✘ Спроектвана БД.
- ✘ Розроблено програмне забезпечення підсистеми "Контроль доходів", що призначена для автоматизації контролю та обміну інформацією з податковою інспекцією.
- ✘ Приведені рекомендації щодо організації робочого місця, а також наведена інформація щодо пожежної та електробезпеки.