

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

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

**Комп'ютерна система автоматизованого проектування програмно-  
технічного комплексу**

---

---

Освітньо-кваліфікаційний рівень “Магістр”  
Спеціальність 123 “Комп'ютерна інженерія” (освітня програма - “Комп'ютерні системи і мережі”)

Науковий керівник роботи:

\_\_\_\_\_

(підпис)

**В.М.Барбарук**

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

**Я.О.Критська**

\_\_\_\_\_

(ініціали, прізвище)

Студент:

\_\_\_\_\_

(підпис)

**Д.В. Вовченко**

\_\_\_\_\_

(ініціали, прізвище)

Група:

**ІТІ-163м**

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 магістр  
Напрямок підготовки \_\_\_\_\_  
(шифр і назва)  
Спеціальність 123 "Комп'ютерна інженерія" (освітня програма - "Комп'ютерні системи і мережі")  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
І.С. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

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

Вовченку Дмитру Вікторовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система автоматизованого проектування програмно-технічного комплексу

керівник проекту (роботи) Барбарук Віктор Миколайович, к.т.н., доц.  
(прізвище, м.я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «18» 10 2018 р. № 208/48

2. Строк подання студентом роботи 21.01.2018

3. Вихідні дані до роботи Матеріали науково-дослідної практики, призначення і загальна характеристика МСКУ-4

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд існуючих методів автоматизованого проектування ПТК АСУ ТП; вибір методу проектування системи компонування підсистем нижнього рівня технічного комплексу МСКУ-4.1; розробка автоматизованої системи компонування підсистем нижнього рівня комплексу МСКУ-4.1; 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	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 18.10.2017

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Аналіз літературних джерел і обґрунтування актуальності	10.09.2017-15.09.2017	
2	Розробка технічного завдання	16.09.2017-22.09.2017	
3	Аналіз існуючих методів автоматизованого проектування ПТК АСУ ТП	23.09.2017-25.09.2017	
4	Проектування та розробка автоматизованої системи компонування підсистем нижнього рівня комплексу МСКУ-4.1	26.09.2017-06.10.2017	
5	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	07.10.2017-25.10.2017	
6	Оформлення пояснювальної записки, автореферату та презентації	26.10.2017-13.01.2018	

Студент

\_\_\_\_\_ ( підпис )

Вовченко Д.В.

\_\_\_\_\_ (прізвище та ініціали)

Науковий керівник

\_\_\_\_\_ ( підпис )

Барбарук В.М.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Вовченко Д. В. Комп'ютерна система автоматизованого проектування програмно-технічного комплексу

Метою роботи є дослідження існуючих методів автоматизованого проектування ПТК АСУ ТП. Систему автоматизованого компонування підсистем нижнього рівня технічного комплексу на базі МСКУ-4.1 реалізовано на основі даних обраного методу автоматизованого проектування ПТК АСУ ТП. Основним завданням розробленого програмного забезпечення є здійснення коректного проектування шаф на базі МСКУ-4.1.

**Ключові слова:** програмно-технічний комплекс, діаграми, структурний підхід, автоматизоване проектування.

## THE ABSTRACT

Vovchenko D.V. Computer system of automated design of software and hardware complex

The work's purpose is to study existing methods of computer-aided design programmatic and technical complexes of automated control systems of technological processes and to develop the system of automated assembly of the subsystem of the lower level of computer system managing MCS-4.1. The main task of the developed software is the correct design cupboards for MCS-4.1.

**Keywords:** software and hardware complex, diagrams, structural approach, automated design.

## АННОТАЦИЯ

Вовченко Д. В. Компьютерная система автоматизированного проектирования программно-технического комплекса

Целью работы является исследование существующих методов автоматизированного проектирования ПТК АСУ ТП. Система автоматизированного компоновки подсистем нижнего уровня технического комплекса на базе МСКУ-4.1 реализовано на основе данных выбранного метода автоматизированного проектирования ПТК АСУ ТП. Основной задачей разработанного программного обеспечения является осуществление корректного проектирования шкафов на базе МСКУ-4.1.

**Ключевые слова:** программно-технический комплекс, диаграммы, структурный подход, автоматизированное проектирование.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	6
ВСТУП .....	7
1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ПТК АСУ ТП .....	9
1.1 Роль програмно-технічних комплексів в АСУ ТП.....	9
1.2 Системний підхід до проектування .....	11
1.3 Структурний підхід до проектування .....	15
1.3.1 Методологія SADT.....	15
1.3.2 Діаграми потоків даних DFD .....	17
1.3.3 Діаграми «сутність-зв'язок» ERD .....	20
1.3.4 Методологія IDEF3 .....	23
1.4 Об'єктно-орієнтований підхід до проектування.....	29
1.4.1 Уніфікована мова моделювання UML.....	29
1.4.2 Діаграми варіантів використання .....	31
1.4.3 Діаграми класів .....	35
1.4.4 Діаграми станів .....	39
1.4.5 Діаграми діяльності.....	41
1.4.6 Діаграми послідовності.....	42
1.4.7 Діаграми компонентів і розгортання.....	44
1.5 Особливості МСКУ-4.1 .....	46
1.5.1 Призначення і загальна характеристика МСКУ-4 .....	46
1.5.2 Архітектура технічних засобів МСКУ-4.1 .....	47
1.5.3 Організація каналів зв'язку з об'єктом.....	52
1.6 Постановка завдання .....	53
1.6.1 Опис предметної області.....	53
1.6.2 Технічне завдання на розробку автоматизованої системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1 .....	54
2 ВИБІР МЕТОДУ ПРОЕКТУВАННЯ СИСТЕМИ КОМПОНУВАННЯ ПІДСИСТЕМ НИЖНЬОГО РІВНЯ ТЕХНІЧНОЇ КОМПЛЕКСУ МСКУ-4.1 .....	56
2.1 Подібності структурного і об'єктно-орієнтованого підходів до проектування.....	56
2.2 Порівняльний аналіз структурного і об'єктно-орієнтованого підходів до проектування .....	59
2.3 Обґрунтування вибору методу проектування системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1 .....	62
3 РОЗРОБКА АВТОМАТИЗОВАНА СИСТЕМА КОМПОНУВАННЯ ПІДСИСТЕМ НИЖНЬОГО РІВНЯ КОМПЛЕКСУ МСКУ-4.1.....	64

3.1 Проектування автоматизованої системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1 .....	64
3.2 Створення бази даних.....	69
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	75
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об'єкту, що мають вплив на персонал .....	75
4.2 Заходи щодо техніки безпеки.....	76
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці .....	79
4.4 Рекомендації по пожежній безпеці .....	82
4.5 Охорона навколишнього природного середовища.....	85
4.5.1 Загальні дані з охорони навколишнього природного середовища.....	85
4.5.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі.....	85
ВИСНОВКИ .....	87
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	88
ДОДАТОК А. Запити в MS Access.....	92
ДОДАТОК Б. Комп'ютерна презентація .....	95

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АЕС - атомна електростанція;  
АСУ ТП - автоматизована система управління технологічними процесами;  
БД - база даних;  
ІДС - інженерно-діагностична станція;  
ІРП - інтерфейс радіальний послідовний;  
ІКС - інформаційна керуюча система;  
КМП - контролер мікропроцесорний КМП-30;  
КМ - каркас монтажний;  
Ксапи - комплекс засобів автоматизації проектування;  
МКО-4 - модуль контролю обладнання МКО-4;  
МСКУ - мікропроцесорні керуючі обчислювальні комплекси;  
МЗО - модуль зв'язку з об'єктом;  
ПКр - панель кросова;  
ПО - програмне забезпечення;  
ПСП - панель сполучна;  
ПТК - програмно-технічний комплекс;  
ПЕОМ - персональна електронна обчислювальна машина;  
САПР - система автоматизованого проектування;  
СУБД - система управління базами даних;  
СУЗ - система управління і захисту;  
ТЗА - технічний засіб автоматизації;  
ЕОМ - електронна обчислювальна машина.

## ВСТУП

У наш час стався величезний стрибок у розвитку техніки, який ознаменувався значним поширенням і впровадженням великих і малих автоматизованих систем управління технологічними процесами (АСУ ТП). Ці системи призначені для управління технологічними процесами великих і особливо небезпечних промислових об'єктів типу атомних електростанцій, хімічних і металургійних підприємств, підприємств по переробці нафти і газу і їм подібних. Саме тому проблема забезпечення надійності та пожежної безпеки таких об'єктів є однією з найважливіших задач теперішнього часу і набуває великого значення.

Автоматизація нових технологічних процесів вимагає розробки специфічного обладнання. Базою для побудови АСУ ТП є програмно-технічні комплекси (ПТК).

Надійність АСУ ТП визначається надійністю її апаратури і програмного забезпечення, тобто Надійністю ПТК, який керує виконанням технологічних процесів. Часто від того, наскільки вірно, надійно і точно АСУ ТП управляє технологічним процесом, залежить безпека оточуючих. Для досягнення цієї мети необхідно, щоб всі названі складові - елементи, апаратура і програмне забезпечення - мали необхідну надійність.

Однією з проблем створення ПТК АСУ ТП є оперативне і коректне проектування підсистеми нижнього рівня, обсяг якої становить 90-95% усього обладнання ПТК.

Підсистеми нижнього рівня ПТК виробництва СНВО «Імпульс» являють собою сукупність промислових контролерів типу МСКУ-4.1, що здійснюють прийом / видачу різних типів сигналів, кількість яких може досягати десятки тисяч шт. МСКУ-4.1 є складною системою, яка складається з типових модулів зв'язку з об'єктом (МСО), призначених для введення / виведення дискретних і аналогових сигналів; панелей кросових, призначених для підключення кабелів від об'єктів; панелей сполучних, призначених для зв'язку кросових панелей з МСО; каркасів монтажних, призначених для установки контролера мікропроцесорного і МСО, і іншого устаткування.

Завдання швидкого і коректного проектування ПТК, в тому числі і підсистем нижнього рівня, може бути вирішена застосуванням системи автоматизованого проектування (САПР). Так як існуючі САПР не враховують особливості обладнання фірми, є дорогими і складно адаптуються, розробка системи автоматизованого проектування підсистем нижнього рівня ПТК є актуальним завданням.



**Об'єкт дослідження:** мікропроцесорний керуючий обчислювальний комплекс МСКУ-4.1

**Предмет дослідження:** методи автоматизованого проектування ПТК АСУ ТП.

**Мета роботи:** дослідження існуючих методів автоматизованого проектування ПТК АСУ ТП. Розробка системи автоматизованого компонування підсистем нижнього рівня технічного комплексу на базі управляючого обчислювального комплексу МСКУ-4.1.

При виконанні завдання **необхідно розглянути** наступні питання:

- дослідження структурного та об'єктно-орієнтованого методів автоматизованого проектування, що входять до складу більш загального системного підходу;
- проаналізовані їх достоїнства та недоліки;
- розглянуті особливості управляючого обчислювального комплексу МСКУ-4.1.

**Основні результати магістерської роботи доповідались** на Міжнародній науково-практичній конференції «Майбутній науковець – 2017», та на Всеукраїнській науково-практичній конференції «Електронні апарати та системи. Проблеми створення. Перспективи розвитку».

**Магістерська робота складається** зі вступу, 4 розділів, висновків, переліку джерел посилань, додатку. Загальний обсяг роботи становить 102 сторінок, 6 таблиць, 37 рисунків.

# 1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ПТК АСУ ТП

## 1.1 Роль програмно-технічних комплексів в АСУ ТП

Розвиток сучасних високих технологій супроводжується впровадженням нових автоматизованих систем управління технологічними процесами.

Термінологічний стандарт визначає АСУ ТП як систему, що складається з персоналу і комплексу засобів автоматизації його діяльності, яка реалізує інформаційну технологію виконання встановлених функцій управління технологічним процесом [1].

Тим часом на практиці під АСУ ТП розуміють за замовчуванням лише одну її частину - комплекс засобів автоматизації. Під комплексом засобів автоматизації мається на увазі що поставляється сукупність взаємоузгоджених комплексів технічних і програмних засобів (виробів), розроблена і виготовлена як продукція виробничо-технічного призначення.

Функціями АСУ ТП на верхніх рівнях є збір і обробка даних про стан обладнання і протіканні виробничих процесів для прийняття рішень щодо завантаження верстатів, щодо виконання технологічних маршрутів [2].

Складовими частинами АСУ ТП можуть бути окремі системи автоматичного управління та автоматизовані пристрої, пов'язані в єдиний комплекс. Як правило, АСУ ТП має єдину систему операторського управління технологічним процесом у вигляді одного або декількох пультів управління, засоби обробки та архівування інформації про хід процесу, типові елементи автоматики: датчики, пристрої управління, виконавчі пристрої. Для інформаційного зв'язку всіх підсистем використовуються промислові мережі.

Автоматизація промислового виробництва одна з найважливіших завдань нашого часу. Автоматизація нових технологічних процесів вимагає розробки специфічного обладнання. Технічною базою для побудови АСУ ТП є програмно-технічні комплекси (ПТК) [3].

Програмно-технічний комплекс називають сукупність технічних засобів, що поставляються комплектно з програмним забезпеченням, необхідним сервісним обладнанням та експлуатаційної документації, і призначених для вирішення широкого кола завдань на підприємствах енергетики, металургії, хімії, транспорту, машинобудування. Згадки про програмно-технічних комплексах зустрічалися в літературі і нормативної документації ще на початку 1980-х років, але лише після розробки та

впровадження норм і правил це поняття стало широко застосовуватися у вітчизняній практиці, особливо при створенні і модернізації інформаційних і керуючих систем (ІКС) в атомній енергетиці [1].

Зазвичай до складу кожної ІКС входить один ПТК, який є головним компонентом системи, що беруть участь в реалізації всіх її основних і додаткових функцій. Рідше ці функції розподіляються між декількома ПТК, що входять до складу однієї системи, наприклад, деякі додаткові функції, такі як технічне діагностування, можуть виконуватися спеціально призначеними для цього програмно-технічними комплексами [4]. Для підвищення надійності передбачаються кілька каналів в одному ПТК і (або) декілька ПТК в одній системі, які одночасно і незалежно беруть участь в реалізації основних функцій, взаємно резервуючи один одного.

Кожен ПТК виконаний у вигляді сукупності експлуатаційно-автономних складових частин, що з'єднуються на місці експлуатації електричними або іншими (наприклад, волоконно-оптичними) сполучними лініями, і під керівництвом свого («вбудованого») програмного забезпечення в загальному випадку здійснює:

- прийом і перетворення в цифрову форму сигналів від периферійних пристроїв;
- прийом повідомлень від інших ПТК по каналах передачі даних;
- обробку прийнятої інформації і вироблення команд управління;
- формування і видачу керуючих сигналів на виконавчі пристрої;
- діагностування стану власного і сполученого устаткування;
- архівування, відображення, реєстрацію поточної і ретроспективної інформації;
- підготовку і видачу повідомлень в канали передачі даних для інших ПТК.

Кожен ПТК є виробом, тобто одиницею продукції виробничо-технічного призначення, налагодженої, перевіреної в заводських умовах і поставляється споживачеві комплектно з програмним забезпеченням, необхідним сервісним обладнанням та експлуатаційної документації [5].

Високий ступінь заводської готовності істотно спрощує і прискорює монтаж-налагоджувальні операції перед пуском енергоблоку і гарантує відповідність ПТК всім вимогам, що пред'являються. ПТК компонується на підприємстві-виробнику з покупних електротехнічних виробів та електронних елементів загального призначення, а також вбудованих складальних одиниць (не призначених для експлуатаційно-автономного застосування), які встановлюються в несучі конструкції (оболонки). Комплектація, збірка і монтаж ПТК здійснюються на підставі конструкторської документації. Повна перевірка функціонування ПТК (з імітаторами сполученого устаткування) здійснюються на підприємстві-виробнику, до поставки замовнику; передбачається додаткова перевірка

(передпускові випробування) на місці експлуатації, перед інтеграцією ПТК з іншими частинами ІУС. Гарантії постачальника поширюються на ПТК в цілому, включаючи застосовані в ньому покупні комплектуючі вироби [1].

На рівні управління технологічним обладнанням в АСУТП виконуються запуск, тестування, виключення верстатів, сигналізація про несправності, вироблення керуючих впливів для робочих органів програмно керованого обладнання. АСУ ТП об'єднують різні об'єкти і пристрої, локальні і віддалені, в єдиний комплекс і дозволяють контролювати і програмувати їх роботу, як в цілому, так і окремо. Цим забезпечується максимальна ефективність і безпеку виробництва, можливість оперативної налагодження і переналагодження, строгий облік і планування показників операційної діяльності, оптимізація бізнес-процесів [6].

## **1.2 Системний підхід до проектування**

В останні роки все більший вплив приділяється проблемам методології проектування. Проектуванням будь-якого об'єкта є створення, перетворення і подання до прийнятої форми образу цього ще не існуючого об'єкта.

Проектування, при якому всі проектні рішення або їх частину отримують шляхом взаємодії людини і ЕОМ, називають автоматизованим, на відміну від ручного (без використання ЕОМ) або автоматичного (без участі людини на проміжних етапах) [7].

Проектування автоматизованих систем - це створення графічних, текстових, програмних та інших документів, достатніх для створення і експлуатації проектованої автоматизованої системи і оформлених на паперових та електронних носіях.

Належна до проектування автоматизована система зазвичай не є якоюсь ізольованою, а являє собою частину більш складної системи. Складною системою називають складений об'єкт (систему), що складається з великого числа взаємопов'язаних відповідно до певними причинами і відносинами елементів [8].

Проектне рішення, що стосується розробляється автоматизованої системи, має бути оптимальним для всієї системи в цілому, а не тільки для розроблюваної.

Автоматизоване проектування складних систем виконується за допомогою організаційно-технічної системи, що складається із сукупності комплексу засобів автоматизації проектування (КСАП) і колективу фахівців підрозділів проектної організації, званої системою автоматизованого проектування (САПР).

Основна функція САПР полягає у виконанні автоматизованого проектування на всіх або окремих стадіях проектування об'єктів та їх складових частин [9].

Складовими структурними частинами САПР, жорстко пов'язаними з організаційною структурою проектної організації, є підсистеми, в яких за допомогою спеціалізованих комплексів засобів вирішується функціонально закінчена послідовність завдань САПР.

За призначенням підсистеми поділяються на:

- проектують;
- обслуговуючі.

Проекують підсистеми мають об'єктну орієнтацію і реалізують певний етап (стадію) проектування або групу безпосередньо пов'язаних проектних завдань, тобто виконують проектні процедури.

Обслуговуючі підсистеми мають загальносистемне застосування і забезпечують підтримку функціонування проектуєть підсистем, а також оформлення, передачу і видачу отриманих в них результатів. Сукупність обслуговуючих підсистем часто називають системної середовищем (або оболонкою) САПР [10].

Формування та використання моделей об'єкта проектування в прикладних задачах здійснюється КСАП системи або підсистеми.

Структурними частинами КСАП в процесі його функціонування є програмно-методичні та програмно-технічні комплекси (комплекси засобів), а також компоненти організаційного забезпечення [11].

Структурними частинами комплексів засобів є компоненти наступних видів забезпечення:

- технічного забезпечення, що включає різні апаратні засоби (ЕОМ, периферійні пристрої, мережеве комутаційне обладнання, лінії зв'язку, вимірювальні засоби);
- програмного забезпечення, яку представляють комп'ютерними програмами САПР;
- математичного забезпечення, що об'єднує методи математичного моделювання об'єктів і процесів проектування, математичні моделі об'єктів і процесів проектування, алгоритми вирішення завдань в процесі проектування;
- інформаційного забезпечення, що складається з баз даних (БД), систем управління базами даних (СКБД), а також інших даних, що використовуються при проектуванні (вся сукупність використовуваних при проектуванні даних називається інформаційним фондом САПР, а БД разом з СУБД носить назву банку даних) ;

- методичного забезпечення, що включає різні документи, що характеризують склад, правила відбору і експлуатації засобів автоматизованого проектування;
- лінгвістичного забезпечення, що виражається мовами спілкування між проектувальниками і ЕОМ, мовами програмування і мовами обміну даними між технічними засобами САПР;
- організаційного забезпечення, яку представляють штатними розкладами, посадовими інструкціями та іншими документами, що регламентують роботу проектного підприємства [10].

Компоненти видів забезпечення виконують в комплексах засобів задану функцію і представляють найменший (неподільний) самостійно розробляється (або покупної) елемент САПР (наприклад: програма, інструкція, дисплей і т.д.).

Ефективне функціонування КСАП і взаємодію структурних частин САПР всіх рівнів має досягатися за рахунок орієнтації на стандартні інтерфейси і протоколи зв'язку, що забезпечують взаємодію комплексів засобів.

Основні ідеї та принципи проектування складних систем виражені в системному підході. Системи автоматизованого проектування і управління є одними з найбільш складних сучасних штучних систем. Тому їх проектування і супровід неможливі без застосування системного підходу, який є методологічною основою проектування [8].

Принцип системного підходу полягає в розгляді частин складної системи з урахуванням їх взаємодії. Системний підхід включає в себе виявлення структури системи, типізацію зв'язків, визначення властивостей (атрибутів) системи, аналіз впливу зовнішнього середовища.

Головним завданням системного аналізу є пошук шляхів по перетворенню складного в просте, з розкладання важкозрозумілі завдання на ряд завдань, що мають рішення [12].

По відношенню до проектування складної програмної системи це означає, що її необхідно розділити (декомпозувати) на невеликі підсистеми, кожна з яких можна розробляти незалежно від інших. Це дозволяє при розробці підсистеми будь-якого рівня мати справу тільки з нею, а не з усіма іншими частинами системи.

Інтерпретація і конкретизація системного підходу мають місце в ряді відомих підходів з іншими назвами. Такі структурний і об'єктно-орієнтований підходи [13].

В основу структурного або функціонально-модульного підходу покладено принцип функціональної декомпозиції, при якій структура системи описується в термінах ієрархії її функцій і передачі інформації між окремими функціональними елементами.

В об'єктно-орієнтованому підході до проектування виражений ряд важливих структурних принципів, використовуваних при розробці інформаційних систем і перш за все їх програмного забезпечення. Цей підхід, використовує об'єктну декомпозицію. При цьому структура системи описується в термінах об'єктів і зв'язків між ними, а поведінка системи описується в термінах обміну повідомленнями між об'єктами. Такий підхід скорочує обсяг специфікацій об'єктів і зменшує ймовірність спотворення даних за рахунок обмеження доступу до них [14].

У структурному аналізі використовуються три групи засобів моделювання:

- діаграми, що ілюструють функції, які система повинна виконувати, і зв'язку між ними (функціональне моделювання); найчастіше використовуються DFD (Data Flow Diagrams) - діаграми потоків даних і діаграми SADT (IDEF0);
- діаграми, що моделюють дані і їх взаємозв'язку (інформаційне моделювання); фактично стандартом тут стали ERD (Entity-Relationship Diagrams) - діаграми «сутність - зв'язок»;
- діаграми, що моделюють поведінку системи, залежне від часу (динамічне моделювання); найбільш часто аспекти поведінки системи в часі моделюються за допомогою методології IDEF3 [13].

Об'єктно-орієнтований підхід до проектування виражений в уніфікованому мовою моделювання (Unified Modeling Language - UML), який в даний час фактично є стандартом у цій галузі.

У UML визначений наступний набір діаграм, кожна з яких демонструє певний аспект опису моделі:

1) структурні діаграми:

- - діаграми класів (class diagram);
- - діаграми компонентів (component diagram);
- - діаграми розгортання (deployment diagram);

2) діаграми поведінки:

- - діаграми станів (state diagram);
- - діаграми діяльності (activity diagram);
- - діаграми варіантів використання (use case diagram);
- - діаграми послідовності (sequence diagram) [14].

## 1.3 Структурний підхід до проектування

### 1.3.1 Методологія SADT

Методологія SADT (Structured Analysis and Design Technique - технологія структурного аналізу і проектування) розроблена Дугласом Т. Россом в 1969 році, для моделювання штучних систем середньої складності. SADT - одна з найвідоміших і широко використовуваних методик проектування. Нова назва методики, прийняте в якості стандарту - IDEF0 [15].

IDEF-методика створювалася в США в рамках програми комп'ютеризації промисловості ICAM (Integrated Computer Aided Manufacturing - вбудована автоматизація виробництва). Назва стандарту розшифровується як Icam DEFinition.

Методологія SADT являє собою сукупність методів, правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Функціональна модель SADT відображає функціональну структуру об'єкта, тобто вироблені їм дії і зв'язку між цими діями. Розробку SADT-моделі починають з формулювання питань, на які модель повинна давати відповіді, тобто формулюють мету моделювання. Процес моделювання в SADT включає збір інформації про досліджувану область, документування отриманої інформації і представлення її у вигляді моделі і уточнення моделі. Далі будують ієрархічну сукупність діаграм з лаконічним описом функцій. SADT унікальна в своїй здатності забезпечити як графічна мова, так і процес створення несуперечливої і корисною системи описів.

У SADT система представляється як сукупність взаємодіючих робіт (або функцій). Зв'язки між роботами визначають технологічний процес або структуру взаємозв'язку всередині організації. Модель SADT являє собою серію діаграм, розбивають складний об'єкт на складові частини [13].

Кожен блок SADT-діаграми може бути представлений декількома блоками, з'єднаними інтерфейсними дугами, на діаграмі наступного рівня. Ці блоки представляють підфункції (підмодулі) вихідної функції. Кожен з підмодулей може бути декомпозирован аналогічним чином.

Правила SADT включають:

- обмеження кількості блоків на кожному рівні декомпозиції (не більше 6 блоків, інакше сприйняття діаграм буде утруднено);
- число рівнів ієрархії не обмежена, але зазвичай їх не більше 5;



- зв'язність діаграм (блоки нумеруються, номер записується в правому нижньому кутку);
- унікальність міток і найменувань (відсутність повторюваних імен);
- синтаксичні правила для графіки (блоків і дуг);
- поділ входів та управлінь (правило визначення ролі даних).

Методологія SADT може використовуватися для моделювання широкого кола систем і визначення вимог і функцій, а потім для розробки системи, яка задовольняє цим вимогам і реалізує ці функції. Для вже існуючих систем SADT може бути використана для аналізу функцій, виконуваних системою, а також для вказівки механізмів, за допомогою яких вони здійснюються.

Результатом застосування методології SADT є модель, яка складається з діаграм, фрагментів текстів і глосарію, що мають посилання один на одного [16].

– Діаграми - головні компоненти моделі. Двома найбільш важливими компонентами діаграм є бізнес функції або роботи (представлені на діаграмах у вигляді блоків) і дані або об'єкти (зображувані у вигляді стрілок), що зв'язують між собою роботи (рисунок 1.1). При цьому стрілки, в залежності від того в яку грань прямокутника роботи вони входять або з якої межі виходять, діляться на п'ять видів:

- стрілки входу (входять в ліву грань роботи) - зображують дані або об'єкти, що змінюються в ході виконання роботи;
- стрілки управління (входять у верхню грань роботи) - зображують правила і обмеження, згідно з якими виконується робота;
- стрілки виходу (виходять з правої межі роботи) - зображують дані або об'єкти, що з'являються в результаті виконання роботи;
- стрілки механізму (входять в нижню межу роботи) - зображують ресурси, необхідні для виконання роботи, але не змінюються в процесі роботи (наприклад, обладнання, людські ресурси ...);
- стрілки виклику (виходять з нижньої межі роботи) - зображують зв'язку між різними діаграмами або моделями, вказуючи на деяку діаграму, де дана робота розглянута більш докладно [15].

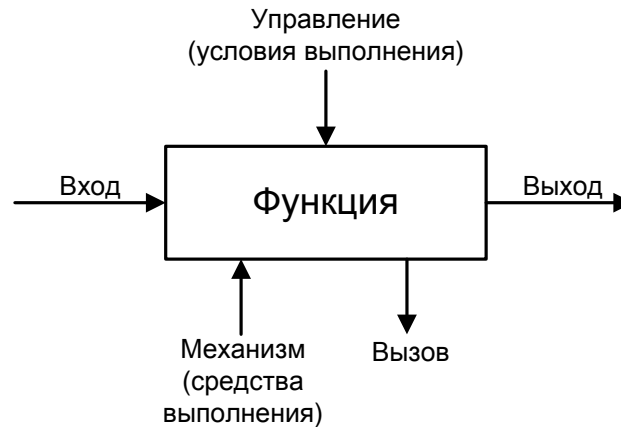


Рисунок 1.1 - Функціональний блок і інтерфейсні дуги

Однією з найбільш важливих особливостей методу SADT є поступове введення все більших рівнів деталізації у міру створення діаграм, що відображають модель. Кожен компонент моделі може бути декомпозований на іншій діаграмі. Кожна діаграма ілюструє "внутрішню будову" блоку на батьківській діаграмі.

Недоліки SADT-моделей - їх слабка формалізованість для автоматичного виконання проектних процедур на їх основі. У SADT-моделі не можна побачити алгоритму виконання процесів. Модель вимагає певної підготовки для розробки і читання нотації. Однак наявність графічного мови діаграм, зручного для сприйняття людиною, обумовлює корисність і придатність методики SADT. Ще однією перевагою методики є те, що вона показує взаємодію процесів в загальному вигляді, без зайвих подробиць.

### 1.3.2 Діаграми потоків даних DFD

Діаграмою потоків даних (Data Flow Diagrams - DFD) називається методологія графічного структурного аналізу, що описує зовнішні по відношенню до системи джерела і адресати даних, логічні функції, потоки даних і сховища даних, до яких здійснюється доступ. Діаграми потоків даних є ієрархією функціональних процесів, пов'язаних потоками даних [17]. Мета такого уявлення - продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Діаграми потоків даних використовуються для опису руху документів і обробки інформації як доповнення до IDEF0. На відміну від IDEF0, де система розглядається як пов'язані між собою роботи і стрілки представляють собою жорсткі взаємозв'язку, стрілки

в DFD показують лише те, як об'єкти (включаючи дані) рухаються від однієї роботи до іншої.

Історично склалося так, що для побудови діаграм DFD традиційно використовуються дві різні нотації, які відповідають методам Йордона-ДеМарко і Гейне-Серсона. Ці нотації мають однакову базову функціональність і незначно відрізняються один від одного графічним зображенням символів.

DFD - це граф, на якому показано рух значень даних від їх джерел через перетворюють їх процеси до їх споживачам в інших об'єктах.

Розробка DFD починається з побудови діаграми верхнього рівня, що відбиває зв'язку програмної системи, представленої у вигляді єдиного процесу, з зовнішнім середовищем. Для позначення елементів середовища функціонування системи використовується поняття зовнішньої сутності. Усередині системи існують процеси перетворення інформації, які породжують нові потоки даних. Потоки даних можуть надходити на вхід до інших процесів, поміщатися (і вилучатись) в накопичувачі даних, передаватися до зовнішніх сутностей [13].

Модель DFD, як і більшість інших структурних моделей - ієрархічна модель. Кожен процес може бути підданий декомпозиції, тобто Розбиття на структурні складові, відносини між якими в тій же нотації можуть бути показані на окремій діаграмі. Декомпозиція процесу проводиться до рівня, на якому фігурують елементарні процеси. Процес нижнього рівня супроводжується міні-специфікацією (текстовим описом).

Основними компонентами діаграм потоків даних DFD є:

- зовнішні сутності
- процеси
- накопичувачі даних або сховища
- системи / підсистеми
- потоки даних

Зовнішня сутність є матеріальним об'єктом або фізична особа, що є джерелом або приймачем інформації, наприклад, замовники, персонал, постачальники, клієнти, склад. Визначення деякого об'єкту або системи в якості зовнішньої суті вказує на те, що вона знаходиться за межами кордонів аналізованої системи. У процесі аналізу деякі зовнішні сутності можуть бути перенесені всередину діаграми аналізованої системи, якщо це необхідно, або, навпаки, частина процесів може бути винесена за межі діаграми і представлена як зовнішня сутність.

Зовнішня сутність позначається прямокутником з тінню (рисунок 1.2), всередині якого вказується її ім'я. При цьому в якості імені рекомендується використовувати іменник в називному відмінку. Іноді зовнішню сутність називають також термінатором.



Рисунок 1.2 - Зображення зовнішньої сутності на діаграмі потоків даних

Процес являє собою перетворення вхідних потоків даних у вихідні відповідно до певного алгоритму або правилом. Весь граф потоку даних являє собою процес високого рівня. Процес на діаграмі потоків даних зображується прямокутником із закругленими вершинами, сенс його збігається зі змістом робіт в SADT діаграмах. Кожен процес має фіксоване число вхідних і вихідних даних, зображуваних стрілками, але не підтримує управління і механізми, як в SADT діаграмах.

Інформаційна модель системи будується як деяка ієрархічна схема у вигляді так званої контекстної діаграми, на якій вихідна модель послідовно представляється у вигляді моделі підсистем відповідних процесів перетворення даних. При цьому підсистема або система на контекстній діаграмі DFD зображається так само, як і процес - прямокутником із закругленими вершинами.

Накопичувач даних або сховище являє собою пасивний об'єкт в складі DFD, в якому зберігаються дані, що переміщуються між процесами, для подальшого доступу. Дані можна в будь-який момент помістити в накопичувач і через деякий час витягнути, причому фізичні способи приміщення і вилучення даних можуть бути довільними. Накопичувач даних може бути фізично реалізований різними способами, але найбільш часто передбачається його реалізація в електронному вигляді на магнітних носіях.

Накопичувач даних на діаграмі потоків даних зображується, як показано на рис. 1.3.



Рисунок 1.3 - Зображення накопичувача на діаграмі потоків даних

Накопичувач даних в загальному випадку є прообразом майбутньої бази даних, і опис зберігаються в ньому даних має відповідати моделі даних.

Потік даних визначає якісний характер інформації, переданої через деякий з'єднання від джерела до приймача. Реальний потік даних може бути інформацією, переданої по кабелю між двома пристроями, що пересилаються поштою листами, по мережі між двома комп'ютерами або будь-яким іншим способом, що допускає вилучення даних та їх відновлення в необхідному форматі. Потік даних на діаграмі DFD зображається лінією зі стрілкою на одному з її кінців, яка показує напрямок потоку даних. Кожен потік даних має своє власне ім'я, що відбиває його зміст.

Нотація DFD - зручний засіб для формування контекстної діаграми, тобто Діаграми, яка б показала, що розробляється автоматизовану інформаційну систему в комунікації із зовнішнім середовищем [17]. Це діаграма верхнього рівня в ієрархії діаграм DFD. Її призначення - обмежити рамки системи, визначити, де закінчується розробляється система і починається середу.

### 1.3.3 Діаграми «сутність-зв'язок» ERD

Найбільш поширеним засобом моделювання даних є діаграми «сутність-зв'язок» (Entity-Relationship Diagrams - ERD).

Діаграми «сутність-зв'язок» (ERD) призначені для розробки моделей даних і забезпечують стандартний спосіб визначення даних і відносин між ними. Фактично за допомогою ERD здійснюється деталізація даних проєктованої системи, а також документуються сутності системи і способи їх взаємодії, включаючи ідентифікацію об'єктів, важливих для предметної області (сутностей), властивостей цих об'єктів (атрибутів) і їх відносин з іншими об'єктами (зв'язків). ERD безпосередньо використовуються насамперед для проєктування реляційних баз даних (хоча також можуть з успіхом застосовуватися і для моделювання як ієрархічних, так і мережевих баз даних) [13].

Мета моделювання даних полягає в забезпеченні розробника ІС концептуальною схемою бази даних у формі однієї моделі або кількох локальних моделей, які відносно легко можуть бути відображені в будь-яку систему баз даних.

Нотація ERD вперше була запропонована П. Ченом в 1976 році і отримала розвиток в роботах Р. Баркера. Подальший розвиток цього методу виражено в нотаціях IDEF1 і IDEF1X. Але, по суті, всі ці варіанти діаграм «сутність-зв'язок» є еквівалентними і виходять з однієї ідеї - малюнок завжди наочніше текстового опису. Адже діаграми «сутність-зв'язок» (ERD) призначені для графічного представлення моделей даних

розробляється програмної системи і пропонують деякий набір стандартних позначень для визначення даних і відносин між ними [16]. Різниця цих нотацій в основному відбуваються саме в позначеннях. За допомогою такого виду діаграм можна описати окремі компоненти концептуальної моделі даних і сукупність взаємозв'язків між ними, що мають важливе значення для розроблюваної системи.

Розглянемо діаграму «сутність-зв'язок», запропоновану П. Ченом, т. К. Вона стала основою для розвитку і вдосконалення в інших нотаціях [18].

Основними поняттями даної нотації є поняття сутності і зв'язку. При цьому під сутністю (entity) розуміється довільна безліч екземплярів реальних або абстрактних об'єктів (людей, подій, станів, ідей, предметів і т.п.), кожен з яких володіє однаковими властивостями і характеристиками. У цьому випадку кожен даний об'єкт може бути екземпляром однієї і тільки однієї сутності, повинен мати унікальне ім'я або ідентифікатор, а також відрізнятися від інших екземплярів даної сутності. При цьому ім'я сутності повинно відображати тип або клас об'єкта, а не його конкретний екземпляр.

Кожна з сутностей може розглядатися з різним ступенем деталізації і на різному рівні абстракції, що визначається конкретно постановкою завдання. Для графічного представлення сутностей використовуються спеціальні позначення, представлені на рис. 1.4.



Рисунок 1.4 - Графічні зображення для позначення сутностей

Незалежна сутність представляє незалежні дані, які завжди присутні в системі. При цьому відносини з іншими сутностями можуть як існувати, так і бути відсутнім. У свою чергу залежна сутність представляє дані, що залежать від інших сутностей в системі. Тому вона повинна завжди мати відносини з іншими сутностями. Асоційована сутність представляє дані, які асоціюються з відносинами між двома і більше сутностями.

Зв'язок (relationship) визначається як відношення або деяка асоціація між двома і більше сутностями. Різні типи зв'язків графічно зображаються у формі ромба з відповідним ім'ям зв'язку з цим (рисунок 1.5).

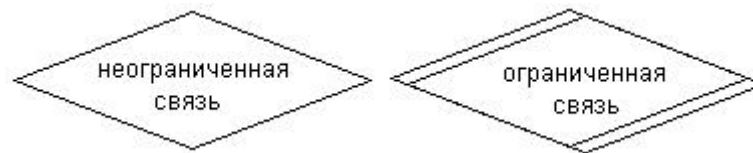


Рисунок 1.5 - Графічні зображення для позначення зв'язків (відносин)

Необмежена (обов'язкова) зв'язок являє собою безумовну зв'язок, тобто зв'язок, яка завжди існує до тих пір, поки існують стосуються справи суті. Обмежена (необов'язкова) зв'язок являє собою умовне відношення між сутностями.

Іншими словами, сутності представляють собою базові типи інформації, що зберігається в базі даних, а відносини показують, як ці типи даних взаємопов'язані один з одним. Введення подібних відносин переслідує дві основні мети:

- забезпечення зберігання інформації в єдиному місці (навіть якщо вона використовується в різних комбінаціях);
- використання цієї інформації різними додатками.

Графічна модель даних будується таким чином, щоб зв'язки між окремими сутностями відображали не тільки семантичний характер відповідного ставлення, а й додаткові аспекти обов'язковості зв'язків, а також кратність беруть участь в цих відносинах примірників сутностей.

Для ідентифікації вимог, відповідно до яких сутності залучаються у відносини, використовуються зв'язки між сутностями. Зв'язок - це асоціація між сутностями, при якій, як правило, кожен екземпляр однієї сутності асоційований з довільним (в тому числі нульовим) кількістю примірників другий сутності. Зв'язки дозволяють по одній сутності знаходити інші сутності, пов'язані з нею [13].

Однією з основних характеристик зв'язку є тип зв'язку. Пара значень зв'язків, що належать одній і тій же суті, визначає тип цієї сутності. Практика показала, що для більшості додатків досить використовувати такі типи зв'язків:

- зв'язок  $1 * 1$  (один-до-одного) означає, що один екземпляр першої суті пов'язаний з одним екземпляром другої сутності. Зв'язок даного типу використовуються, як правило, на верхніх рівнях ієрархії моделі даних, а на нижніх рівнях зустрічаються порівняно рідко;
- зв'язок  $1 * N$  (один-ко-многим) означає, що один екземпляр першої суті пов'язаний з декількома екземплярами другої суті. Зв'язок даного типу є найбільш часто респонденти користуються послугами;
- зв'язок  $N * M$  (багато-до-багатьох) означає, що кожен екземпляр першої суті може бути пов'язаний з декількома екземплярами другої суті, і кожен примірник другої

сутності може бути пов'язаний з декількома екземплярами першої сутності. Зв'язок даного типу зазвичай використовуються на ранніх етапах проектування з метою прояснення ситуації. В подальшому кожна з таких відносин має бути перетворено в комбінацію відносин типів 1 і 2 (можливо, з додаванням допоміжних сутностей і з введенням нових відносин).

Атрибут - поійменована характеристика сутності, значима для розглянутої предметної області. За допомогою атрибутів моделюються властивості сутностей. Кожна сутність володіє одним або декількома атрибутами, які однозначно ідентифікують кожен екземпляр сутності. При цьому будь-який атрибут може бути визначений як ключовий, який унікальним чином ідентифікує екземпляри сутності.

Діаграми «сутність-зв'язок» дозволяють використовувати наочні графічні позначення для моделювання сутностей і їх взаємозв'язків. Основна перевага методу полягає в тому, що модель будується методом послідовних уточнень первинних діаграм.

### **1.3.4 Методологія IDEF3**

Поведінковий моделювання складних систем використовують для дослідження динаміки їх функціонування. В основі поведінкового моделювання лежать моделі і методи імітаційного моделювання, тобто моделі, які описують поведінку системи, як послідовність зміни станів.

Поведінкові аспекти додатків відображає методика IDEF3, що є частиною сімейства стандартів IDEF, була розроблена в кінці 1980-х років. Ця методика призначена для таких моделей процесів, в яких важливо зрозуміти тимчасову послідовність виконання дій і алгоритмів виконання робіт, а також взаємозалежності між ними [18].

Методологія моделювання IDEF3 - це стандарт документування процесів, що відбуваються в системі, який надає собою інструментарій для наочного дослідження і моделювання їх сценаріїв. Сценарієм називається опис послідовності змін властивостей об'єкта, в рамках даного процесу.

Методологія IDEF3 показує причинно-наслідкові зв'язки між ситуаціями і подіями, що відбуваються в системі, в зрозумілій експерту формі, використовуючи структурний метод вираження знань про те, як функціонує система або процес. Система описується як упорядкована послідовність подій з одночасним описом об'єктів, що мають відношення до моделюваному процесу.



Отже, якщо методика SADT, призначена для опису процесів верхнього рівня, пов'язана з функціональними аспектами і дозволяє відповідати на питання "Що робить система?", то в методиці IDEF3, призначеної для опису процесів нижнього рівня, деталізуються і конкретизуються SADT-функції. IDEF3-модель відповідає на питання "Як система це робить?". Мова IDEF3 - мова діаграм, який допомагає розробнику моделей наочно уявити моделюються процеси.

IDEF3 складається з двох методів:

- Process Flow Description (PFD) - опис технологічних процесів, із зазначенням того, що відбувається на кожному етапі технологічного процесу;
- Object State Transition Description (OSTD) - опис переходів станів об'єктів, із зазначенням того, які існують проміжні стани у об'єктів в моделюється системі.

В даний час застосовуються діаграми стандарту IDEF3 двох типів. Вони являють собою опис одних і тих же сценаріїв процесів виробництва, проте, розглянутих з різних точок зору.

Типи діаграм:

- Process Flow Description Diagrams (PFDD) - процес-орієнтовані діаграми, що виражаються у вигляді послідовності операцій, які використовуються для опису послідовностей етапів відповідного процесу;
- Object State Transition Network (OSTN) - об'єкт-орієнтовані опису, що виражаються діаграмами переходу станів, які визначають стану об'єктів і їх змін в процесі [13].

Тобто Діаграми PFDD розглядають процес "з точки зору спостерігача" (використовуються для опису процесу), а діаграми OSTN дозволяють розглядати той же самий процес "з точки зору об'єкта" (використовуються для ілюстрації трансформацій об'єкта, які відбуваються на кожній стадії виконання відповідних робіт).

На рис. 1.6 представлений приклад PFDD діаграми. Тут функціональні елементи або елементи поведінки (UOB - Unit of Behavior), які позначають подія, стадію процесу або прийняття рішення, показані прямокутниками з горизонтальною лінією, яка відділяє верхню секцію з назвою функції від нижньої секції, яка містить унікальний номер функції.

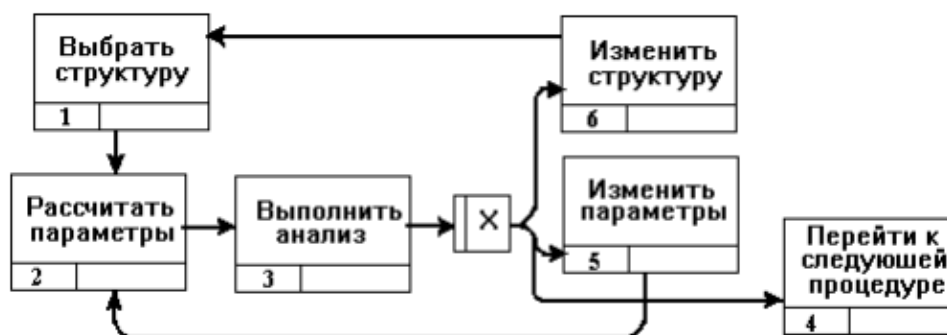


Рисунок 1.6 - Приклад PFDD діаграми (діаграма послідовності операцій)

Зв'язки (рисунок 1.6), що відображають послідовність виконання функцій, зображуються суцільними лініями-стрілками. У класі діаграм IDEF3 PFDD зв'язку між роботами діляться на три типи, позначення, назви та змів яких, наведені в таблиці 1.1.

Таблиця 1.1 - Типи зв'язків між роботами

Назва зв'язку	Вид зв'язку	Сенс зв'язку
1	2	3
зв'язок передування		Позначає, що друга робота починає виконуватися після завершення першої роботи.
зв'язок відносини		Позначає, що друга робота може початися і навіть закінчитися до того моменту, коли закінчиться виконання першої роботи.
Зв'язок потоків об'єктів		Одночасно позначає тимчасову послідовність робіт і матеріальний чи інформаційний потік. В даному випадку друга робота починає виконуватися після завершення першої роботи. При цьому виходом першої роботи об'єкт, назва якого надписано над стрілкою (в даному випадку документ). Цей зв'язок також означає, що об'єкт, який породжується першою роботою, використовується в наступних роботах.

Для вказівки розгалужень і злиттів зв'язків (їх прийнято називати перехрестями) використовують квадрати, у яких одна або обидві вертикальні боку представлені подвійними лініями, а всередині квадрата записаний логічний оператор (рисунок 1.6). При

розбіжності ці символи означають реакцію всіх, деяких або тільки однієї з наступних функцій на вхідний вплив відповідно. Аналогічний сенс мають логічні оператори при сходженні - подальша функція починає виконуватися після закінчення всіх, деяких або тільки однієї з вхідних операцій [17].

Перехрестя також діляться на кілька типів: "виключає АБО", "І" і "АБО". У таблиці 1.2 наведені позначення, назви та значення всіх типів перехресть, як в схемах сходження, так і в схемах розбіжності.

На рис. 1.7 представлений приклад діаграми мережі зміни станів об'єктів (OSTN). У таких діаграмах є засоби для зображення станів системи, активності, переходів зі стану в стан і умов переходу.

Таблиця 1.2 - Позначення, назви і зміст типів перехресть в схемах сходження і розбіжності.

Назва перехресть		позначення перехресть	сенс перехресть	
			схема розбіжності	схема сходження
1		2	3	4
<b>"Виключає АБО"</b>			Тільки одна подальша робота запускається	Тільки одна попередня робота повинна бути завершена
<b>"І"</b>	<b>асинхронний</b>		Усі наступні роботи запускаються всі попередні роботи повинні бути завершені	Усі наступні роботи запускаються всі попередні роботи повинні бути завершені
	<b>синхронний</b>		Усі наступні роботи запускаються одночасно всі попередні роботи повинні бути завершені одночасно	Усі наступні роботи запускаються одночасно всі попередні роботи повинні бути завершені одночасно

Продовження табл. 1.1.

1		2	3	4
" АБО "	<b>асинхронний</b>	□ ○	Одна або кілька наступних робіт запускаються Одна або кілька попередніх робіт повинні бути завершені	Одна або кілька наступних робіт запускаються Одна або кілька попередніх робіт повинні бути завершені
	<b>синхронний</b>	□ □ ○	Одна або кілька наступних робіт запускаються одночасно Одна або кілька попередніх робіт повинні бути завершені одночасно	Одна або кілька наступних робіт запускаються одночасно Одна або кілька попередніх робіт повинні бути завершені одночасно



Рисунок 1.7 - Приклад OSTN діаграми (діаграма переходу станів)

Діаграма OSTN розглядає можливі зміни, які відбуваються з об'єктами в ході окремого процесу, тому ключовими поняттями OSTN діаграми є стан об'єкта і зміна стану.

Стану об'єкта відображаються колами з найменуванням об'єкта всередині цих кіл, а їх зміни спрямованими лініями. Кожна лінія має посилання на відповідний функціональний блок UOB, в результаті якого відбулося відображається їй зміна стану об'єкта.

Для зображення послідовностей переходів об'єктів в OSTN діаграмах використовуються зв'язку переходів. Зв'язку переходів можуть бути слабкими і сильними.

Слабкі зв'язку переходів зображуються суцільними простими стрілками (рисунок 1.8, а) і говорять про те, що об'єкту виду В передуює об'єкт виду А або що станом В деякого об'єкту передуює його стан А.



Рисунок 1.8 - Зв'язку переходів в OSTN діаграмах

Сильні зв'язку переходів зображуються подвійними односпрямованими стрілками (рисунок 1.8, б) і підкреслюють, що об'єкту виду В повинен передувати об'єкт виду А або що стан В об'єкта можна досягти тільки зі стану А.

Механізм перехрестя є інструментом для опису і відображення логіки можливих альтернативних варіантів перетворень об'єктів.

На діаграмах OSTN перехрестя, як і символи об'єктів, зображуються кружками, всередині яких міститься умовне позначення логічної функції, яка реалізується вузлом. Як і в PFDD діаграмах вузли можуть бути вузлами сходження і розбіжності. Як логічних функцій можуть використовуватися І (&), АБО (O) і виключає Або (X) [19].

Опис процесу в IDEF3 може містити діаграми PFDD і OSTN або діаграми будь-якого одного типу.

На рис. 1.9 відображена дисципліна взаємного розташування діаграм в різних методиках.

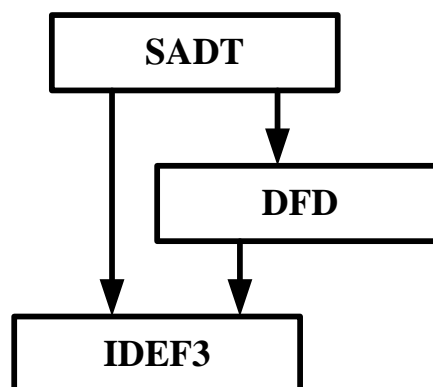


Рисунок 1.9 - Розташування діаграм в різних методиках

Дані аналізу, отримані з використанням моделювання SADT, зазвичай використовуються на стадії розробки моделей IDEF3 і діаграм потоків даних DFD. SADT діаграма може бути декомпозована або в SADT, або в IDEF3, або в DFD. У свою чергу DFD діаграма може бути декомпозована в DFD або в IDEF3. IDEF3 діаграма декомпозірується тільки в IDEF3.

## **1.4 Об'єктно-орієнтований підхід до проектування**

### **1.4.1 Уніфікована мова моделювання UML**

Об'єктно-орієнтоване проектування засноване на об'єктно-орієнтованій декомпозиції. При цьому структура системи описується в термінах об'єктів і зв'язків між ними, а поведінка системи описується в термінах обміну повідомленнями між об'єктами.

При об'єктно-орієнтованій декомпозиції кожен об'єкт володіє своєю власною поведінкою і кожен з них моделює деякий об'єкт реального світу. З цієї точки зору об'єкт є цілком відчутною річчю, яка демонструє цілком певну поведінку. Об'єкти щось роблять, і, пославши їм повідомлення, можна попросити їх виконати деякі операції.

Фактично всі складні системи можна уявити однієї і тієї ж канонічної формою - у вигляді двох ортогональних ієрархій однієї системи: класів і об'єктів. Кожна ієрархія є багаторівневою, причому в ній класи і об'єкти більш високого рівня побудовані з більш простих. Який клас або об'єкт обраний в якості елементарного, залежить від розглянутої задачі. Об'єкти одного рівня мають чітко виражені зв'язку, особливо це стосується компонентів структури об'єктів. Усередині будь-якого розглянутого рівня знаходиться наступний рівень складності. Структури класів і об'єктів не є незалежними: кожен елемент структури об'єктів представляє специфічний екземпляр певного класу. Об'єктів в складній системі зазвичай набагато більше, ніж класів. З введенням структури класів в ній розміщуються загальні властивості екземплярів класів.

Популярність об'єктно-орієнтованих технологій привела до появи різних методологій з власними наборами нотацій. Нотація є важливою складовою будь-якої моделі, так як вона служить сполучною ланкою між процесами. За визначенням Граді Буча нотація виконує наступні функції:

- є мовою для опису взаємодій, які неочевидні або не можуть бути отримані безпосередньо з коду;
- забезпечує достатню семантику, що дозволяє охопити важливі стратегічні і тактичні рішення;

– пропонує конкретну форму, що допомагає людині міркувати про предметну область, а засобами моделювання втілювати описані ідеї.

Уніфікована мова моделювання (Unified Modeling Language - UML) пропонує досить повну нотацію, яка розширюється при переході від аналізу до проектування.

Мова UML є об'єкто-орієнтованою мовою візуального моделювання, який розроблений для специфікації, візуалізації, проектування та документування компонентів програмного забезпечення, бізнес-процесів і інших систем [17]. Мова UML одночасно є простим і потужним засобом моделювання, який може бути ефективно використаний для побудови концептуальних, логічних і графічних моделей складних систем самого різного цільового призначення. Ця мова увібрав в себе найкращі якості методів програмної інженерії, які з успіхом використовувалися впродовж останніх років при моделюванні великих і складних систем. В даний час UML фактично є стандартом в області об'єкто-орієнтованого аналізу і проектування.

Моделювання в UML проводиться як поуровневого спуск від концептуальної моделі до логічної, а потім до фізичної моделі програмної системи [14].

Концептуальна модель виражається у вигляді діаграми варіантів використання, яка описує функціональне призначення системи.

Логічна модель дозволяє визначити два різних погляди на систему: статичний і динамічний.

Статичний підхід виражається діаграмами класів. Поведінка конкретного класу може бути виражено за допомогою діаграм станів, що визначають модель кінцевого автомата.

Динамічний підхід описується також діаграмами поведінки, до яких відносяться діаграми послідовності і діаграми діяльності.

Фізична модель задається діаграмою компонентів, яка описує розподіл реалізації класів по модулях, і діаграмою розгортання.

Діаграми не закріплені жорстко за окремими етапами життєвого циклу розробки проекту. Більшість діаграм з різним ступенем деталізації може бути використано на етапах системного аналізу, проектування, реалізації і т.д.

Кожен вид діаграми відображає різні аспекти бачення і розуміння моделі. У моделі може бути кілька діаграм кожного з перерахованих типів.

### 1.4.2 Діаграми варіантів використання

Концептуальна модель виражається у вигляді діаграми варіантів використання, яка описує функціональне призначення і поведінку системи, і дозволяє замовнику, кінцевому користувачеві і розробнику спільно обговорювати проєктовану або існуючу систему. У кожній системі зазвичай є головна діаграма, яка відображає межі системи (акторів) і основне функціональне поведінка системи (варіанти використання).

Розробка діаграми варіантів використання переслідує такі цілі:

- визначити спільні кордони і контекст модельованої предметної області на початкових етапах проєктування системи;
- сформулювати загальні вимоги до функціонального поведінки проєктованої системи;
- розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей;
- підготувати вихідну документацію для взаємодії розробників системи з її замовниками і користувачами.

Суть даної діаграми полягає в тому, що проєктована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою варіантів використання. При цьому актором або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на моделювану систему так, як визначить сам розробник [14].

Варіант використання служить для опису сервісів, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який чинять системою при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів з системою.

Графічні позначення актора і варіанти використання представлені на рис. 1.10 а, б.

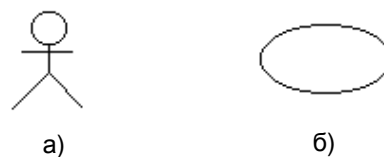


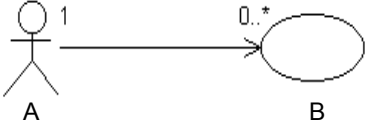
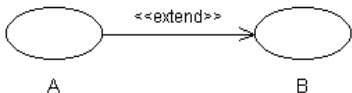
Рисунок 1.10 - Графічне позначення актора (а) і варіанти використання (б)

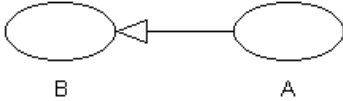


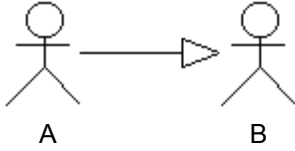
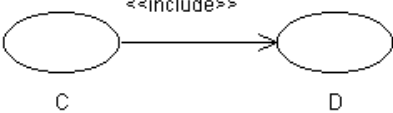
Між компонентами діаграми варіантів використання можуть існувати різні відносини, які описують взаємодію екземплярів одних акторів і варіантів використання з екземплярами інших акторів і варіантів. Один актор може взаємодіяти з декількома варіантами використання. В цьому випадку цей актор звертається до кількох сервісів даної системи. У свою чергу один варіант використання може взаємодіяти з декількома акторами, надаючи для всіх них свій сервіс.

В UML є кілька стандартних видів відносин між акторами і варіантами використання, які наведені в таблиці 1.3.

Таблиця 1.3 - Види відносин між акторами і варіантами використання

вид відносини	Графічне зображення відносини	опис відносини
Ставлення асоціації (association relationship)	 <p>Позначається суцільною лінією між актором і варіантом використання. Ця лінія може мати додаткові умовні позначення, такі, наприклад, як ім'я, напрям і кратність *.</p>	Служить для позначення специфічної ролі актора в окремому варіанті використання. Встановлює, яку конкретну роль грає актор при взаємодії з екземпляром варіанту використання.
<p>* Кратність характеризує загальна кількість конкретних екземплярів даного компонента, які можуть виступати в якості елементів даної асоціації. Для позначення кратності використовуються цифри і символ "*" (зірочка). Якщо кратність відносини асоціації не вказана, то за замовчуванням приймається її значення, рівне 1.</p>		
Ставлення розширення (extend relationship)	 <p>Позначається суцільною лінією між варіантами використання. Ставлення</p>	Визначає взаємозв'язок екземплярів окремого випадку використання з більш загальним варіантом, властивості якого визначаються на основі способу спільного об'єднання даних екземплярів (тобто Один з варіантів використання може приєднувати до своєї поведінки деяку додаткову поведінку,

	<p>позначається ключовим словом «extend» («розширює»).</p> <p>Стрілка вказує на те, що властивості екземпляра варіанту використання В можуть бути доповнені завдяки наявності властивостей у розширеного варіанту використання А.</p>	<p>певне для іншого варіанту використання).</p> <p>Відносини розширення застосовуються для відображення: додаткових режимів; режимів, які запускаються тільки за певних умов; альтернативних потоків, які запускаються на вибір актора.</p> <p>Один з варіантів використання може бути розширенням для декількох базових варіантів, а також мати в якості власних розширень кілька інших варіантів. Базовий варіант використання може додатково ніяк не залежати від своїх розширень.</p>
<p>Ставлення узагальнення (generalization relationship)</p>	 <p>Позначається суцільною лінією зі стрілкою в формі не закрашеного трикутника, яка вказує на батьківський варіант використання. Ця лінія зі стрілкою має спеціальну назву - стрілка «узагальнення».</p>	<p>Ставлення узагальнення служить для вказівки того факту, що деякий варіант використання А може бути узагальнено до варіанту використання В. В цьому випадку варіант А буде спеціалізацією варіанту В. При цьому В називається предком або батьком по відношенню А, а варіант А - нащадком по відношенню до варіанту використання В. Нащадок успадковує всі властивості і поведінку свого батька, а також може бути доповнений новими властивостями і особливостями поведінки.</p> <p>Ставлення узагальнення між варіантами використання застосовується в тому випадку, коли необхідно відзначити, що дочірні варіанти використання мають всі атрибути і особливостями поведінки батьківських варіантів. При цьому дочірні варіанти використання беруть участь у всіх відносинах батьківських варіантів. У свою чергу, дочірні варіанти можуть наділятися</p>

		<p>новими властивостями поведінки, які відсутні у батьківських варіантів використання, а також уточнювати або модифікувати успадковані від них властивості поведінки.</p>
	 <p>Позначається стрілкою узагальнення, тобто суцільною лінією зі стрілкою в формі не закрашеного трикутника, яка вказує на батьківського актора.</p>	<p>Між окремими акторами також може існувати відношення узагальнення. Дане відношення є спрямованим і вказує на факт спеціалізації одних акторів щодо інших.</p> <p>Наприклад, ставлення узагальнення від актора А до актора В відзначає той факт, що кожен екземпляр актора А є одночасно екземпляром актора В і володіє всіма його властивостями. В цьому випадку актор У є батьком по відношенню до актора А, а актор А, відповідно, нащадком актора В. При цьому актор А має здатність грати таку ж безліч ролей, що і актор В.</p>
<p>Ставлення включення (include relationship)</p>	 <p>Позначається лінією зі стрілкою, спрямованою від базового варіанту використання до такого, що включається варіанту використання. При цьому відношення позначається ключовим словом «include» («включає»).</p> <p>Стрілка вказує на те, що кожен екземпляр варіанту 3 включає в себе</p>	<p>Вказує, що деякий заданий поведінка для одного варіанта використання включається як складовий компонент в послідовність поведінки іншого варіанту використання. Дане відношення є спрямованим бінарним відношенням в тому сенсі, що пара екземплярів варіантів використання завжди впорядкована щодо включення.</p> <p>Один варіант використання може бути включений в кілька інших варіантів, а також включати в себе інші варіанти. Включається варіант використання може бути незалежним від базового варіанту в тому сенсі, що він надає останньому деякий інкапсульоване поведінку, деталі реалізації якого приховані</p>

	функціональні властивості, задані для варіанту D. Ці властивості спеціалізують поведінку відповідного варіанту C на даній діаграмі.	від останнього і можуть бути легко перерозподілені між декількома включаються варіантами використання. Більш того, базовий варіант може залежати тільки від результатів виконання включається в нього поведінки, але не від структури включаються в нього варіантів.
--	---	--

### 1.4.3 Діаграми класів

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування, їх атрибути, методи і взаємозв'язку між ними. На стадії аналізу діаграми класів використовуються, щоб виділити загальні ролі і обов'язки сутностей, що забезпечують необхідну поведінку системи. Діаграма класів може відбивати, зокрема, різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи відносин. На діаграмі не вказується інформація про тимчасові аспектах функціонування системи.

Клас в мові UML служить для позначення безлічі об'єктів, які мають однаковою структурою, поведінкою і відносинами з об'єктами з інших класів. Графічно клас зображується у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на розділи або секції (малюнок 1.11). У цих розділах можуть зазначатися ім'я класу, атрибути (змінні) і операції (методи) [14].

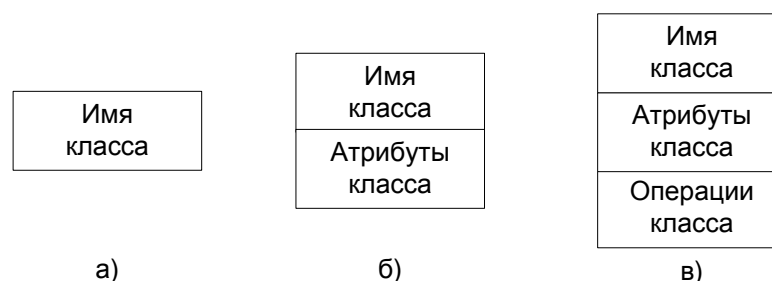


Рисунок 1.11 - Графічне зображення класу на діаграмі класів

Обов'язковим елементом позначення класу є його ім'я. На початкових етапах розробки діаграми окремі класи можуть позначатися простим прямокутником із

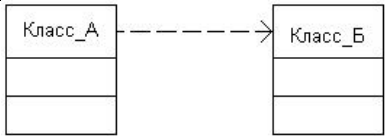
зазначенням тільки імені відповідного класу (малюнок 1.11, а). Згодом діаграми опису класів можуть доповнюватися атрибутами або властивостями (рисунок 1.11, б) і операціями або методами класу (малюнок 1.11, в). Операція являє собою деякий сервіс, що надає кожний екземпляр класу за певним вимогу. Сукупність операцій характеризує функціональний аспект поведінки класу.

Якщо секція атрибутів і операцій є марною, в позначенні класу вона виділяється горизонтальною лінією, щоб відразу відрізнити клас від інших елементів мови UML [20].


Ім'я класу повинно мати безпосереднє відношення до модельованої предметної області і функціональному призначенню проектованої системи.

Крім внутрішнього устрою або структури класів на відповідній діаграмі вказуються різні відносини між класами (таблиця 1.4). Кожне з цих відносин має власне графічне представлення на діаграмі, яке відображає взаємозв'язки між об'єктами відповідних класів.



Таблиця 1.4 - Види відносин на діаграмі класів

Вид відносини	Графічне зображення відносини	Опис відносини
1	2	3
Ставлення залежності (dependency relationship)	 <p>Зображується пунктирною лінією між відповідними елементами зі стрілкою на одному з її кінців. Дане відношення пов'язує окремі класи між собою, при цьому стрілка спрямована від класу-клієнта (Класс_А) залежно до незалежного класу або класу-джерела (Класс_Б).</p>	<p>Вказує деякий семантичне відношення між двома елементами моделі або двома множинами таких елементів, яке не є відношенням асоціації, узагальнення або реалізації. Воно стосується тільки самих елементів моделі. Ставлення залежності використовується в такій ситуації, коли деяка зміна одного елемента моделі може зажадати зміни іншого залежного від нього елемента моделі.</p>

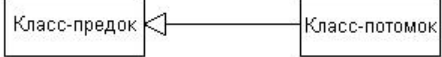

Продовження табл. 1.3.

1	2	3
<p>Ставлення асоціації (association relationship)</p>	 <p>Позначається суцільною лінією з додатковими спеціальними символами, які характеризують окремі властивості конкретної асоціації і є необов'язковими елементами (ім'я *, кратність **).</p> <p>Додаткові позначення також можуть використовуватися для відносин агрегації і композиції.</p>	<p>Показує, що об'єкти одного класу деяким чином пов'язані з об'єктами іншого або того ж самого класу.</p> <p>Допускається, щоб обидва кінці асоціації ставилися до одного класу. В асоціації можуть зв'язуватися два класи, і тоді вона називається бінарною, або відразу n класів (вони називаються n-арними асоціаціями).</p> <p>Спеціальною формою або окремим випадком відносини асоціації є відношення агрегації, яке, в свою чергу, теж має спеціальну форму - відношення композиції.</p>
<p>* Ім'я характеризує природу зв'язку. Сенс імені уточнюється за допомогою чорного трикутника, який розташовується над лінією зв'язку справа або зліва від імені асоціації. Цей трикутник вказує напрямок читання ім'я зв'язку. Його відсутність означає, що порядок проходження класів в даному відношенні не визначений.</p> <p>** Кратність полюса асоціації вказує, скільки об'єктів даного класу (з боку даного полюса) беруть участь в зв'язку. Кратність може бути задана як конкретне число (в кожному зв'язку з боку даного полюса бере участь рівно стільки об'єктів, скільки вказано) або як діапазон можливих значень (число об'єктів, що беруть участь в зв'язку має перебувати в межах зазначеного діапазону). При вказівці кратності можна використовувати символ *, який позначає невизначений число.</p>		

Продовження табл. 1.3.

1	2	3
<p>Ставлення агрегації (окремий випадок відносини асоціації)</p>	 <p>Зображується суцільною лінією, один з кінців якої являє собою не зафарбований усередині ромб. Цей ромб вказує на той з класів, який являє собою «ціле». Решта класи є його «частинами».</p>	<p>Ставлення агрегації має місце між декількома класами в тому випадку, якщо один з класів є деякою сутністю, що включає в себе в якості складових частин інші сутності. Застосовується для подання системних взаємозв'язків типу «частина-ціле».</p> <p>Показує, з яких компонентів складається система і як вони пов'язані між собою. Це відношення по своїй суті описує декомпозицію або розбиття складної системи на більш прості складові частини, які також можуть бути піддані декомпозиції, якщо в цьому виникне необхідність в подальшому. При цьому частини системи не повинні наслідувати її властивості і поведінку, оскільки є цілком самостійними сутностями. Більш того, частини цілого мають свої власні атрибути і операції, які істотно відрізняються від атрибутів і операцій цілого.</p>
<p>Ставлення композиції (окремий випадок відносини агрегації)</p>	 <p>Зображується суцільною лінією, один з кінців якої являє собою зафарбований усередині ромб. Цей ромб вказує на той з класів, який представляє собою клас-композицію або «ціле». Решта класи є його «частинами».</p>	<p>Служить для виділення спеціальної форми відносини «частина-ціле», при якій складові частини в деякому сенсі перебувають всередині цілого. Специфіка взаємозв'язку між ними полягає в тому, що частини не можуть виступати у відриві від цілого, тобто зі знищенням цілого знищуються і всі його складові частини.</p>

Продовження табл. 1.3.

1	2	3
<p>Ставлення узагальнення (generalization relationship)</p>	 <p>Позначається суцільною лінією з трикутною стрілкою на одному з кінців. Стрілка вказує на більш загальний клас (клас-предок або суперклас), а її відсутність - на більш спеціальний клас (клас-нащадок або підклас).</p>	<p>Ставлення узагальнення є відношенням між більш загальним елементом (батьком або предком) і більш приватним або спеціальним елементом (дочірнім або нащадком).</p> <p>Ставлення описує ієрархічну будову класів та успадкування їх властивостей і поведінки. При цьому передбачається, що клас-нащадок має всі властивості і поведінкою класу-предка, а також має свої власні властивості і поведінку, які відсутні у класу-предка.</p>
<p>Ставлення реалізації (realization relationship)</p>	 <p>Позначається пунктирною лінією з трикутною НЕ зафарбованою стрілкою на одному з кінців, як щось середнє між відносинами узагальнення і залежності. Стрілка спрямована на ту сутність, яка задає правила власної реалізації для сутності (постачальник), яка ці правила виконує (клієнт).</p>	<p>Реалізація - відношення між двома елементами моделі, в якому один елемент визначає зобов'язання (постачальник), а інший гарантує його виконання (клієнт). Реалізація - це відношення ціле-частина.</p>

#### 1.4.4 Діаграми станів

Діаграма станів є орієнтований граф, в якому вершини позначають стану, а дуги показують переходи між двома станами



Діаграма станів описує процес зміни станів тільки одного класу, а точніше - одного примірника певного класу, тобто Моделює всі можливі зміни в стані конкретного об'єкта. При цьому зміна стану об'єкта може бути викликано зовнішніми впливами з боку інших об'єктів або ззовні. Саме для опису реакції об'єкта на подібні зовнішні впливи і використовуються діаграми станів [14].

Стан - це певний стан в житті об'єкта, при якому він задовольняє певній умові, виконує деяку дію або очікує події. Стан об'єкта можна описати за допомогою значень одного або декількох атрибутів класу. Не кожен атрибут класу може характеризувати його стан. Як правило, мають значення тільки такі властивості елементів системи, які відображають динамічний або функціональний аспект її поведінки.

Переходи між станами є зміну одного стану іншим. Перебування модельованого об'єкта в першому стані може супроводжуватися виконанням деяких дій, а перехід в другий стан буде можливий після завершення цих дій, а також після задоволення деяких додаткових умов. У цьому випадку говорять, що перехід спрацьовує, До спрацьовування переходу об'єкт знаходиться в попередньому від нього стані, а після його спрацьовування об'єкт знаходиться в подальшому від нього стані (цільовому стані).

Спрацьовування переходу може залежати не тільки від настання деякої події, а й від виконання певної умови, званого сторожовим умовою. Об'єкт перейде з одного стану в інший в тому випадку, якщо сталося вказане подія і сторожова умова прийняло значення «істина».

Перехід може бути спрямований в той же стан, з якого він виходить. В цьому випадку його називають переходом в себе.

Термін подія вимагає окремого пояснення, оскільки є самостійним елементом мови UML. Формально, подія є специфікацію деякого факту, що має місце в просторі і в часі. Про події говорять, що вони «відбуваються», при цьому окремі події повинні бути впорядковані в часі. Після настання деякої події не можна вже повернутися до попередніх подій, якщо така можливість не передбачена явно в моделі.

Сторожове умова, якщо воно є, є деякий булевское вираз. Булевское вираз має приймати одне з двох взаємно виключають значень: «істина» або «брехня». Введення для переходу умови дозволяє явно специфікувати семантику його спрацьовування. Якщо умова приймає значення «істина», то відповідний перехід може спрацювати, в результаті чого об'єкт перейде в цільовий стан. Якщо ж умова приймає значення «брехня», то перехід не може спрацювати, і при відсутності інших переходів об'єкт не може перейти в цільовий стан по цьому переходу.

У загальному випадку з одного стану може бути кілька переходів з одним і тим же подією. При цьому ніякі два сторожових умови не повинні одночасно приймати значення «істина». Кожне з сторожових умов необхідно обчислювати щоразу при настанні відповідного події.

У діаграмах станів використовуються наступні умовні позначення:

- коло, що позначає початковий стан;
- коло з маленьким колом усередині, що позначає кінцеве стану (якщо є);
- прямокутник з округленими вершинами, що позначає стан. Верхівка прямокутника містить назву стану. В середині може бути горизонтальна лінія, під якою записуються активності, що відбуваються в даному стані;
- стрілка, що позначає перехід [21].

#### 1.4.5 Діаграми діяльності

При моделюванні поведінки проектованої системи виникає необхідність не тільки представити процес зміни її станів, але і деталізувати особливості алгоритмічної і логічної реалізації виконуваних системою операцій. Для моделювання процесу виконання операцій в мові UML використовуються діаграми діяльності.

Діаграма діяльності - UML-діаграма, на якій показано розкладання деякої діяльності на її складові частини. В контексті мови UML діяльність являє собою деяку сукупність окремих обчислень, виконуваних автоматом. При цьому окремі елементарні обчислення можуть призводити до деякого результату або дії. На діаграмі діяльності відображається логіка або послідовність переходу від однієї діяльності до іншої (від виходів одного вузла до входів іншого), при цьому увагу фіксується на результаті діяльності. Сам же результат може привести до зміни стану системи або повернення деякого значення [21].

Діаграми діяльності складаються з обмеженої кількості фігур, з'єднаних стрілками, які йдуть від початку до кінця процесу і показують послідовність переходів. Основні фігури:

- прямокутники з закругленими кутами позначають дії;
- ромби позначають рішення;
- широкі смуги позначають початок (розгалуження) і закінчення (сходження) розгалуження дій;

- чорний коло позначає початок процесу (початковий стан);
- чорний коло з обведенням позначає закінчення процесу (кінцевий стан).

Графічно діаграма діяльності видається у формі графа, вершинами якого є стани дії, а дугами - переходи від одного стану дії до іншого.

Застосовувана в діаграмах діяльності графічна нотація багато в чому схожа на нотацію діаграми станів, оскільки на діаграмах діяльності також присутні позначення станів і переходів. Відмінність полягає в семантиці станів, які використовуються для уявлення не діяльностей, а дій, і у відсутності на переходах сигнатури подій. Кожне стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні цієї операції.

Діаграми діяльності можна вважати окремим випадком діаграм станів. Саме вони дозволяють реалізувати в мові UML особливості процедурного і синхронного управління, обумовленого завершенням внутрішніх діяльностей і дій. Основним напрямком використання діаграм діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно представити алгоритми їх виконання. При цьому кожне стан може бути виконанням операції деякого класу або її частини, дозволяючи використовувати діаграми діяльності для опису реакцій на внутрішні події системи [14].

#### 1.4.6 Діаграми послідовності

Діаграма послідовності - діаграма, на якій показано взаємодію об'єктів (обмін між ними сигналами і повідомленнями), впорядковане за часом їх прояву, з відображенням тривалості обробки і послідовності їх прояви.

Діаграми послідовності в мові UML відносяться до діаграм взаємодії, які використовуються для подання тимчасових особливостей передачі і прийому повідомлень між об'єктами. Діаграми взаємодії, як правило, охоплюють поведінку тільки одного варіанту використання [14].

На діаграмі послідовності зображаються тільки ті об'єкти, які безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами. Діаграма послідовності має два виміри. Одне - зліва направо у вигляді вертикальних ліній, кожна з яких зображує лінію життя окремого об'єкта, який бере участь у взаємодії. Графічно кожен об'єкт зображується прямокутником і розташовується у верхній частині своєї лінії життя (рисунок 1.12). У середині прямокутника записується ім'я об'єкта і ім'я класу [13].

Ім'я об'єкта може бути відсутнім. У цьому випадку вказується тільки ім'я класу, а сам об'єкт вважається анонімним.

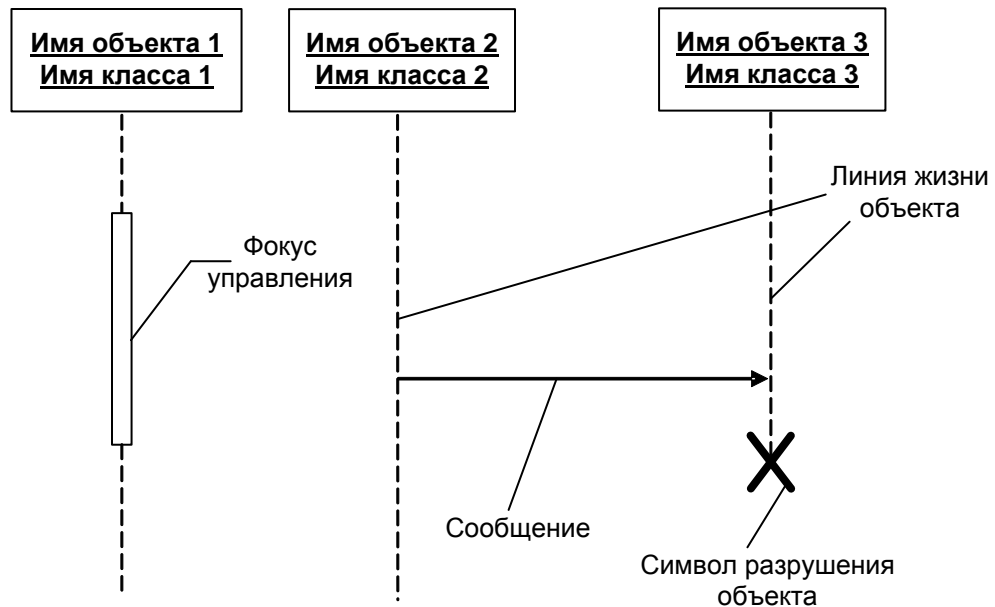


Рисунок 1.12 - Графічні примітиви діаграми послідовності

Крайнім зліва на діаграмі зображується об'єкт, який є ініціатором взаємодії. Праворуч зображується інший об'єкт, який безпосередньо взаємодіє з першим. Таким чином, всі об'єкти на діаграмі послідовності утворюють деякий порядок, який визначається ступенем активності цих об'єктів при взаємодії один з одним.

Другий вимір діаграми послідовності - вертикальна тимчасова вісь, спрямована зверху вниз. Початкового моменту часу відповідає сама верхня частина діаграми. При цьому взаємодії об'єктів реалізуються за допомогою повідомлень, які надсилаються одними об'єктами іншим. Повідомлення зображуються у вигляді горизонтальних стрілок з ім'ям повідомлення і так само утворюють порядок за часом свого виникнення. Іншими словами, повідомлення, розташовані на діаграмі послідовності вище, ініціюються раніше тих, які розташовані нижче. При цьому масштаб на осі часу не вказується, оскільки діаграма послідовності моделює лише тимчасову упорядкованість взаємодій типу «раніше - пізніше» [14].

### 1.4.7 Діаграми компонентів і розгортання

Діаграма компонентів - статична структурна діаграма, яка показує розбиття програмної системи на структурні компоненти (атомарні складові системи з точки зору її збірки, конфігураційного управління і розгортання) і зв'язку (залежності) між компонентами. Як фізичних компонентів можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети і т. П. Діаграма компонентів забезпечує узгоджений перехід від логічного представлення до конкретної реалізації проекту у формі програмного коду [21].

Графічно діаграма компонентів представляється у вигляді графа з ребрами і вершинами. Основними графічними елементами діаграми компонентів є компоненти, які зображуються у вигляді прямокутника з вкладками на лівій стороні (рисунок 1.13).

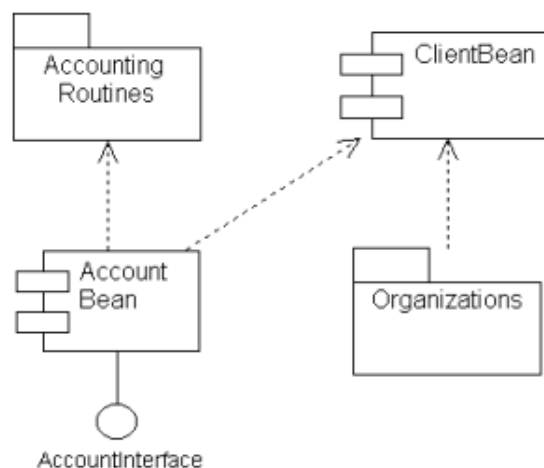


Рисунок 1.13 - Приклад діаграми компонентів

Зв'язки, що показують залежності між компонентами, зображуються пунктирними стрілками. Один компонент залежить від іншого, якщо він не може бути використаний у відсутності цього іншого компонента в конфігурації системи. Зв'язок між загальною сутністю (батьком) і її конкретним втіленням (нащадком) зображується у вигляді лінії з великої не зафарбованою стрілкою, спрямованої на батька (узагальнення). Зв'язок одного типу об'єктів з об'єктами іншого типу зображується у вигляді суцільної лінії (асоціація). Компоненти можуть також реалізовувати інтерфейси.

На діаграмі компонентів, зображеної на рис. 1.13, можна також переглянути список пакунків, зображуваних у вигляді «папок», точніше - прямокутників з прямокутними «наростами» над лівим верхнім кутом. Пакети є просторами імен і засобом угруповання діаграм і інших модельних елементів UML - класів, компонентів тощо.

Вони можуть з'являтися на діаграмах класів і компонентів для вказівки залежностей між ними і окремими класами і компонентами. Іноді на такий діаграмі можуть бути присутніми тільки пакети з залежностями між ними.

Фізичне представлення програмної системи не може бути повним, якщо відсутня інформація про тому, на якій платформі і на яких обчислювальних коштах вона реалізована. Якщо розробляється програма, що виконується локально на комп'ютері користувача і не використовує периферійних пристроїв і ресурсів, то в розробці додаткових діаграм немає необхідності. Під час розробки ж корпоративних додатків наявність діаграм розгортання може бути вкрай корисним для вирішення завдань раціонального розміщення компонентів з метою ефективного використання розподілених обчислювальних і комунікаційних ресурсів мережі, забезпечення безпеки та інших.

Діаграми розгортання показують декомпозицію системи на фізичні пристрої різних видів - сервери, робочі станції, термінали, принтери, маршрутизатори та ін. - і зв'язку між ними, представлені різного роду мережевими і індивідуальними сполуками. Тобто Діаграма розгортання відображає фізичні взаємозв'язку між програмними і апаратними компонентами системи [21].

Фізичні пристрої, які називаються вузлами системи, зображуються у вигляді кубів або паралелепіпедів, а фізичні з'єднання між ними - у вигляді ліній (рисунок 1.14).

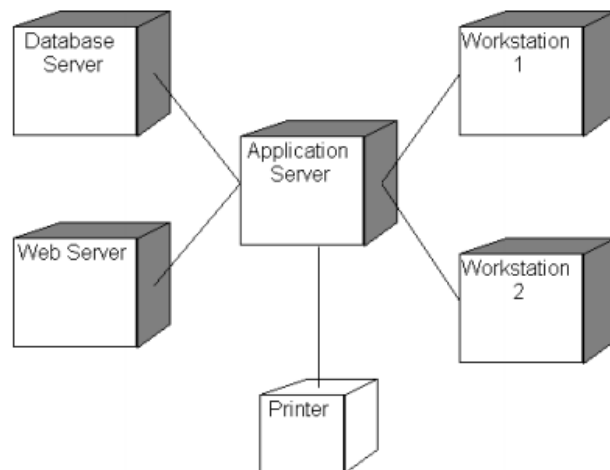


Рисунок 1.14 - Приклад діаграми розгортання

Вузли можуть мати підвузли, які представляються як вкладені прямокутні паралелепіпеди. Один вузол діаграми розгортання може концептуально представляти безліч фізичних вузлів, таких як кластер серверів баз даних.

Діаграми компонентів і розгортання використовуються досить рідко.

## 1.5 Особливості МСКУ-4.1

### 1.5.1 Призначення і загальна характеристика МСКУ-4

Мікропроцесорні керуючі обчислювальні комплекси МСКУ-4 - це сімейство проектно-компонованих, гнучко програмованих логічних промислових контролерів, призначених для вирішення широкого кола завдань в АСУ ТП і для застосування в якості:

- промислових контролерів відмовостійких систем автоматизації особливо відповідальних об'єктів (атомної і теплової енергетики, залізничного транспорту, нафтогазового комплексу та ін.),
- підсистем нижнього рівня АСУ ТП;
- інтелектуальних автономних систем контролю і управління [22].

Інформація, що міститься в позначенні мікропроцесорного керуючого обчислювального комплексу МСКУ-4, представлена на рис. 1.15.

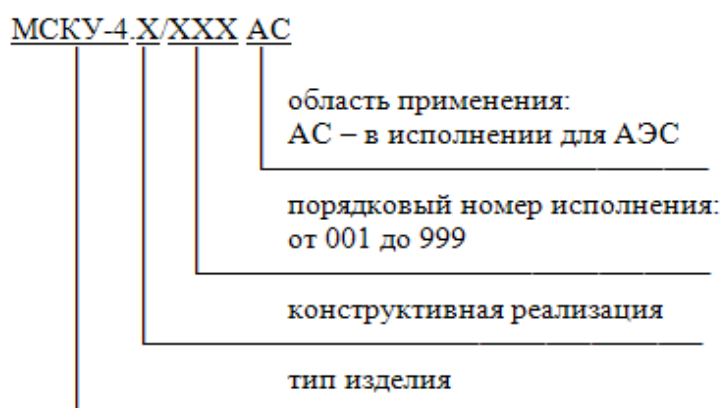


Рисунок 1.15- Позначення МСКУ-4

МСКУ-4 компонуються в підлогових шафах зі ступенем захисту IP21. Шафа, що має в своєму складі КМП, називається ведучим, який не має КМП - веденим (шафу розширення) [23].

Основними перевагами МСКУ-4 є:

- можливість компоновання структур як з одного, так і з декількох шаф з єдиною центральною частиною і великою кількістю (до 11 000) каналів зв'язку з об'єктом за допомогою високої продуктивності мікропроцесорного контролера (КМП);
- можливість резервування як центральної частини, так і каналів введення / виводу;

– більш гнучка адаптація до вимог замовника, за рахунок наявності трьох різних конструкцій шафи (конструкції МСКУ-4 представлені у таблиці 1.5).

Таблиця 1.5 - Конструкції МСКУ-4

Шифр	Ступінь захисту	конструкція
МСКУ-4.0/XXX	IP21	класична
МСКУ-4.1/XXX	IP21	модульна
МСКУ-4.2/XXX	IP21	крейт-крос

МСКУ-4 забезпечує виконання наступних основних функцій:

- збір, перетворення, первинна обробка та зберігання інформації, отриманої від об'єкта;
- формування сигналів і видача керуючих впливів на виконавчі механізми;
- реалізація алгоритмів контролю та управління, різних законів регулювання, захистів, блокувань, пуску і зупинки устаткування;
- взаємозв'язок із зовнішніми абонентами по інтерфейсах Ethernet, RS-422, RS-485 [22].

Далі пристрій і робота мікропроцесорного керуючого обчислювального комплексу буде розглядатися на прикладі модульної конструкції МСКУ-4.1.

### 1.5.2 Архітектура технічних засобів МСКУ-4.1

Складові частини МСКУ-4.1:

- контролери мікропроцесорні (КМП) на основі інтерфейсу PCIExpress. Кожен КМП оснащений двома портами оптичного інтерфейсу Ethernet, двома портами Ethernet і 13 портами внутрішнього радіального інтерфейсу ІРП-4. Кожен КМП може мати 4, 8 або 12 додаткових портів оптичного Ethernet при установці одного, двох або трьох модулів зв'язку МСВ-41 відповідно;
- модулі зв'язку з об'єктом (МСО), призначені для введення / виведення дискретних і аналогових сигналів, оснащені трьома портами ІРП-4;
- модулі зв'язку МСВ-42, призначені для зв'язку між оптичним Ethernet і 49 портами ІРП-4;



- модулі МСВ-43, призначені для зовнішніх зв'язків по інтерфейсах RS-422 і RS-485;
- модулі контролю обладнання МКО-4, призначені для контролю працездатності і стану обладнання, розміщеного в шафі МСКУ-4.1;
- панелі кросові (ПКР), призначені для підключення кабелів від об'єктів;
- панелі сполучні (ПКД), призначені для зв'язку ПКР з МСО;
- каркаси монтажні (КМ), призначені для установки КМП і МСО;

Склад конкретного МСКУ-4.1 визначається особливостями його застосування в системі автоматизації об'єкта [22].

Робота МСКУ-4.1 здійснюється під управлінням входить до його складу мікропроцесорного контролера КМП-30 і програмного забезпечення, записаного в FLASH-пам'ять КМП-30. Програмне забезпечення розробляється для кожного МСКУ-4.1, в залежності від функцій, які виконуються ним у складі програмно-технічного комплексу конкретної системи автоматизації.

Виходячи з призначення і виконуваних функцій, МСКУ-4.1 відносяться до технічних засобів автоматизації (ТЗА) серійного виробництва, які виготовляються для конкретних інформаційних управляючих систем (ІУС) або програмно-технічних комплексів.

МСКУ-4.1 по компоновці поділяються на «провідні» МСКУ-4.1, в складі яких є один або три КМП-30, і «ведені» МСКУ-4.1 з набором МСО, що працюють під управлінням «ведучого» МСКУ-4.1.

МСКУ-4.1 "провідний" складається з наступних частин:

- постійної (базової), що визначає конструктивну компоновку обладнання МСКУ-4.1 з необхідним захистом від впливу навколишнього середовища;
- змінної, що забезпечує об'єктну орієнтацію МСКУ-4.1, компонований з одного / трьох мікропроцесорних контролерів КМП-30 і відповідного набору функціональних модулів зв'язку з об'єктом, а також комплекту кросового обладнання та інших складових частин;
- комплекту контрольного і метрологічного обладнання, що враховує структуру і склад основного обладнання.

МСКУ-4.1 "ведений" складається з наступних частин:

- постійної (базової), що визначає конструктивну компоновку обладнання МСКУ-4.1 з необхідним захистом від впливу навколишнього середовища;

- змінної, що забезпечує об'єктну орієнтацію МСКУ-4.1, компонований відповідним набором функціональних модулів зв'язку з об'єктом, а також комплекту кросового обладнання та інших складових частин;
- комплекту контрольного і метрологічного обладнання, що враховує структуру і склад основного обладнання [23].

Модифікації шаф МСКУ-4.1 і їх відмінні риси [22] наведені в таблиці 1.6. Для підрахунку числа каналів введення / виводу слід кількість МСО помножити на 16.

Таблиця 1.6 - Модифікації шаф МСКУ-4.1 і їх відмінні риси

Тип шафи	Відмінні особливості				Кількість відомих шаф, шт.
	Кількість КМп-30, шт.	Кількість МСв-41, шт.	Кількість МСв-42, шт.	Кількість МСО, не більш, шт.	
ведучий	1	Від 1 до 3	-	48	12
	3	3 або 6 або 9		72	12
ведучий	-	-	1	48	-
	-	-	3	72	-

МСКУ-4.1 в залежності від конкретної конфігурації технічних і програмних засобів забезпечує виконання наступних основних функцій:

- введення і перетворення безперервних сигналів постійного струму і напруги низького і середнього рівня;
- введення і перетворення безперервних сигналів у вигляді мінливого активного опору;
- введення і перетворення безперервних сигналів у вигляді відносини активних опорів;
- введення дискретних сигналів у вигляді мінливого активного опору;
- введення дискретних сигналів у вигляді змінюється напруги змінного або постійного струму;
- обробка результатів введення і підготовка дій, що управляють;
- формування вихідних безперервних сигналів постійного струму;
- формування вихідних дискретних сигналів «замкнуто-розімкнуте»;
- організація двонапрямлених каналів зв'язку з інтерфейсом RS-485 / RS-422;

- зв'язок з технологічної ПЕОМ або ІДС по одному або двом інтерфейсів Ethernet 100BASE-FX, передачу інформації про поточні значення параметрів управління;
- самодіагностику з локалізацією несправності до змінного функціонального модуля.

У МСКУ-4.1 передбачена можливість компоновки як нерезервованих (один КМП), так і троїрованих (три КМП) виконань. З підключення вхідних та вихідних сигналів - передбачена можливість нерезервованої (до одного МСО), дубльованого (до двох МСО) і троїрованого (до трьох МСО) підключення, що працює з одним і тим же датчиком або виконавчим механізмом [23]. Схеми підключення датчиків до МСКУ-4.1 з трьома резервованими КМП наведені на рис. 1.16-1.18. Схеми підключення датчиків до МСКУ-4.1 з одним КМП наведені на рис. 1.19 і 1.20.

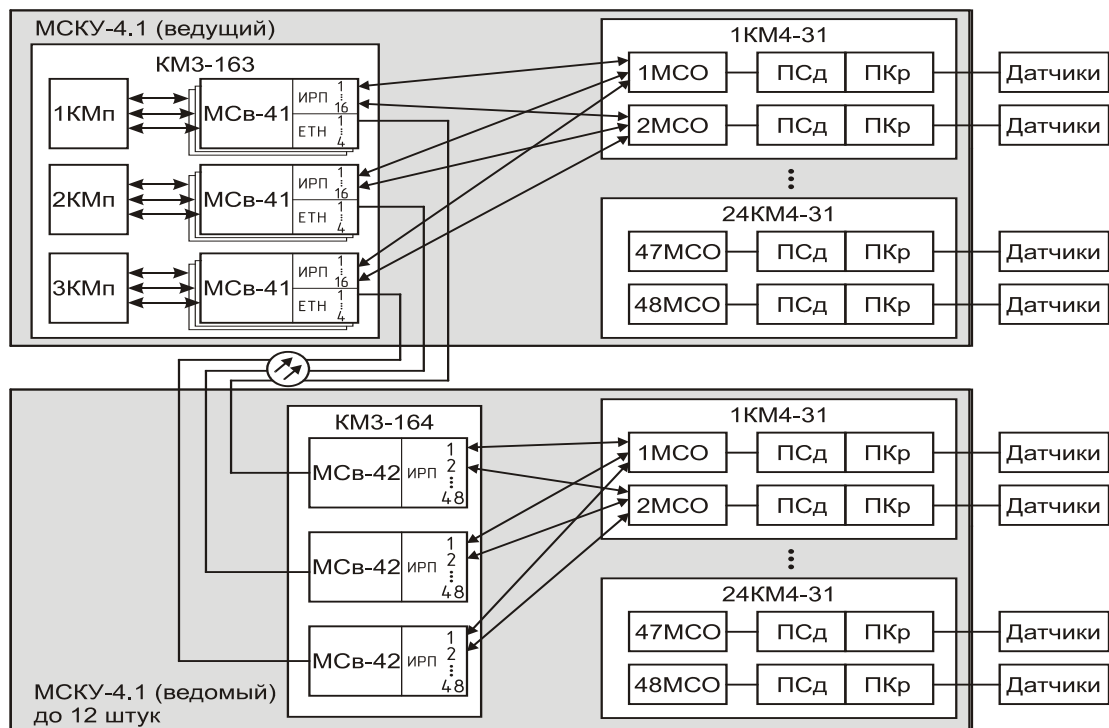


Рисунок 1.16 - Схема підключення нерезервованих МСО до трьох резервованих КМП

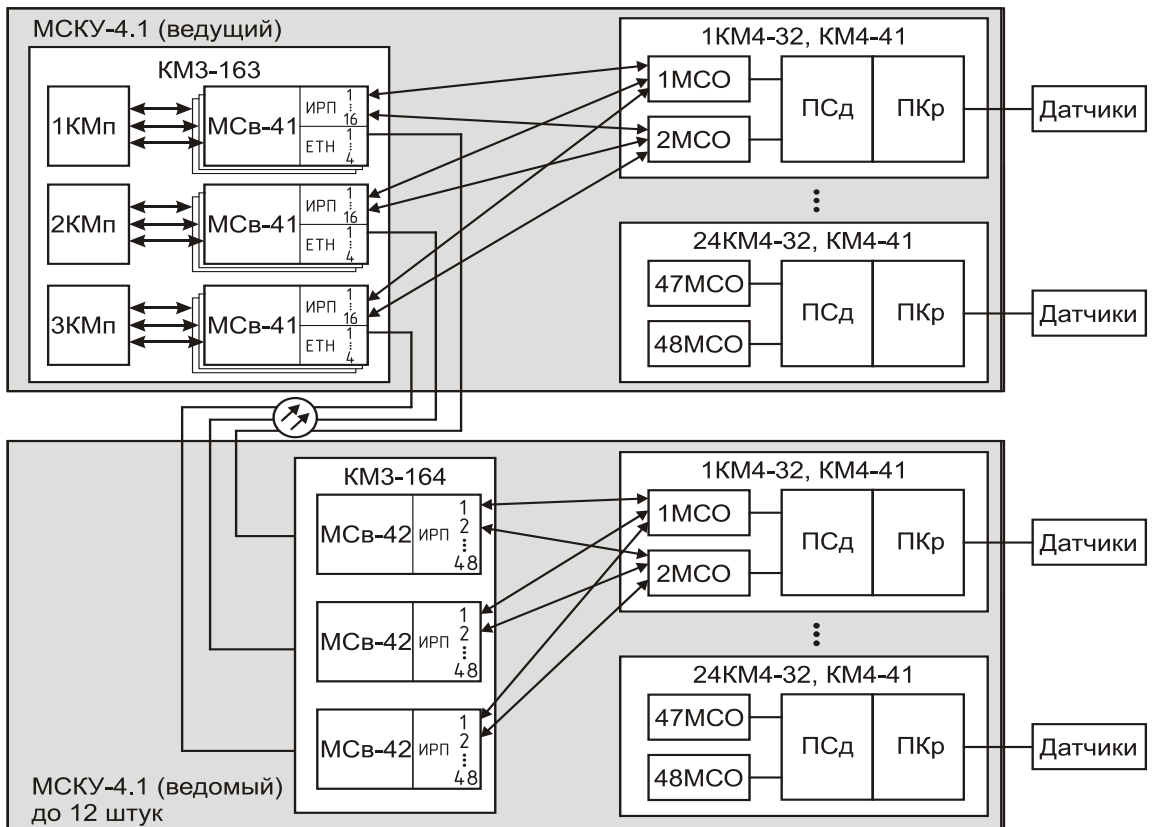


Рисунок 1.18 - Схема підключення трьох резервованих МСО до трьох резервованим КМп

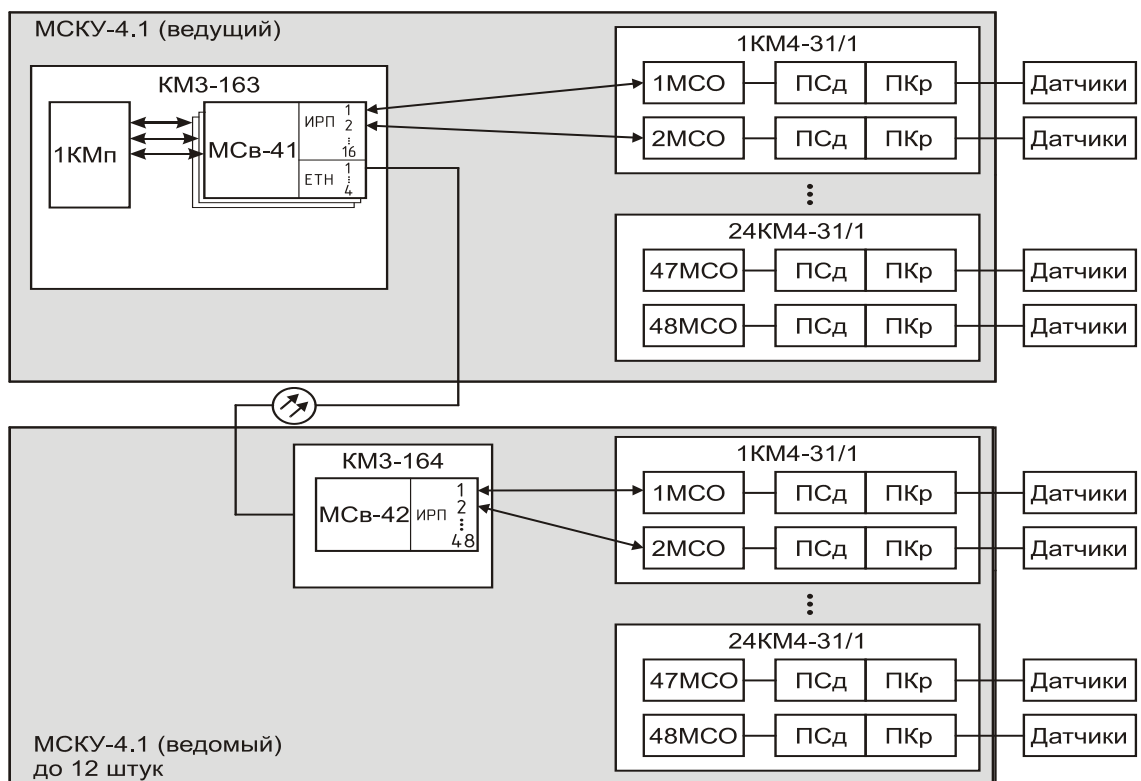


Рисунок 1.19 - Схема підключення нерезервованих МСО до нерезервованої КМп

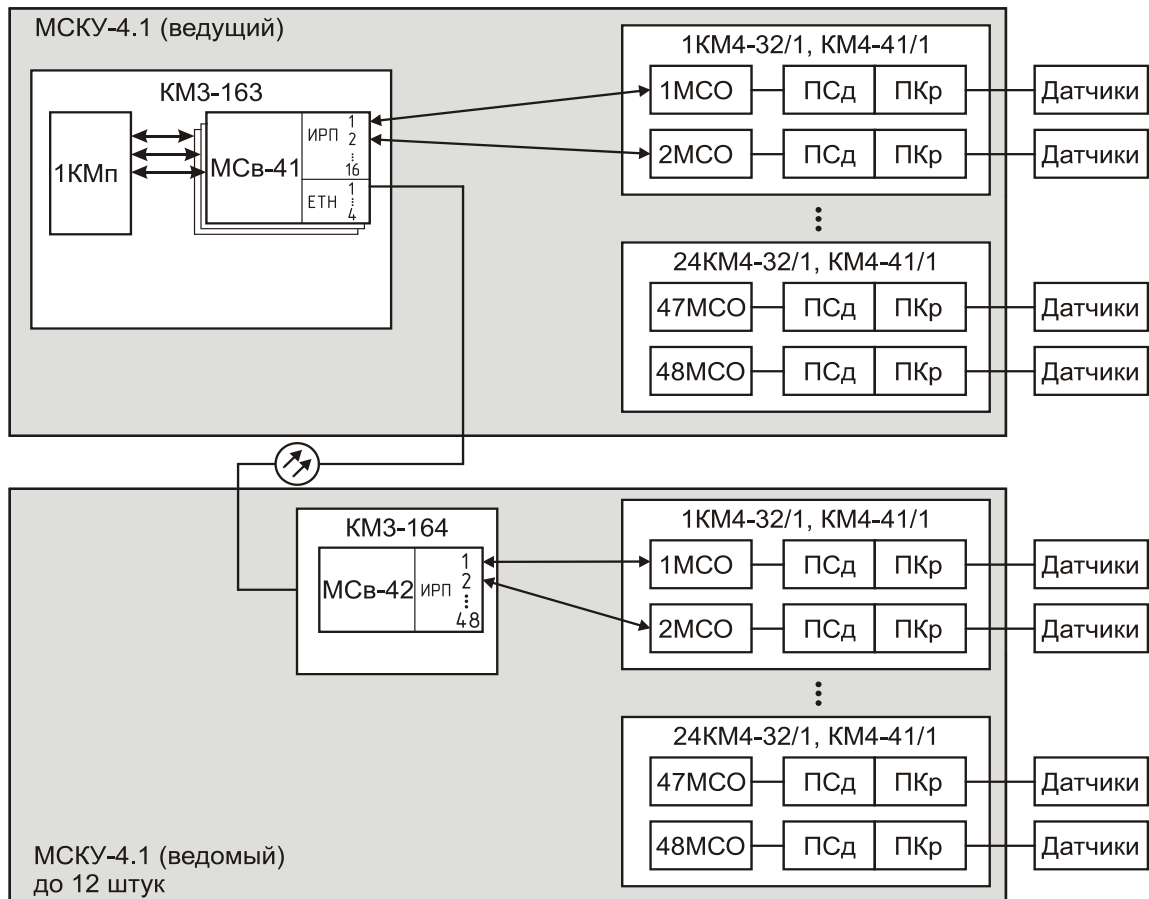


Рисунок 1.20 - Схема підключення двох резервованих МСО до нерезеровованої КМп

### 1.5.3 Організація каналів зв'язку з об'єктом

Введення даних від об'єкта управління і виведення керуючих впливів на об'єкт здійснюється в МСКУ-4.1 через функціональні МСО, ПКД та системні кабелі. Каркаси монтажні мають чотири місця під установку МСО і ПКД. МСО є основним елементом перетворення вхідних і вихідних сигналів МСКУ-4.1.

Кожен канал зв'язку з об'єктом створюється з наступних компонентів (в порядку проходження з боку підключення об'єктових кабелів):

- панель кросова ПКР;
- панель сполучна ПКД;
- системний кабель;
- МСО;
- КМП-30.

Підключення датчиків до МСО виконується через об'єктові кабелі, які підключаються до панелей кросовим ПКР, що встановлюються на ПКД. Зв'язок модулів з

крос обладнанням (ПКД) здійснюється за допомогою системних кабелів. Зовнішні види ПКР і ПКД наведені на рис. 1.21.

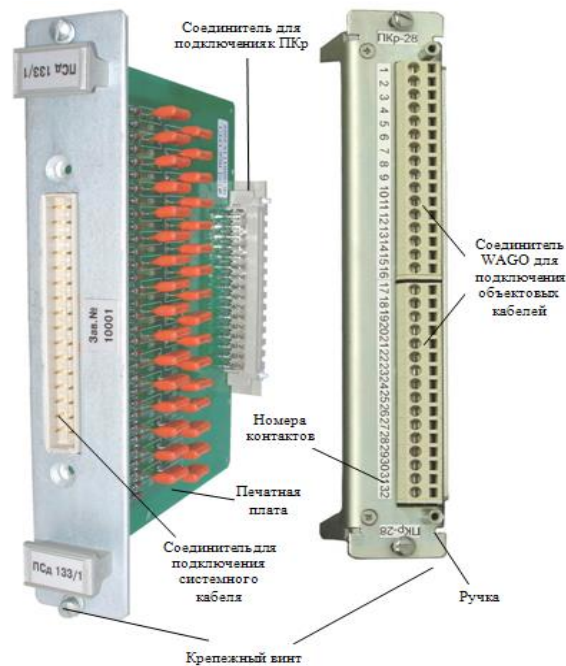


Рисунок 1.21 - Зовнішній вигляд сполучної ПКД та кросової ПКР панелей

ПКР виконана на друкованій платі з встановленими на ній соединителями фірми WAGO для приєднання жил виконано об'єктових кабелів і з'єднувача GdsA-F фірми Harting для під'єднання до ПКД. Одна ПКР і ПКД може прийняти до 16 двопровідних ліній зв'язку. Точки підключення до кожної ПКР промарковані на лицьовій панелі наскрізною нумерацією від 1 до 32 [23].

## 1.6 Постановка завдання

### 1.6.1 Опис предметної області

МСКУ-4.1 є проектно-компонований виробом. Це досягається за рахунок використання типових модулів зв'язку з об'єктом в різних комбінаціях під конкретні цілі. МСО призначені для введення / виведення дискретних і аналогових сигналів. МСО функціонують під управлінням контролера мікропроцесорного, в складі якого функціонує програмне забезпечення.

До складу МСКУ-4.1 також входять панелі кросові, призначені для підключення кабелів від об'єктів, панелі сполучні, призначені для зв'язку ПКР з МСО і каркаси монтажні, призначені для установки КМП і МСО.

Кожен канал зв'язку з об'єктом створюється з наступних компонентів (в порядку проходження з боку підключення об'єктових кабелів):

- панель кросова ПКР;
- панель сполучна ПКД;
- системний кабель;
- МСО;
- КМП-30.

МСО є основним елементом перетворення вхідних і вихідних сигналів МСКУ-4.1.

Зв'язок модулів з крос обладнанням (ПКД) здійснюється за допомогою системних кабелів по каналах. До кросової панелі (ПКР) підключення відбувається по контактам, полярність яких визначається конкретним типом МСО.

З розвитком архітектури МСКУ зросла кількість їх виконань і можливих місць для установки МСО. Для того щоб цими відомостями було зручно користуватися, їх необхідно систематизувати, тобто розбити на деякі категорії, які будуть забезпечувати найбільш повне і точне опис кожної складової бази даних. Основною метою роботи є створення бази даних та засобів діалогу з нею, які дозволять коректно проектувати шафи на базі МСКУ-4.1.

Програма необхідна для того, щоб зменшити кількість помилок при проектуванні шаф, що трапляються у зв'язку людського фактора, і повинна забезпечувати зручне формування, зміна, додавання і видалення даних з бази.

## **1.6.2 Технічне завдання на розробку автоматизованої системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1**

### **1.6.2.1 Вихідні дані для розробки автоматизованої системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1**

Система автоматизованої компоновки підсистем нижнього рівня технічних комплексів повинна розроблятися для конкретного ПТК на базі мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1.

Розробляється база даних повинна містити інформацію, необхідну для проектування ПТК на базі МСКУ-4.1. Ця інформація є вихідними даними для розробки автоматизованої системи. Також ці дані є статичними, т. К. Для кожного ПТК існує свій встановлений (фіксований) набір компонентів, необхідних для компоновки цього

виробу. Для МСКУ-4.1 до набору таких компонентів відносяться ПКР, ПКД, системні кабелі, МСО, КМП. Кожен з цих елементів (компонентів) виконує свою певну функцію.

Тому в розробляється БД повинна містити інформацію про:

- тип сигналу;
- типі МСО;
- типі ПКД;
- місці установки і номер каналу МСО;
- місці установки і номер каналу ПКД.

Також всі ці компоненти мають зв'язки один з одним в певному порядку, що дає можливість МСКУ-4.1 працювати і виконувати поставлені перед ним функції та завдання.

#### **1.6.2.2 Завдання системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1**

Основним завданням, яке повинна виконувати розробляється інформаційна система, є отримання таблиці. Ця таблиця має незмінну структуру і повинна наочно показувати взаємозв'язку між компонентами мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1.

Щоб отримати таку таблицю, вихідні дані необхідно обробити і перетворити за допомогою різних методів обробки даних (наприклад, SQL-запитів, макросів в MS Access і т. Д.). Тобто Необхідно перетворити надходить потік інформації в вихідний потік, оформлений згідно з властивими йому критеріями і правилами.

#### **1.6.2.3 Засоби реалізації**

Засобами реляційної СУБД MS Access повинна бути розроблена форма, за допомогою якої користувач зможе, вибираючи певні значення в полях форми, заносити в результуючу таблицю необхідні для компонування шаф дані про сигнали МСКУ-4.1.

Деякі поля на формі мають між собою залежність. Реалізація цих залежностей може бути здійснена за допомогою засобів MS Access (запитів, макросів), а також із застосуванням інтерпретується мови програмування Visual Basic for Applications.



## 2 ВИБІР МЕТОДУ ПРОЕКТУВАННЯ СИСТЕМИ КОМПОНУВАННЯ ПІДСИСТЕМ НИЖНЬОГО РІВНЯ ТЕХНІЧНОЇ КОМПЛЕКСУ МСКУ-4.1

### 2.1 Подібності структурного і об'єктно-орієнтованого підходів до проектування

Незважаючи на принципову методологічну різницю в розглянутих підходах до проектування, між ними багато спільного [24]. У таблиці 2.1 показані типові завдання, що виникають на етапі концептуального проектування ПО і відповідні методи їх вирішення в рамках розглянутих підходів.

Таблиця 2.1 - Завдання концептуального проектування ПО і методи їх вирішення

Типова задача проектування	методи рішення	
	структурний підхід	Об'єктно-орієнтований підхід
Функціональний розподіл на верхньому рівні абстракції	SADT (структура і ієрархія процесів)	UML-діаграми варіантів використання
Визначення типів предметів, взаємозв'язків між ними і їх властивостей	ERD (атрибути сутностей і зв'язку між ними)	UML-діаграма класів (класи об'єктів, їх атрибути і методи, зв'язку між ними)
Опис руху потоків даних	DFD (рух даних між процесами і сховищами)	UML-діаграма послідовностей (обмін повідомленнями між об'єктами різних класів)
Опис логічної послідовності виконання функцій бізнес-процесу	IDEF3 / PFDD (функції процесу, логічні оператори)	UML-діаграма діяльності
Опис зміни станів предмета	IDEF3 / OSTN	UML-діаграми станів

Аналіз таблиці 2.1 дозволяє зробити висновок про те, що найбільш популярні методи існуючих підходів до розробки ПЗ, мають деяку схожість між собою.

Наприклад, UML-діаграма класів об'єктно-орієнтованого підходу і ERD діаграма структурного підходу мають однакове призначення, яке полягає у визначенні статичної структури моделі системи. Ці діаграми необхідні, щоб описати об'єкти проекрованої системи, їх властивості та атрибути, а також взаємозв'язку між ними. Різниця полягає в тому, що діаграми класів об'єктно-орієнтованого підходу дозволяють показати також і операції, які можуть бути виконані над атрибутами.

Таким чином, можна зробити висновок, що обидва підходи, і структурний, і об'єктно-орієнтована, дозволяють виконати первинну типізацію предметної області з метою виділення основних понять, їх властивостей і відносин між ними, але більш наочною є UML-діаграми класів (в порівнянні з ERD діаграмами структурного підходу), т. к. вона враховує більше можливостей.

Проте, застосування тієї чи іншої діаграми залежить від цілей проектування і поставлених завдань. Наприклад, якщо при проектуванні необхідно вирішувати задачу розробки логічної і фізичної моделі бази даних, то доцільніше вибрати ERD діаграму структурного підходу.

Функціональний розподіл на верхньому рівні абстракції в структурному підході до проектування здійснюється за допомогою методології SADT, а в об'єктно-орієнтованому підході за допомогою UML-діаграм варіантів використання. Обидва підходи відображають функціональну структуру об'єкта, тобто послідовність дій, вироблених об'єктом, і зв'язку між цими діями. Фактично UML-діаграма варіантів використання є деяким аналогом методології SADT, де варіант використання замінює роботу, а актор - один з механізмів її виконання.

Для опису руху потоків даних структурний метод DFD є більш придатним в порівнянні з об'єктно-орієнтованою UML-діаграмою послідовностей. Діаграма DFD дозволяє описати рух і перетворення даних через бізнес-процеси, при цьому в діаграмі можна відобразити сховища даних. Також в цій діаграмі є можливість вказівки виконавців процесів, які можуть бути показані у вигляді зовнішніх сутностей. Діаграми послідовностей здійснюють обмін повідомленнями між об'єктами різних класів, показуючи, в якому саме порядку надсилаються ці повідомлення. Діаграма послідовності ідеальна для демонстрації впорядкованої передачі повідомлень під час взаємодії об'єктів.

Виходячи з цього, DFD діаграми мають суттєву перевагу по відношенню до діаграм послідовності, дозволяючи робити більш складне розгалуження, т. К. Формат і характер діаграм послідовностей дозволяють показувати тільки просте розгалуження.

Аналогічну ситуацію близькості структурного і об'єктно-орієнтованого підходів можна відзначити на прикладі схем IDEF3 / OSTN і UML-діаграм станів, призначених для

опису набору станів досліджуваного предмета і переходів між ними. Як і більшість методів для опису алгоритму дій і поведінки складних систем, в їх основі лежить теорія графів і її різні модифікації. IDEF3 / OSTN дозволяє чітко показати процес, внаслідок якого стан предмета змінюється, а в UML можна відобразити лише операції і умови переходу з одного стану предмета в інше. Це підтверджує той факт, що структурні методи в першу чергу орієнтовані на процеси, а не на їх оточення, на відміну від об'єктно-орієнтованих методів.

Ще одним загальним принципом побудови формальних моделей за допомогою графічних нотацій є дотримання ієрархічної декомпозиції. Так в об'єктно-орієнтованому підході за допомогою UML-діаграм можливе створення узагальненого стану проектованої системи, детальне розкриття якого виконується на окремій вкладеній діаграмі. Аналогічна каскадний декомпозиція процесів є основою відповідних структурних методів: від контекстної діаграми, яка показує проектовану систему на абстрактному рівні, до детального викладу послідовності дій на діаграмах декомпозиції (методології SADT).

На відміну від структурного методу в об'єктно-орієнтованому підході є UML-діаграми компонентів, які дозволяють визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами, і UML-діаграми розгортання, які відображають фізичні взаємозв'язку між програмними і апаратними (фізичні пристрої різних видів) компонентами системи. Ці діаграми відображають інший аспект представлення моделі, а саме фізичне уявлення, на відміну від усіх розглянутих раніше діаграм, які відображали концептуальні аспекти побудови моделі системи і ставилися до логічному рівню уявлення.

Особливість логічного уявлення полягає в тому, що воно оперує поняттями, які не мають самостійного матеріального втілення (класи, стани, повідомлення і т. Д.), ТОБТО Які не існують матеріально або фізично. Вони лише відображають розуміння структури фізичної системи або аспекти її поведінки. Основне призначення логічного представлення полягає в аналізі структурних і функціональних відносин між елементами моделі системи. Однак для створення конкретної фізичної системи необхідно певним чином реалізувати всі елементи логічного представлення в конкретні матеріальні сутності. Цим і займаються UML-діаграми компонентів і розгортання, хоча і використовуються досить рідко.

Підводячи підсумок опису найбільш поширених методів структурного і об'єктно-орієнтованого підходів і їх застосування в проектуванні, можна зробити висновок про те, що розглянуті методи успішно доповнюють один одного і можуть сукупно використовуватися для вирішення відповідних завдань.

## 2.2 Порівняльний аналіз структурного і об'єктно-орієнтованого підходів до проектування

Не дивлячись на те, що діаграми структурного і об'єктно-орієнтованого підходів мають деяку схожість між собою і спрямовані на виконання однакових завдань проектування, при розгляді підходів в цілому, вони мають ряд розбіжностей, які визначають переваги і недоліки кожного з методів. У таблиці 2.2 наведено порівняльний аналіз структурного і об'єктно-орієнтованого підходів до проектування.

Таблиця 2.2 - Порівняння структурного і об'єктно-орієнтованого підходів до проектування

Критерій оцінки	Структурний підхід	Об'єктно-орієнтований підхід
1	2	3
Принцип проектування інформаційної системи	Реалізація моделей за принципом "зверху-вниз", коли вся система в цілому слід розуміти як один великий функціональний блок, який може бути декомпозований на безліч підфункцій, які, в свою чергу, розбиваються на підфункції і т.д.	В першу чергу виділяються класи об'єктів, а далі в залежності від можливих станів об'єктів визначаються методи їх обробки (функціональні процедури).
Ступінь динамічності предметної області	Підходить для більш фіксованих, упорядкованих і регламентованих завдань.	Підходить для більш адаптивних завдань (наприклад, управління робочими потоками).
Наочність	Модель наочна і добре систематизована, т. К. Їй властива процедурна строгість декомпозиції ІС.	Модель менш наочна, т. К. Орієнтована на людське сприйняття світу, а не на комп'ютерну реалізацію (природна).

Продовження табл. 2.2.

1	2	3
Адаптивність системи до змін (можливість повторного використання моделі)	Як правило, модель не придатна для повторного використання.	Підвищено рівень приведення моделей до єдиної формі (уніфікація), отже, придатність для повторного використання вище. Даний підхід не дає негайної віддачі. Ефект від його застосування позначається після розробки двох-трьох проектів і накопичення повторно використовуваних компонентів, тому при одноразовому використанні моделі його краще не застосовувати.
Спосіб представлення даних	Предметна область розглядається як набір функцій, що перетворює надходить потік інформації в вихідний потік, де методи обробки даних чітко відокремлені від самих даних.	Предметна область розглядається як набір взаємодіючих об'єктів (предметів або явищ), що мають чітко визначається поведінка.
Рівень опису інформаційної системи	На рівні загального опису системи дана методика допускають значний ступінь сваволі у виборі загальних інтерфейсів системи, її механізмів і т.д., тобто у визначенні меж системи. Структурний підхід в цілому краще дає уявлення про існуючі функціях і методах їх реалізації.	Дозволяє добре описати систему на рівні загального опису системи, т. К. Даний підхід заснований на понятті сценарію використання. Ключовим є поняття про сценарії використання як про сеанс взаємодії дійової особи з системою, в результаті якого дійова особа отримує щось, що має для нього цінність.

За результатами порівняння структурного і об'єктно-орієнтованого підходів до проектування, наведених у таблиці 2.2, можна виділити основні позитивні і негативні сторони, які, безумовно, є в обох підходах (таблиця 2.3).

Таблиця 2.3 - Переваги і недоліки структурного і об'єктно-орієнтованого підходів до проектування

Підхід	Переваги	Недоліки
структурний	<p>1) Можливість проведення глибокого аналізу процесів, виявлення вузьких місць; комплексне застосування дозволяє виявити всі можливі неузгодженості та неточності.</p> <p>2) Застосування універсальних графічних мов моделювання SADT, DFD, ERD і IDEF3 забезпечує логічну цілісність і повноту опису, необхідних для досягнення точних і несуперечливих результатів.</p> <p>3) Висока наочність моделі на окремих рівнях декомпозиції.</p> <p>4) перевірений часом і широке поширення серед аналітиків і розробників.</p>	<p>1) Низька наочність для непідготовлених користувачів при збільшенні кількості рівнів представлення моделі.</p> <p>2) Скрутні модифікації моделей.</p> <p>3) Складність сприйняття ієрархічно впорядкованої інформації.</p> <p>4) Необхідність проходження жорсткої структури, яка не завжди необхідна.</p>
Об'єктно-орієнтована	<p>1) Можливість за допомогою методів класу відобразити поведінку сутностей реального світу.</p> <p>2) Системи більш гнучкі і відкриті, а також вони легше піддаються внесенню змін, оскільки їх конструкція базується на стійких проміжних формах. Це дає можливість системі розвиватися</p>	<p>1) Неможливість проведення детального аналізу процесів.</p> <p>2) Діаграми, що відображають специфіку об'єктного підходу, набагато менш наочні і погано зрозумілі непрофесіоналами.</p> <p>3) Неповнота і незавершеність деяких видів діаграм, можливість їх невірної інтерпретації.</p> <p>4) Підхід не дає негайної віддачі.</p>

	<p>поступово і не призводить до повної її переробки навіть у разі істотних змін вихідних вимог.</p> <p>3) Високий рівень уніфікації розробки і придатність для повторного використання моделі.</p>	
--	--	--

При виборі методу проектування в першу чергу необхідно керуватися завданнями його застосування і зручністю використання. Для більш систематизованого, точного і повного визначення властивостей і функцій ІС, а також методів обробки даних, які чітко відокремлені від самих даних, краще застосовувати структурні методи проектування. Якщо ж головним завданням проектування ІС є дослідження поведінки системи і способів взаємодії об'єктів в цій системі, то найбільш підходящим методом буде об'єктно-орієнтована.

### **2.3 Обґрунтування вибору методу проектування системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1**

Вибір методу проектування визначається цілями проекту і в значній мірі впливає на весь його подальший хід.

Для виконання даного проекту необхідна розробка бази даних, яка міститиме інформацію, необхідну для компонування шаф на базі МСКУ-4.1, а також формування таблиці, в якій на основі даних з БД повинні бути наочно продемонстровані взаємозв'язку між компонентами МСКУ-4.1.

Для кожного ПТК існує свій встановлений набір компонентів, необхідних для компонування цього виробу, тому зручніше розглядати дану предметну область, як набір функцій, що перетворюють надходить потік інформації в вихідний потік, де методи обробки даних чітко відокремлені від самих даних. Отже, завдання розбивається на два етапи.

Перший полягає в тому, щоб структурувати дані про МСКУ-4.1, тобто Знайти необхідну інформацію, обробити її і занести в БД. Другий етап пов'язаний з формуванням результуючої таблиці, де необхідно продемонструвати поведінку даних, тобто Логічну послідовність виконання дій і алгоритму виконання робіт.

Такий принцип проектування ІС відповідає принципу "зверху-вниз", коли великий функціональний блок може бути декомпозований на безліч підфункцій, що збігається з позицією структурного підходу.

Дана предметна область (розглянута в 1.6) не є динамічною, а має фіксований і упорядкований набір завдань, що також відповідає структурному підходу. Отже, найбільш підходящим методом для проектування автоматизованої системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1 є структурний метод.

Отже, ієрархічну впорядкованість виконання завдань можна реалізувати за допомогою SADT-діаграм, поведінка системи - за допомогою методології IDEF3, а структурування інформації в БД - за допомогою діаграм "сутність - зв'язок" (ERD).



## **3 РОЗРОБКА АВТОМАТИЗОВАНА СИСТЕМА КОМПОНУВАННЯ ПІДСИСТЕМ НИЖНЬОГО РІВНЯ КОМПЛЕКСУ МСКУ-4.1**

### **3.1 Проектування автоматизованої системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1**

Після порівняння структурного і об'єктно-орієнтованого підходів до проектування для розробки автоматизованої системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1 був обраний структурний підхід.

Для автоматизації процесу побудови діаграм, що дозволяють в наочній формі моделювати предметну область, аналізувати цю модель на всіх стадіях розробки і супроводу, використовуються програмно-технологічні засоби, які отримали назву CASE-засобів (Computer Aided Software / System Engineering- комп'ютерна підтримка розробки програмного забезпечення систем ), що реалізують CASE-технологію створення і супроводу інформаційних систем [18].

CASE-технологія являє собою сукупність методологій аналізу, проектування, розробки і супроводу складних систем ПО. CASE - це інструментарій для системних аналітиків, розробників і програмістів, який замінює їм папір і олівець.

Застосування CASE-продуктів не тільки сприяє підвищенню продуктивності праці на всіх етапах створення ПЗ, але також зменшує ймовірність дефектів, викликаних помилками розробників, і дозволяє своєчасно виявляти ці дефекти, підвищуючи тим самим Гарантоздатні розроблюваного програмного забезпечення.

Більшість існуючих CASE-засобів засновано на методах як структурного, так і об'єктно-орієнтованого аналізу і проектування, але всі вони засновані на наочних діаграмних техніках, які використовуються для опису зовнішніх вимог, зв'язків між моделями системи, динаміки поведінки системи та архітектури програмних засобів.

Для проектування інформаційної системи були обрані структурні методології. Вони забезпечують строгий і наочний опис проектованої системи, яке починається з її загального огляду і потім деталізується, набуваючи ієрархічну структуру з дедалі більшим числом рівнів.

Реалізація методології структурного моделювання здійснюється за допомогою програмних засобів CA ERWin Process Modeler (BRWin) і CA ERWin Data Modeler (ERWin), що входять в лінійку CASE-продуктів AllFusion Modeler Suite (Computer Associates International, Inc.). Вони являють собою засіб для збору всієї необхідної

інформації про деяку систему і графічного зображення цієї інформації у вигляді цілісної і несуперечливої моделі [17].

BPWin підтримує відразу три стандартні нотації, які дозволяють комплексно описувати предметну область:

- SADT / IDEF0 (функціональне моделювання);
- DFD (моделювання потоків даних);
- IDEF3 (моделювання потоків робіт).

ERwin призначений для проектування і документування баз даних. Моделі даних допомагають візуалізувати і наочно відображати складні структури даних. ERwin підтримує нотацію ERD / IDEF1X, яка призначена для розробки моделей даних і забезпечує стандартний спосіб визначення даних і відносин між ними. Зручна у використанні графічне середовище ERwin спрощує розробку бази даних і автоматизує безліч трудомістких завдань, зменшуючи терміни створення високоякісних і високопродуктивних транзакційних баз даних і сховищ даних.

На рис. 3.1 - 3.4 представлені діаграми, розроблені за допомогою CASE-засобу BPWin і відображають функціональну структуру системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1.

Контекстна діаграма (рисунок 3.1) служать для опису взаємодії системи (в даному випадку роботи «Формування таблиці" Сигнали МСКУ ") з навколишнім світом. Реалізація цього процесу проводиться за допомогою стрілок.

Діаграми декомпозиції першого і другого рівня (малюнки 3.2 і 3.3) дозволяють деталізувати елементи SADT-діаграми, представлені у вигляді робіт, які взаємодіють між собою, а також показують інформаційні, людські та виробничі ресурси, що споживаються кожною роботою.

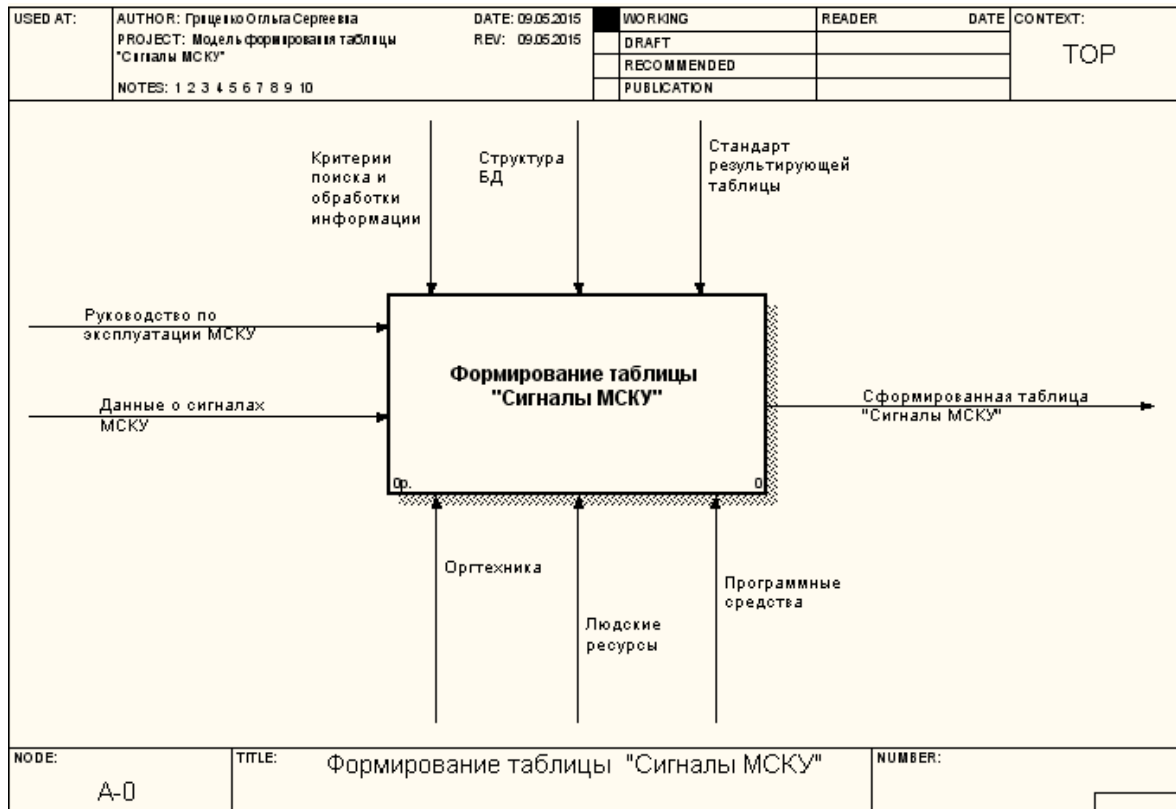


Рисунок 3.1 - Контекстна діаграма

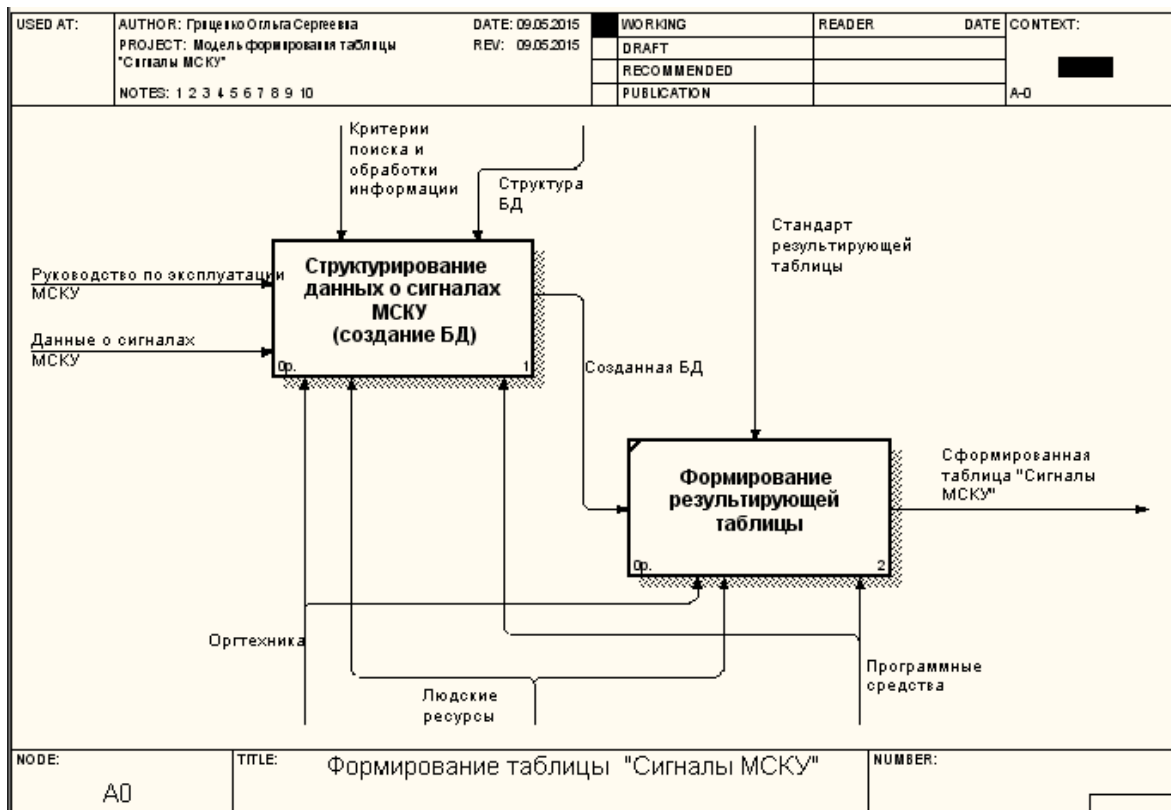


Рисунок 3.2 - SADT-діаграма декомпозиції роботи «Формування таблиці» Сигнали МСКУ

"»

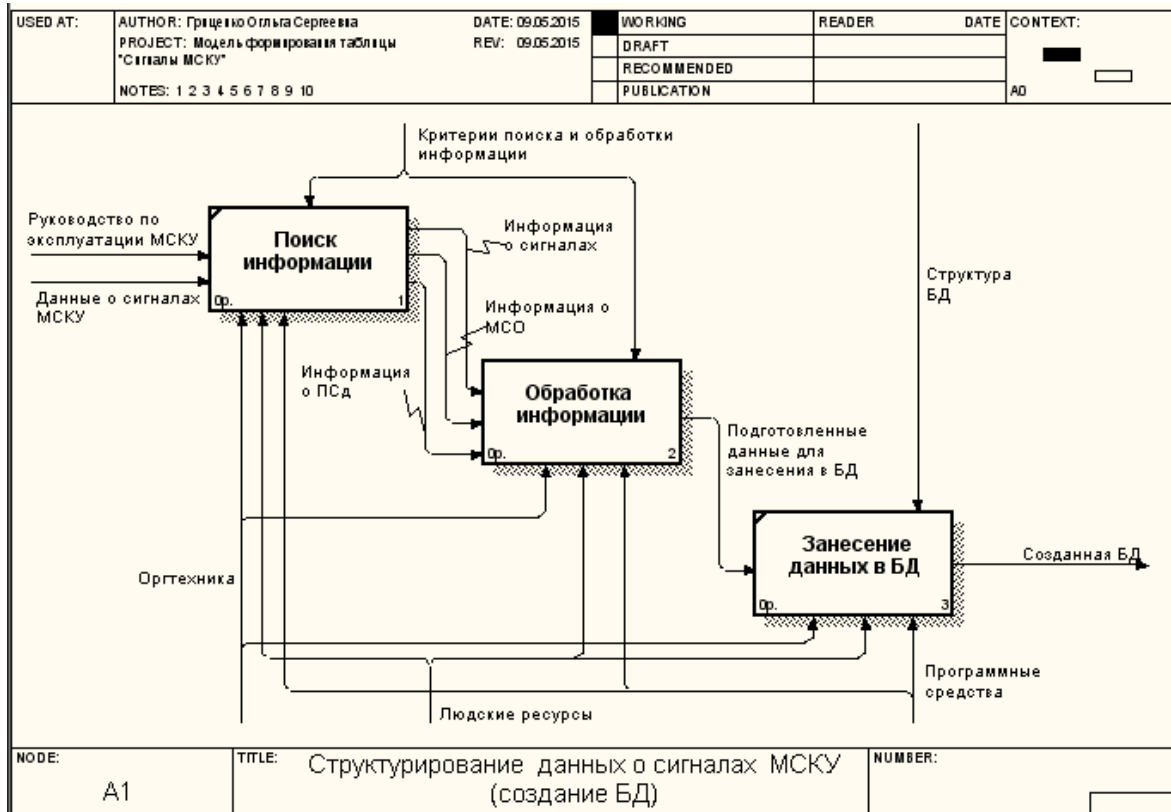


Рисунок 3.3 - SADT-діаграма декомпозиції другого рівня «Структурні дані про сигнали МСКУ (створення БД)»

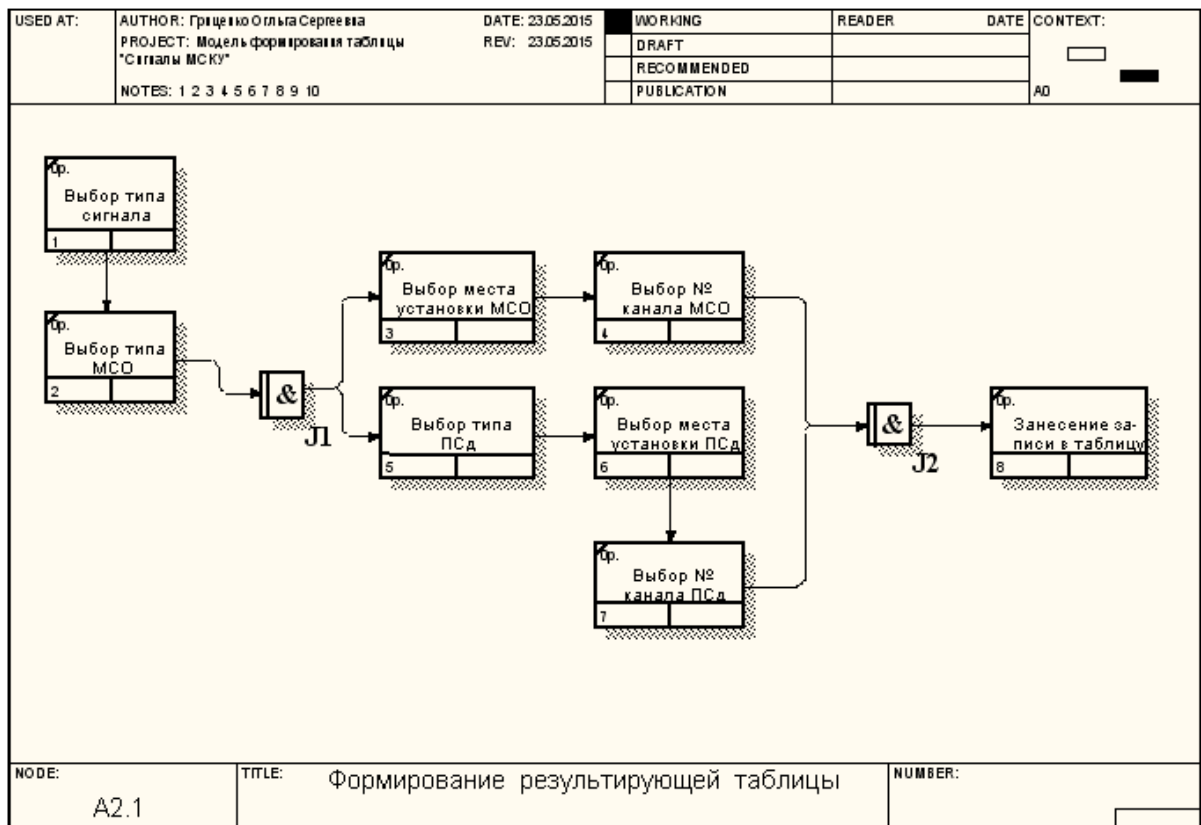


Рисунок 3.4 - IDEF3-діаграма декомпозиції другого рівня роботи «Формування результируючої таблиці»

Діаграма IDEF3 (рисунок 3.4) дозволяє отримати більш точну картину моделі, т. К. Вона розглядає предметну область з точки зору послідовності виконуваних робіт (поведінки системи). Цей метод привертає увагу до черговості виконання подій. У IDEF3 включені елементи логіки, що дозволяє моделювати і аналізувати альтернативні сценарії розвитку процесу.

На рис. 3.5 і 3.6 представлені діаграми, розроблені за допомогою CASE-засобу ERwin і відображають інформаційну модель даних системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1 і їх взаємозв'язку.

Опису предметної області, представленого в пункті 2.6.1 досить для того, щоб побудувати модель даних верхнього рівня - рівня сутностей (рисунок 3.5). Метою цієї діаграми є формування загального погляду на систему для її подальшої деталізації.

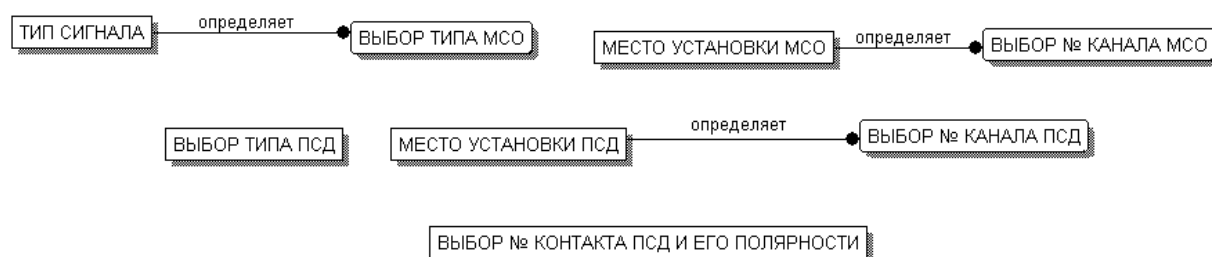


Рисунок 3.5 - ERD / IDEF1X рівня сутностей

Вибір унікальних імен первинних ключів сутностей в межах діаграми полегшує створення нормалізованої моделі даних. Первинні ключі сутностей представлені в таблиці 3.1.

Таблиця 3.1 - Первинні ключі сутностей

Сутність	Первинний ключ
Тип сигналу	Тип сигналу
Вибір типу МСО	Тип сигналу, тип модуля
Місце установки МСО	Місце модуля
Вибір № каналу МСО	Місце модуля, № каналу модуля
Вибір типу ПСд	Тип модуля, Тип ПСд
Місце установки ПСд	Місце ПСд
Вибір № каналу ПСд	Місце ПСд, № каналу ПСд
Вибір № контакту ПСд і його полярності	Тип МСО, тип ПСд, номер каналу ПСд, номер контакту ПСд

ERD рівня атрибутів наведена на рис. 3.6.

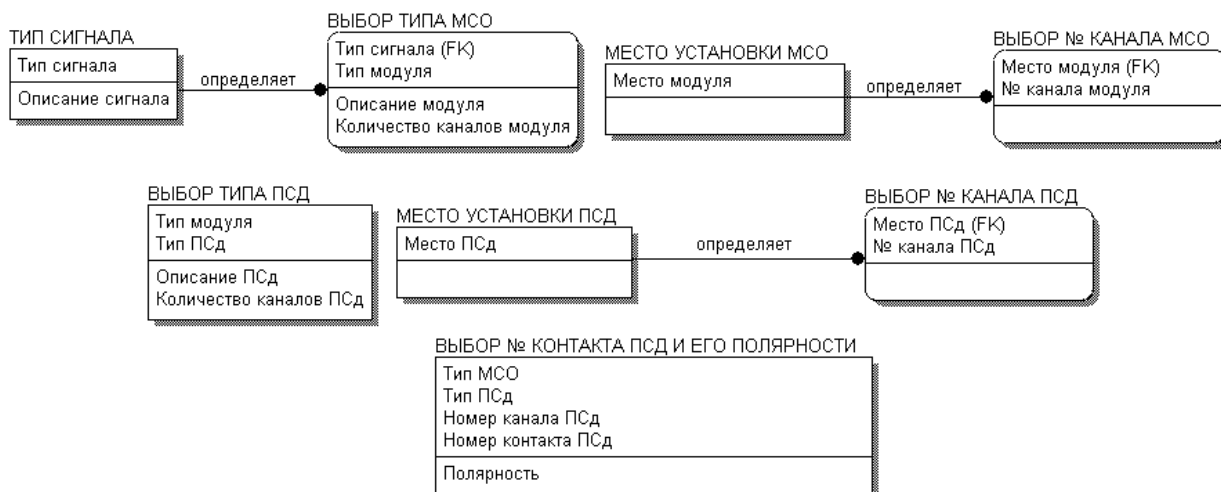


Рисунок 3.6 - ERD / IDEF1X рівня атрибутів

### 3.2 Створення бази даних

База даних є основним компонентом проекту програми Microsoft Access і може включати до свого складу таблиці, форми, запити, звіти, макроси, модулі. Для реалізації БД необхідно створити макети таблиць і схему даних. Цього можна уникнути, імпортувавши розроблену в пункті 3.1 інформаційну модель даних системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1 за допомогою CASE-засобу ERwin в реляційну систему управління базами даних MS Access. Отримана схема даних в MS Access представлена на рис. 3.7.

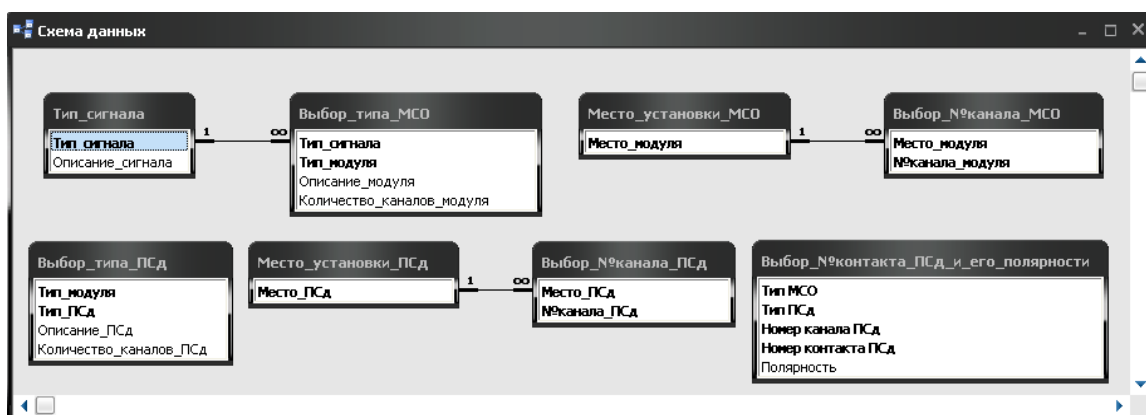


Рисунок 3.7- Схема даних

Приклад заповненої даними таблиці «Вибір типу МСО» наведено на рис. 3.8.

Тип сигнала	Тип модуля	Описание модуля	Количество каналов модуля
±0,5 μA	ПКИ-41	Г/развязка - групповая.	7
±80 mV	ПНК-42	Пассив. прием ±10 mV, ±20 mV, ±40 mV, ±80 mV, Iвх ≤ 0,1 nA.	16
0-10 V	ПНК-41/1	Пассив. прием 0-2,5 V, 0-10 V, Iвх ≤ 1 μA	16
0-100-500 Ω	ПСК-41/1	Пассив. прием 0-100-300 Ω, 0-100-500 Ω, 4-х провод. подключение.	8
0-100-500 Ω	ПСК-41/2	Пассив. прием 0-100-300 Ω, 0-100-500 Ω, 3-х провод. подключение.	8
0-2*10-5 1/Ω	ПКИ-41	Г/развязка - групповая.	14
0-20 mA	ПНК-41/2	Пассив. прием 0-5 mA, 0-20 mA, Rвх ≤ 50 Ω.	16
0-20 mA	ПТК-41/1	Актив. прием ((10-13) V), резервированный.	16
0-20 mA	ПТК-41/2	Актив. прием ((20-24) V), нерезервированный.	16
0-20 mA	ФТ-41/1	Вых. ток, Rнагруз. ≤ 50 Ω, резервированный.	16
0-20 mA	ФТ-41/2	Вых. ток, Rнагруз. ≤ 50 Ω, нерезервированный.	16
0-5 μA	ПКИ-41	Г/развязка - групповая.	7
0-5 mA	ПТК-42	Актив. прием ((35,28 - 36,72) V).	8
0-50-250 Ω	ПСК-41/1	Пассив. прием 0-50-150 Ω, 0-50-250 Ω, 4-х провод. подключение.	8
0-50-250 Ω	ПСК-41/2	Пассив. прием 0-50-150 Ω, 0-50-250 Ω, 3-х провод. подключение.	8
100-5000 Ω	ПСР-41/1	4-х провод. подключение.	8
100-5000 Ω	ПСР-41/2	3-х провод. подключение.	8
45-55 Hz	НД-43		4
RS-477	МСР-43/1	I	8

Рисунок 3.8 - Таблица «Вибір типу МСО»

Аналогічним чином були імпортовані і заповнені даними таблиці: «Сигнал», «Місце установки МСО», «Вибір номера каналу МСО», «Вибір типу ПКД», «Місце установки ПКД», «Вибір номера каналу ПКД», «Вибір номера контакту ПКД і його полярності».

Таблиці, представлені на схемі даних (малюнок 3.7), є вихідними даними для формування таблиць «Сигнали МСКУ» і «Сигнали МСКУ 2», так як містять інформацію, необхідну для проектування шаф на базі мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1.

Засобами реляційної СУБД MS Access була розроблена форма, представлена на рис. 3.9, за допомогою якої користувач зможе, вибираючи певні значення в полях форми, заносити в результуючу таблицю «Сигнали МСКУ» необхідні для компонування шаф дані про сигнали МСКУ-4.1.

Деякі поля на формі мають між собою залежність. Реалізація цих залежностей здійснюється за допомогою запитів на отримання даних з таблиць, написаних на структурованому мовою запитів SQL.

SQL (Structured Query Language) - формальний непроцедурного мова програмування, застосовуваний для створення, модифікації та управління даними в довільній реляційної базі даних, керованої відповідної СУБД. SQL ґрунтується на обчисленні кортежів.

Рисунок 3.9 – Форма

Запит «Вибір типу модуля» (малюнок 3.10) дозволяє користувачеві при виборі зі списку «Тип сигналу» на формі одного значення типу сигналу отримати всі відповідні йому типи МСО з описом. Так як виникає необхідність фільтрувати запит по параметру, введеному в поле форми, посилатися на поле форми можна за допомогою конструкції: [Forms]! [Імя\_Форми]! [Ім'я\_поля].

```

Выбор_типа_модуля_запрос : запрос на выборку
SELECT Выбор_типа_МСО.Тип_модуля, Выбор_типа_МСО.Описание_модуля
FROM Выбор_типа_МСО
WHERE ((Выбор_типа_МСО.Тип_сигнала)=Forms![Сигналы МСКУ]!fieldID1));

```

Рисунок 3.10 - Запит «Вибір типу модуля»

Результат виконання запиту виводиться в випадяючому списку «Тип МСО» на формі (рисунок 3.11).



**Сигналы МСКУ**

Наименование сигнала		
Тип сигнала	СК	
Тип МСО		
Место установки МСО	НК-41	НС ≤ 80 Ω, I <sub>опр</sub> =15 mA. NO (резистор 5,1 kΩ). Кл. ▲
Номер канала МСО	НК-42	НС ≤ 80 Ω, I <sub>опр</sub> =25 mA. NO (резистор 5,1 kΩ). Кл. □
Тип ПСд	НК-43	Общ "+". НС (резистор 510 Ω), I <sub>опр</sub> =7 mA. NO (р
Место установки ПСд	НК-43/1	Общ "-". НС (резистор 510 Ω), I <sub>опр</sub> =7 mA. NO (р
Номер канала ПСд	ФДС-41	Г/развязка - групповая. Уком – 187-242 V. I <sub>ком</sub> ≤ .
Примечание	ФДС-41/1	Г/развязка - групповая. Уком – 187-242 V. I <sub>ком</sub> ≤ .
	ФДС-42	Общ "-". Уком – 12-36 V. I <sub>ком</sub> ≤ 0,4 А
	ФДС-42/1	Общ "+". Уком – 12-36 V. I <sub>ком</sub> ≤ 0,4 А

Рисунок 3.11 - Результат выполнения запроса «Выбор типа модуля»

Реализация зависимостей інших полів на формі, здійснених за допомогою запитів, наведена в додатку А.

Форма містить 5 кнопок, кожна з яких виконує певну дію.

При натисканні кнопки «Занести в таблицю», всі дані обрані в полях форми заносяться до відповідних їм поля таблиці «Сигналы МСКУ».

При натисканні кнопки «Копія останнього рядка», копіюються всі дані останнього рядка таблиці «Сигналы МСКУ», крім місця установки МСО, ПКД та номера каналу МСО, ПКД, і заносяться у відповідні поля на формі. Кнопку зручно використовувати, якщо подальша запис в таблицю «Сигналы МСКУ» має не значні відмінності від попереднього запису (в полях «Місце установки МСО», «Місце установки ПКД», «Номер каналу МСО» і «Номер каналу ПКД»). Копія даних здійснюється за запитом «Копія останнього рядка таблиці», представленою на рис. 3.12. Результат натискання кнопки «Копія останнього рядка» представлено на рис. 3.13.

```

Копия последней строки таблицы : запрос на добавление
INSERT INTO [Сигналы МСКУ] ( [Наименование сигнала], [Тип сигнала], [Тип МСО], [Тип ПСд], Примечание )
SELECT [Сигналы МСКУ].[Наименование сигнала], [Сигналы МСКУ].[Тип сигнала], [Сигналы МСКУ].[Тип МСО],
[Сигналы МСКУ].[Тип ПСд], [Сигналы МСКУ].Примечание
FROM [Сигналы МСКУ]
WHERE ((([Сигналы МСКУ].id)=(SELECT MAX([Сигналы МСКУ].[id]) FROM [Сигналы МСКУ])););

```

Рисунок 3.12 - Запит «Копія останнього рядка таблиці»

**Сигналы МСКУ**

Наименование сигнала: Состояние контактора К1

Тип сигнала:  $\pm 0,5 \mu\text{A}$

Тип МСО: ПКИ-41

Место установки МСО: Занятые места установки: 4

Номер канала МСО:

Тип ПСд: ПСд-180

Место установки ПСд: Занятые места установки: 8

Номер канала ПСд:

Примечание:

Занести в таблицу      Копия последней строки      Открыть таблицу

Очистить таблицу      Добавление контактов

Рисунок 3.13 - Результат нажатия кнопки «Копія останнього рядка»

При нажатии кнопки «Відкрити таблицю», можна переглянути всі записи, які вже занесені в таблицю «Сигналы МСКУ». На рисунке 3.14 наведено можливий варіант таблиці «Сигналы МСКУ», отриманий в результаті заповнення полів форми.

Наименование сигнала	Тип сигнала	Тип МСО	Место установки МСО	Номер канала МСО	Тип ПСд	Место установки ПСд	Номер канала ПСд	Примечание
Сигнал "Неисправность СГИУ"	$\pm 0,5 \mu\text{A}$	ПКИ-41	1		1 ПСд-180	1		1
Сигнал "Неисправность СГИУ"	$\pm 0,5 \mu\text{A}$	ПКИ-41	1		2 ПСд-180	1		6
Обобщенный сигнал АЗ от ключа БЩУ	0-20 mA	ПТК-41/1	3		14 ПСд-202	5		14
Взвод АЗ с БЩУ	0-20 mA	ПТК-41/1	2		5 ПСд-202	5		5
Состояние контактора К1	$\pm 0,5 \mu\text{A}$	ПКИ-41	4		3 ПСд-180	8		3

Рисунок 3.14 - Таблица «Сигналы МСКУ»

При нажатии кнопки «Очистити таблицю», весь вміст таблиці «Сигналы МСКУ» видаляється. Видалення здійснюється за допомогою запиту «Очистити таблицю», представленого на рис. 3.15.

```
Очистить_таблицу : запрос на удаление
DELETE *
FROM [Сигналы МСКУ];
```

Рисунок 3.15 - Запит «Очистити таблицю»

При нажатии кнопки «Додавання контактів», формується таблиця «Сигналы МСКУ 2», в яку до вже існуючої таблиці «Сигналы МСКУ» додається два поля - «Номер

контакту ПКД» і «Полярність». Нова таблиця «Сигнали МСКУ 2» створюється за допомогою запиту «Створення нової таблиці з контактами», представленого на рис. 3.16.

```

Создание_новой_таблицы_с_контактами : запрос на создание таблицы
SELECT [Сигналы МСКУ].[Наименование сигнала], [Сигналы МСКУ].[Тип сигнала], [Сигналы МСКУ].[Тип МСО],
[Сигналы МСКУ].[Место установки МСО], [Сигналы МСКУ].[Номер канала МСО], [Сигналы МСКУ].[Тип ПСд],
[Сигналы МСКУ].[Место установки ПСд], [Сигналы МСКУ].[Номер канала ПСд],
Выбор_№контакта_ПСд_и_его_полярности.[Номер контакта ПСд], Выбор_№контакта_ПСд_и_его_полярности.Полярность,
[Сигналы МСКУ].Примечание INTO [Сигналы МСКУ 2]
FROM [Сигналы МСКУ], Выбор_№контакта_ПСд_и_его_полярности
WHERE ((([Сигналы МСКУ].[Тип ПСд]=Выбор_№контакта_ПСд_и_его_полярности.[Тип ПСд]) And
([Сигналы МСКУ].[Тип МСО]=Выбор_№контакта_ПСд_и_его_полярности.[Тип МСО]) And
([Сигналы МСКУ].[Номер канала ПСд]=Выбор_№контакта_ПСд_и_его_полярности.[Номер канала ПСд]));

```

Рисунок 3.16 - Запит «Створення нової таблиці з контактами»

Так як одному номеру каналу ПКД відповідає кілька контактів, то кожен рядок таблиці «Сигнали МСКУ» буде дублюватися стільки раз, скільки контактів відповідає каналу ПКД в цьому рядку. Результат записується в таблицю «Сигнали МСКУ 2» (малюнок 3.17).

Наименование сигнала	Тип сигнала	Тип МСО	Место установки МСО	Номер канала МСО	Тип ПСд	Место установки ПСд	Номер канала ПСд	Номер контакта ПСд	Полярность	Примечание
Сигнал "Неисправность СГИУ"	±0,5 µA	ПКИ-41	1		1 ПСд-180	1		1	-	
Сигнал "Неисправность СГИУ"	±0,5 µA	ПКИ-41	1		1 ПСд-180	1		1	2 +	
Сигнал "Неисправность СГИУ"	±0,5 µA	ПКИ-41	1		2 ПСд-180	1		6	11 -	
Сигнал "Неисправность СГИУ"	±0,5 µA	ПКИ-41	1		2 ПСд-180	1		6	12 +	
Обобщенный сигнал АЗ от клс 0-20 mA		ПТК-41/1	3		14 ПСд-202	5		14	27 +	
Обобщенный сигнал АЗ от клс 0-20 mA		ПТК-41/1	3		14 ПСд-202	5		14	28 -	
Взвод АЗ с БЩУ	0-20 mA	ПТК-41/1	2		5 ПСд-202	5		5	9 +	
Взвод АЗ с БЩУ	0-20 mA	ПТК-41/1	2		5 ПСд-202	5		5	10 -	
Состояние контактора К1	±0,5 µA	ПКИ-41	4		3 ПСд-180	8		3	5 -	
Состояние контактора К1	±0,5 µA	ПКИ-41	4		3 ПСд-180	8		3	6 +	

Рисунок 3.17 - Таблица «Сигнали МСКУ 2»

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення. Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [36] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу

слід приділяти біологічним ефектам низькочастотних електромагнітних полів(аж до порушення ДНК).

Відповідно до [37], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

#### **4.2 Заходи щодо техніки безпеки**

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної(нагрів тканин і біологічних середовищ), електролітичної(розкладання крові і плазми) і біологічної(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;

– індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [38]:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення ятерів;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де  $U_\phi$  - номінальна фазна напруга мережі, В;

$Z_\Pi$  - повний опір петлі, створене фазними і нульовими дротами, Ом;

$Z_T$  - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [39]:  $Z_T / 3 = 0,1$  Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де  $R_\Pi = R_\phi + R_0$  - сумарний активний опір фазного  $R_\phi$  і нульового  $R_0$  дротів, Ом;

$X_\Pi$  - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту  $S = 2.5$  мм, тоді згідно таблицям 5 і 6 [39], має такий опір:

$X_\Pi = 0,11$  Ом;

$R_\phi = 7,55$  Ом;

$R_0 = 7,55 \text{ Ом}$ .

Отже,  $R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}$ .

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де  $I_n$  - номінальний струм спрацьовування плавкої вставки, А;

$k$  - коефіцієнт кратності нелінійного струму  $I_n$ , А.

Коефіцієнт кратності нелінійного струму  $I_n$  розраховується по формулі (4.4.) :

$$I_n = P / U, \quad (4.4)$$

де  $P = 220 \text{ Вт}$  - споживана потужність;

$U = 220 \text{ В}$  - робоча напруга;

$k = 3 \text{ А}$  - для плавких вставок.

Отже,  $I_n = 220 / 220 = 1 \text{ А}$ .

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

### 4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [40], СНіП, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [41] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [42] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 м<sup>3</sup> в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для



оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплєну).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [43].

Стандарт [43] пред'являє жорсткі вимоги в таких областях: ергономіка(фізична, візуальна і зручність користування), енергія, випромінювання(електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [44].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [45] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛПО2П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки  $a=12$  м, шириною  $b=10$  м, заввишки  $z=4$  м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де  $F$  - світловий потік = 3120 лм;

$E$  - максимально допустима освітленість робочих поверхонь = 200 лк;

$S$  - площа підлоги = 120 м<sup>2</sup>;

$Z$  - поправочний коефіцієнт світильника = 1,2;

$k$  - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

$n$  - кількість світильників;

$U$  - коефіцієнт використання освітлювальної установки = 0,6;

$M$  - кількість ламп у світильнику = 2.

З формули (4.5) виразимо  $n$  (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже,  $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$ .

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

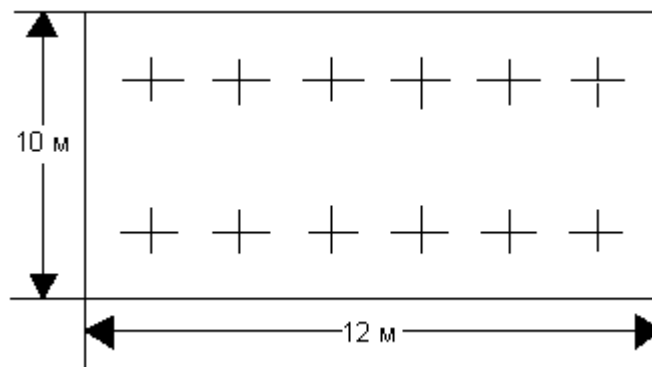


Рисунок 4.1 - Схема розташування світильників

#### 4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 З ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 З, температура самозаймання 530 З, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [46].

Згідно [47] приміщення відносяться до категорії В(пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогненебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегрів від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [48] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізолюване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;

- організація автоматичного контролю параметрів, що визначають джерела запалення;

- ліквідація можливості самозаймання речовин і матеріалів .

- Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;

- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;

- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [49] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, лом, багри, шерстяну або азбестову ковдру) [20]. Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

## **4.5 Охорона навколишнього природного середовища**

### **4.5.1 Загальні дані з охорони навколишнього природного середовища**

Діяльність за темою магістерської роботи, а саме розробці автоматизованої системи моделювання рівноважного складу впливає на навколишнє природне середовище і регламентується нормами діючого законодавства [51 - 56].

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на віддалення (знешкодження), утилізацію, тощо в ІТ галузі.

В процесі створення/розробки програми на робочому місці виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки
- Змінні носії інформації - IV клас небезпеки
- Відпрацьовані вогнегасники - IV клас небезпеки
- Макулатура - IV клас небезпеки
- Відпрацьовані фільтрувальні засоби індивід. захисту (респіратори, протигази) - IV клас небезпеки
- Побутові відходи - IV клас небезпеки

### **4.5.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі**

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог Державних санітарних правил і норм [57].

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця тимчасового зберігання відходів відповідно до затвердженої схеми їх розміщення. Зазначені для зберігання відходів місця чи об'єкти повинні використовуватися лише для заявлених відходів.

Способи тимчасового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- Відходи I класу небезпеки зберігаються в герметичній тарі (сталеві бочки, контейнери). У міру наповнення тару з відходами закривають герметично сталевий кришкою;

- Відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автотранспортом перевантажують у самоскид і доставляють на місце утилізації або захоронення;

Особливий контроль наділяється збору і зберіганню відпрацьованих ртутьвмісних ламп (енергоощадних) як відходам I класу небезпеки, що збираються і обов'язково передаються на утилізацію підприємствам, що мають ліцензію на поводження з такими небезпечними відходами.

Всі відходи, що утворюються в процесі діяльності/роботи, підлягають обліку.

Побутові та будівельні відходи вивозяться на полігон твердих побутових відходів міста, також відповідно до договору з комунальним дорожньо-експлуатаційним управлінням.

Особи, винні в порушенні встановленого порядку поводження з відходами (порушення правил обліку відходів, самовільне складування і видалення відходів, передача відходів в інші підприємства/організації з порушенням встановлених правил), згідно законодавства несуть дисциплінарну, адміністративну або кримінальну відповідальність.

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

## ВИСНОВКИ

В процесі виконання магістерської роботи були отримані наступні результати:

1. Розглянуто існуючі методи створення систем автоматизованого проектування ПТК, до яких відносяться методи структурного і об'єктно-орієнтованого проектування, що входять до складу більш загального системного підходу.

2. Проведено порівняльний аналіз даних методів автоматизованого проектування. На основі проведеного аналізу, було визначено найбільш оптимальний метод для автоматизованого проектування ПТК на базі МСКУ-4.1 - структурний метод.

3. Розглянуто особливості мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1.

4. Розроблено і налагоджена в середовищі розробки MS Access база даних і засоби діалогу з нею для коректного проектування підсистеми нижнього рівня ПТК на базі мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1 відповідно до вимог технічного завдання.

5. У розділі «Охорона праці та безпека в надзвичайних ситуаціях» виконаний аналіз потенційних небезпек при роботі із засобами обчислювальної техніки, розроблені заходи з охорони праці, з безпеки в надзвичайних ситуаціях. Були проведені відповідні розрахунки.



**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ**

- 1) Ястребенецький М. А. Системи управління і захисту ядерних реакторів / М. А. Ястребенецький, Ю. В. Розен, С. В. Виноградська, Г. Джонсон, В. В. Єлісєєв, А. А. Сіора, В. В. Скляр, Л. І. Спектор, В. С. Харченко; Під ред. М. А. Ястребенецький - К.: Основа-Принт, 2011. - 768с. - (Безпека атомних станцій).
- 2) Близько А. Р. Автоматизовані системи управління технологічними процесами [мет. сел.] / А. Р. Близько, А. А. Москаленко - Мінськ: БНТУ, 2012.
- 3) Єлісєєв В. В. Програмно-технічні комплекси АСУ ТП. Навчальний посібник / В. В. Єлісєєв, В. А. Ларгін, Г. Ю. Пивоваров - К.: Видавничо-поліграфічний центр «Київський університет», 2003. - 429 с.
- 4) Прохоров Н. Л. Програмно-технічні комплекси в системах автоматизації технологічних процесів на атомних станціях / Н. Л. Прохоров, В. І. Глухів, В. А. Прилипко - 2008.
- 5) Єлізаров І. А. Технічні засоби автоматизації / І. А. Єлізаров, Ю. Ф. Мартем'янов, А. Г. Схиртладзе, С. В. Фролов - [2-ге вид.], - М.: Видавництво "Машинобудування -1 ", 2004.
- 6) Вайрадян Л. С. Надійність автоматизованих систем управління / Л. С. Вайрадян, Ю. Н. Федосєєв - М.: МІФІ, 1974.
- 7) ГОСТ 22.487-77 «Проектування автоматизоване. Терміни та визначення»
- 8) Ахремчик О. Л. Методологія автоматизованого проектування технічного забезпечення АСУ ТП: дис. ... доктора техн. наук / Ахремчик Олег Леонідович. - Твер, 2009. - 350 с.
- 9) ГОСТ 34.601-90 «Автоматизовані системи. Стадії створення »
- 10) ГОСТ 23501.101-87 «Системи автоматизованого проектування. Основні положення»
- 11) РД 50-680-88 «Автоматизовані системи. Основні положення»
- 12) Липа В. В. Системне проектування складних програмних засобів для інформаційних систем / Липаєв В. В. - М.: СІНГЕГ, 2002. - 268 с.
- 13) Норенков І. П. Основи автоматизованого проектування. Учеб. для вузів / Норенков І. П. - [2-ге вид., перераб. і доп.], - М.: Изд-во МГТУ ім. Н. Е. Баумана, 2002. - 336 с.
- 14) Каюмова А. В. Візуальне моделювання систем в StarUML. Учеб.для вузів / Каюмова А. В. - Казань: Казанський федеральний університет, 2013. - 104 с.

- 15) РД IDEF0-2000 «Методологія функціонального моделювання IDEF0»
- 16) Вендров, А. М. CASE-технології. Сучасні методи і засоби проектування інформаційних систем / Вендров А. М. - М.: Фінанси і статистика, 1998. - 176 с.
- 17) Похилько О. Ф. CASE-технологія моделювання процесів з використанням засобів BPWin і ERWin [уч. сел.] / А. Ф. Похилько, І. В. Горбачов. - Ульяновськ: УЛГТУ, 2008. - 120 с.
- 18) Похилько О. Ф. Моделювання процесів і даних з використанням CASE-технологій / А. Ф. Похилько, І. В. Горбачов, С. В. Рябов. - Ульяновськ: УЛГТУ, 2014. - 163 с.
- 19) Маклаков, С. В. BPWin і ERWin. CASE-засобу розробки інформаційних систем / Маклаков С. В. - М.: ДІАЛОГ-МІФІ, 1999. - 256 с.
- 20) Летуновский Д. В. StarUML. Керівництво розробника / Летуновский Д. В., 2007
- 21) Кулямин В. В. Технології програмування. Компонентний підхід / Кулямин В. В.
- 22) Інтернет - ресурс. [http://www.imp.lg.ua/booklet/npp\\_book.pdf](http://www.imp.lg.ua/booklet/npp_book.pdf)
- 23) Керівництво по експлуатації на комплекс керуючий обчислювальний МСКУ-4.1.- 147 с.
- 24) Вічугова А. А. Методи і засоби концептуального проектування інформаційних систем: порівняльний аналіз структурного і об'єктно-орієнтованого підходів / Вічугова А. А.
- 25) ГОСТ 12.1.013-78 «Будівництво. Електробезпека. Загальні вимоги»
- 26) ГОСТ 12.1.003-74 "ССБТ. Оборудование производственное. Загальні вимоги безпеки »
- 27) СНиП II-4-79 «Природне і штучне освітлення»
- 28) ОНТП 24-86 «Визначення категорій приміщень і будинків по вибухопожежної і пожежної небезпеки»
- 29) Ареф'єва Е. А. Об'єктно-орієнтована система моделювання в автоматизованому управлінні: дис. ... канд. техн. наук / Ареф'єва Е. А. - Тула, 1999..
- 30) Когаловскій М. Р. Енциклопедія технологій баз даних / Когаловскій М. Р. ; - М.: Фінанси і статистика, 2002. - 800 с.
- 31) Кузин А. В., Левонісова С. В. Бази даних / Кузін А. В., Левонісова С. В. ; - Академія, 2012. - 320 с.
- 32) Гончаров А. Ю. ACCESS 2003. Самовчитель із прикладами / Гончаров А. Ю. ; - М.: Кудіц - образ, 2004. - 272 с.

- 33) Міхеєва В., Харитоновна І. Microsoft Access 2003 / Міхеєва В., Харитоновна І. - СПб .: БХВ-Петербург, 2004. - 1069 с.
- 34) Ржеуцкая С. Ю. Базы данных. Язык SQL / Ржеуцкая С. Ю .; - Вологда: ВоГТ, 2010. - 159 с.
- 35) Алан Бьюлі - Вивчаємо SQL / Алан Бьюлі; - O'Reilly-Символ, 2007. - 311 с.
- 36) ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
- 37) ГОСТ 12.0.003-74 Опасные и вредные производственные факторы. Классификация.
- 38) НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів
- 39) ГОСТ 12.1.009-76. ССБТ. Электробезопасность. Термины и определения.
- 40) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів.
- 41) ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
- 42) ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
- 43) TCO' 07 Certified Displays. © 2007 Copyright TCO Development AB
- 44) ДСанПіН 3.3.2.007-98, Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
- 45) ДБН В.2.5-28-2006. Природне і штучне освітлення
- 46) ГОСТ 12.1.044-89 Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
- 47) НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.
- 48) ГОСТ 12.1.004-91. "Система стандартов безопасности труда. Пожарная безопасность. Общие требования".
- 49) НАПБ А.01.001-2014 "Правила пожежної безпеки в Україні"
- 50) НАПБ Б.03.001-2004. Про затвердження Типових норм належності вогнегасників.
- 51) Закон України «Про охорону навколишнього природного середовища»
- 52) Закон України «Про забезпечення санітарного та епідемічного благополуччя населення»
- 53) Закон України «Про відходи»

- 54) Закон України «Про охорону атмосферного повітря»
- 55) Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру»
- 56) Водний кодекс України
- 57) ДСанПіН 2.2.7.029-99. Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення.

## ДОДАТОК А.

### Запити в MS Access

Запит «Вибір кількості модулів» (рисунок А.1) виводить в випадяючому списку «Місце установки МСО» задану кількість значень (це значення не повинно перевищувати 72). Результат виконання запити представлений на рисунку А.2.

```

Выбор_количества_модулей : запрос на выборку
SELECT Место_установки_МСО.Место_модуля
FROM Место_установки_МСО
WHERE (((Место_установки_МСО.Место_модуля)<=Forms![Сигналы МСКУ]!field1) And ((Forms![Сигналы МСКУ]!field1)<=72));

```

Рисунок А.1 - Запит «Вибір кількості модулів»

Рисунок А.2 - Результат виконання запити «Вибір кількості модулів»

Запит «Зайняті місця обраним модулем» (рисунок А.3) виводить в список «Зайняті місця установки» значення місць установки МСО, які для цього типу модуля вже були занесені в результуючу таблицю «Сигналы МСКУ» (рисунок А.4). Це поле є підказкою користувачеві про те, що обраний модуль на виведених місцях установки вже займає якісь номери каналів.

```

Занятые_места_выбранным_модулем_запрос : запрос на выборку
SELECT DISTINCT [Сигналы МСКУ].[Место установки МСО]
FROM [Сигналы МСКУ]
WHERE ((([Сигналы МСКУ].[Тип МСО])=Forms![Сигналы МСКУ]!fieldID2);

```

Рисунок А.3 - Запит «Зайняті місця обраним модулем»

Сигналы МСКУ	
Наименование сигнала	<input type="text"/>
Тип сигнала	±0,5 μА <span style="float: right;">▼</span>
Тип МСО	ПКИ -41 <span style="float: right;">▼</span>
Место установки МСО	<input type="text"/> <span style="float: right;">▼</span> Занятые места установки
Номер канала МСО	<input type="text"/> <span style="float: right;">▼</span> 4 ▲ 6 ▼

Рисунок А.4 - Результат виконання запиту «Зайняті місця обраним модулем»

Поле «Номер каналу МСО» має відображати тільки ті номери каналів, які є вільними для обраного раніше МСО і місця його установки. Ці дії виконує запит «Вибір № каналу з видаленням зайнятих» (рисунок А.5).

```

Выбор_№канала_с_удалением_зайнятых_запрос : запрос на выборку
SELECT DISTINCT Выбор_№канала_МСО.№канала_модуля
FROM Выбор_№канала_МСО, Определение_количества_каналов_модуля_запрос
WHERE (((Выбор_№канала_МСО.№канала_модуля) Not In (SELECT [Номер канала МСО]
FROM Занятые_№каналов_выбранным_модулем_и_местом_запрос) And
(Выбор_№канала_МСО.№канала_модуля)<=((Определение_количества_каналов_модуля_запрос).[Количество_каналов_модуля])) AND
((Выбор_№канала_МСО.Место_модуля)=[Forms]![Сигналы МСКУ]![fieldID3]));

```

Рисунок А.5 - Запит «Вибір № каналу з видаленням зайнятих»

У запиті «Вибір № каналу з видаленням зайнятих» додатково використовується два вкладених запиту: «Визначення кількості каналів модуля» (рисунок А.6), який за типом обраного модуля визначає максимальну кількість каналів, і «Зайняті №каналов обраним модулем і місцем» ( малюнок А.7), який визначає вже зайняті номери каналів для обраного модуля і місця його установки.

```

Определение_количества_каналов_модуля_запрос : запрос на выборку
SELECT DISTINCT Выбор_типа_МСО.Количество_каналов_модуля
FROM Выбор_типа_МСО
WHERE (Выбор_типа_МСО.Тип_модуля)=[Forms]![Сигналы МСКУ]![fieldID2];

```

Рисунок А.6 - Запит «Визначення кількості каналів модуля»

```

Занятые_№каналов_выбранным_модулем_и_местом_запрос : запрос на выборку
SELECT [Сигналы МСКУ].[Номер канала МСО]
FROM [Сигналы МСКУ]
WHERE ((([Сигналы МСКУ].[Тип МСО])=[Forms]![Сигналы МСКУ]![fieldID2] And ((([Сигналы МСКУ].[Место установки МСО])=[Forms]![Сигналы МСКУ]![fieldID3]))
ORDER BY [Сигналы МСКУ].[Номер канала МСО];

```

Рисунок А.7 - Запит «Зайняті № каналів обраним модулем і місцем»

Запит «Вибір типу ПКД» (рисунок А.8) дозволяє користувачеві при виборі зі списку «Тип МСО» на формі одного значення типу модуля отримати всі відповідні йому типи ПКД з описом. Результат виконання запиту виводиться в випадіючому списку «Тип ПКД» на формі (рисунок А.9).

```

Выбор_типа_ПСд_запрос : запрос на выборку
SELECT Выбор_типа_ПСд.Тип_ПСд, Выбор_типа_ПСд.Описание_ПСд
FROM Выбор_типа_ПСд
WHERE (((Выбор_типа_ПСд.Тип_модуля)=Forms![Сигналы МСКУ]!fieldID2)) And ((Выбор_типа_ПСд.Тип_модуля)<>"");

```

Рисунок А.8 - Запит «Вибір типу ПСд»

The screenshot shows a web form titled "Сигналы МСКУ". The form contains several input fields and dropdown menus. The "Тип ПСд" field is selected, showing a list of options: "ПСд-204 Без резервирования" and "ПСд-180 С резервированием". Other fields include "Наименование сигнала", "Тип сигнала" (set to ±0,5 μA), "Тип МСО" (set to ПКИ-41), "Место установки МСО" (set to 4), "Номер канала МСО" (set to 2), and "Занятые места установки" (set to 4). The "Место установки ПСд" field is also visible, showing a list of options: "ПСд-204" and "ПСд-180".

Рисунок А.9 - Результат виконання запиту «Вибір типу ПСд»

Поля «Місце установки ПКД», «Номер каналу ПКД», «Зайняті місця установки» для ПКД повністю аналогічні полях «Місце установки МСО», «Номер каналу МСО», «Зайняті місця установки» для МСО. Тому і вибір значень в ці поля здійснюється за запитами «Вибір кількості ПСД», «Вибір № каналу ПКД з видаленням зайнятих», «Зайняті місця обраним ПКД», які аналогічні запитам «Вибір кількості модулів», «Вибір № каналу з видаленням зайнятих», «Зайняті місця обраним модулем», представленим на рис. А.1, А.6 і А.3.

**ДОДАТОК Б.**  
**Комп'ютерна презентація**

*Східноукраїнський національний університет ім.В.Даля*

*Комп'ютерна система автоматизованого  
проектування програмно-технічного комплексу*

Студент групи КСМ-16дм

Вовченко Д.В.

Керівник проекту

Барбарук В.М.

2

**МЕТА РОБОТИ**

*Метою виконання даної магістерської роботи є:*

- ✓ *дослідження існуючих методів створення систем автоматизованого проектування ПТК АСУ ТП;*
- ✓ *Розробка системи автоматизованого компонування підсистем нижнього рівня мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1.*



## ПОСТАНОВКА ЗАДАЧІ

Одною з проблем створення ПТК АСУ ТП є оперативне і коректне проектування підсистеми нижнього рівня, обсяг якої становить 90-95% усього обладнання ПТК.

Завдання швидкого і коректного проектування ПТК, в тому числі і підсистем нижнього рівня, може бути вирішена застосуванням САПР.

Необхідно виконати:

- ✓ вибір методу проектування системи компоновки підсистем нижнього рівня технічного комплексу МСКУ-4.1;
- ✓ розробку автоматизованої системи компоновки підсистем нижнього рівня технічного комплексу на базі МСКУ-4.1

## СИСТЕМНИЙ ПІДХІД ДО ПРОЕКТУВАННЯ

Основні ідеї та принципи проектування складних систем, до яких відносяться системи автоматизованого проектування ПТК, виражені в системному підході.

Головним завданням системного підходу є пошук шляхів по перетворенню складного в просте, з розкладання важкозрозумілого завдання на ряд завдань, що мають рішення.

Інтерпретація та конкретизація системного підходу виражена в:

- ✓ структурному підході;
- ✓ об'єктно-орієнтованому підході.

## СТРУКТУРНИЙ ПІДХІД ДО ПРОЕКТУВАННЯ

В основу структурного або функціонально-модульного підходу покладено принцип функціональної декомпозиції, при якій структура системи описується в термінах ієрархії її функцій та передачі інформації між окремими функціональними елементами.

У структурному аналізі використовуються три групи засобів моделювання:

- ✓ функціональне моделювання - діаграми потоків даних (DFD) і діаграми SADT;
- ✓ інформаційне моделювання - діаграми «сутність - зв'язок» (ERD);
- ✓ динамічне моделювання - діаграми IDEF3.

## ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД ДО ПРОЕКТУВАННЯ

Об'єктно-орієнтований підхід до проектування використовує об'єктну декомпозицію. При цьому структура системи описується в термінах об'єктів та зв'язків між ними, а поведінка системи описується в термінах обміну повідомленнями між об'єктами.

Даний підхід виражений в уніфікованій мові моделювання (UML), яка в даний час фактично є стандартом у цій галузі.

## ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД ДО ПРОЄКТУВАННЯ

В UML визначено наступний набір діаграм, кожна з яких демонструє певний аспект опису моделі:

### 1) структурні діаграми:

- ✓ діаграма класів;
- ✓ діаграми компонентів;
- ✓ діаграми розгортання;

### 2) діаграми поведінки:

- ✓ діаграми станів;
- ✓ діаграми діяльності;
- ✓ діаграми варіантів використання;
- ✓ діаграми послідовності.

## ПОДІБНІСТЬ СТРУКТУРНОГО ТА ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПІДХОДІВ

Типова задача проектування	Методи вирішення	
	Структурний підхід	Об'єктно-орієнтований підхід
Функціональний розподіл на верхньому рівні абстракції	SADT	UML-діаграми варіантів використання
Визначення типів предметів, взаємозв'язків між ними і їх властивостей	ERD	UML-діаграма класів
Опис руху потоків даних	DFD	UML-діаграма послідовностей
Опис логічної послідовності виконання функцій бізнес-процесу	IDEF3/PFDD	UML-діаграма діяльності
Опис зміни станів предмета	IDEF3/OSTN	UML-діаграми станів

## ПЕРЕВАГИ ТА НЕДОЛІКИ СТРУКТУРНОГО ПІДХОДУ

Головна перевага комплексного застосування універсальних графічних структурних методологій SADT, DFD, ERD і IDEF3 - це можливість забезпечити логічну цілісність і повноту опису моделі, необхідних для досягнення точних і несуперечливих результатів, а також виявити всі можливі неузгодженості та неточності завдяки високій наочності моделі на окремих рівнях декомпозиції.

Недоліком даного підходу є складність сприйняття ієрархічно впорядкованої інформації, а також складність модифікації моделі.

## ПЕРЕВАГИ ТА НЕДОЛІКИ СТРУКТУРНОГО ПІДХОДУ

Перевагою об'єктно-орієнтованого підходу є можливість відображати поведінку сутностей реального світу, висока адаптивність системи до змін і високий рівень уніфікації розробки, що робить модель придатною для повторного використання.

Недоліки даного підходу полягають в низькій наочності діаграм, що відображають специфіку об'єктного підходу, особливо для непрофесіоналів. Також даний підхід не дає негайної віддачі. Ефект від його застосування позначається після розробки двох-трьох проектів і накопичення повторно використовуваних компонентів.

11

## МІКРОПРОЦЕСОРНІ УПРАВЛЯЮЧІ ОБЧИСЛЮВАЛЬНІ КОМПЛЕКСИ

МСКУ-4.1 - це сімейство проектно-компонованих, гнучко програмованих логічних промислових контролерів, призначених для вирішення широкого кола завдань в автоматизованих системах управління технологічними процесами і для застосування в якості:

- ✓ промислових контролерів відмовостійких систем автоматизації особливо відповідальних об'єктів;
- ✓ підсистем нижнього рівня АСУ ТП;
- ✓ інтелектуальних автономних систем контролю і управління.

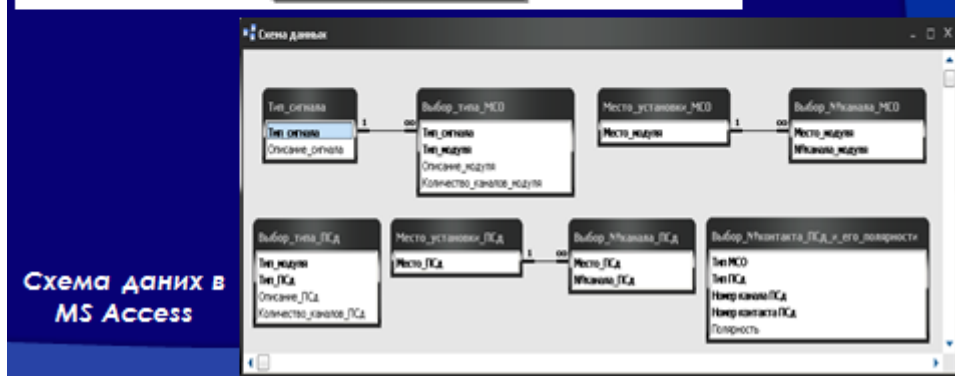
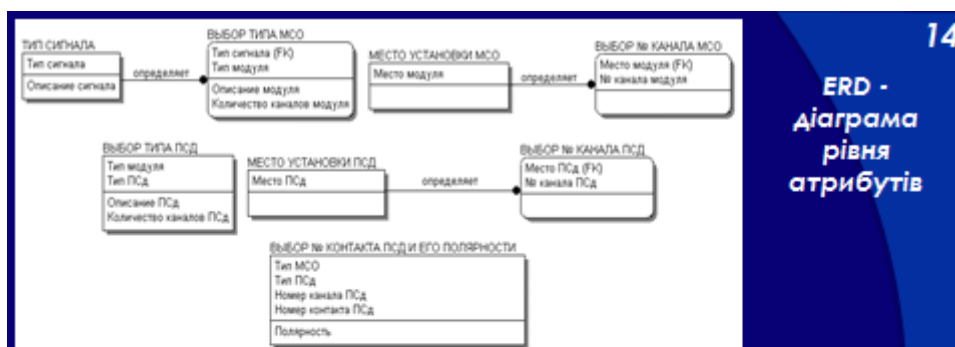
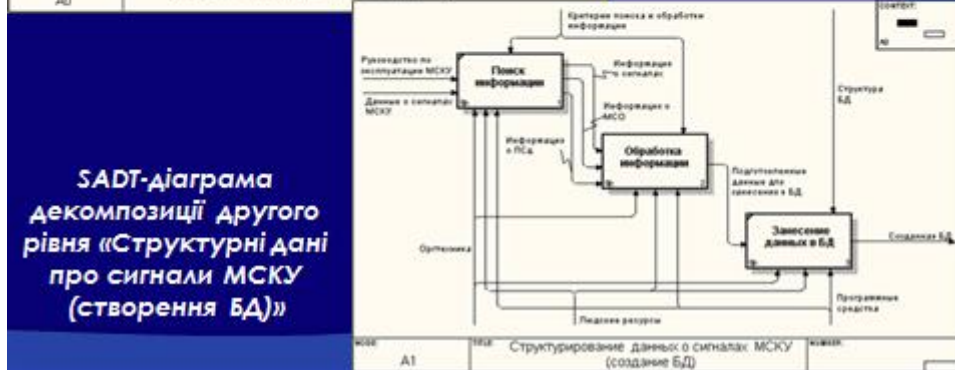
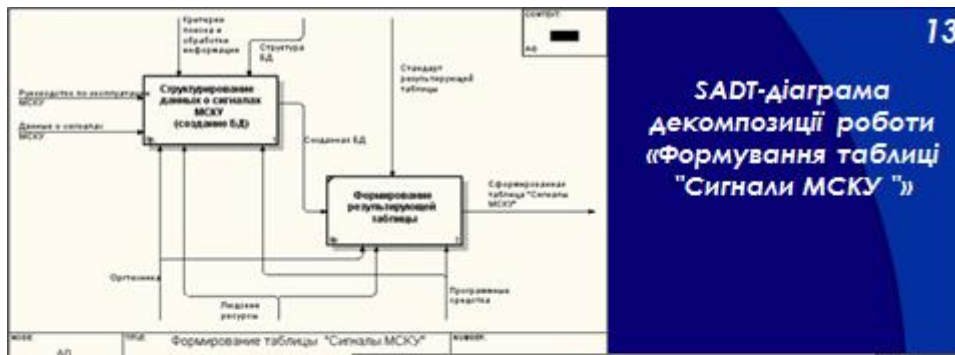
12

## ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОМПОНОВКИ ПІДСИСТЕМ НИЖНЬОГО РІВНЯ КОМПЛЕКСУ МСКУ-4.1

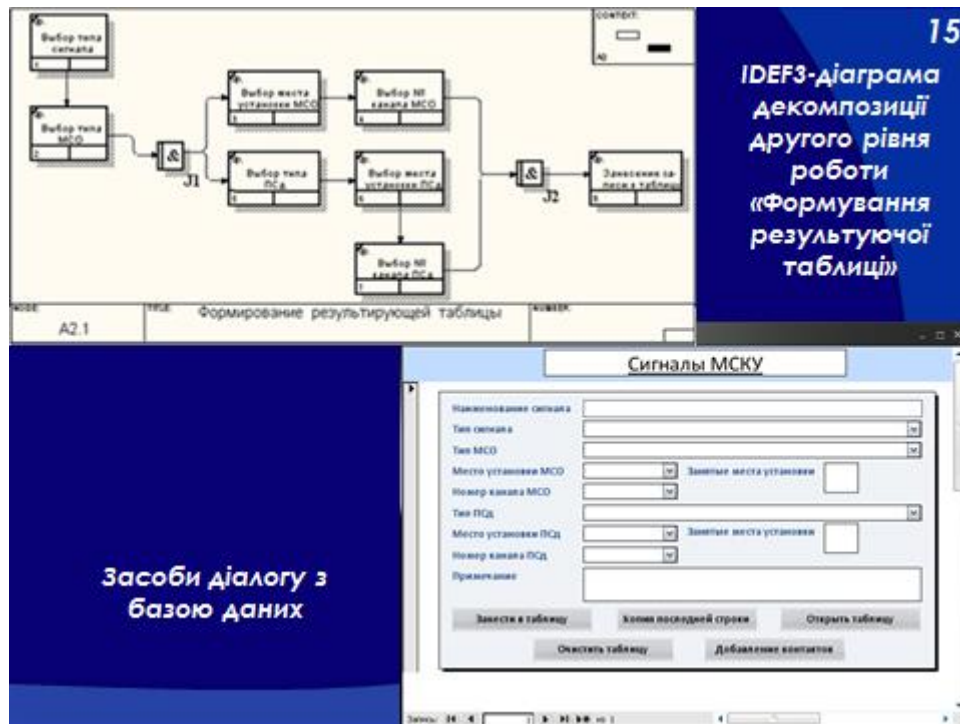
Для розробки автоматизованої системи компоновки підсистем нижнього рівня комплексу МСКУ-4.1 були обрані структурні методології.



SADT-діаграма (контекстна діаграма)







16

## ВИСНОВКИ

В процесі виконання майстерської роботи були отримані наступні результати:

1. Розглянуто існуючі методи створення систем автоматизованого проектування ППК, до яких відносяться методи структурного і об'єктно-орієнтованого проектування.
2. Проведено порівняльний аналіз даних методів, на підставі чого, було визначено найбільш оптимальний метод для автоматизованого проектування ППК на базі МСКУ-4.1 - структурний метод.
3. Розглянуто особливості мікропроцесорного керуючого обчислювального комплексу МСКУ-4.1.
4. Розроблено і налагоджена в середовищі розробки MS Access база даних і засоби діалогу з нею для коректного проектування підсистеми нижнього рівня ППК на базі МСКУ-4.1