

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка
перший (бакалаврський)
(рівень вищої освіти)

Розробка програмного модуля інформаційної системи
управління приладобудівним виробництвом
(тема)

Виконав:
студент 4 курсу, групи АКТАКІТ-20-2
Кравченко Д. С.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Автоматизація та
комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник доц. Аллахверанов Р. Ю.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАР

Невлюдов І. Ш.
(прізвище, ініціали)

2024р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри КІТАР _____
(підпис)
«_____» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Кравченка Давиду Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного модуля інформаційної системи управління приладобудівним виробництвом

затверджена наказом по університету від “ 03 ” червня 2024р. № 544 Ст.

2. Термін подання студентом роботи “ 27 ” червня 2024р.

3. Вихідні дані до роботи _____

3.1 Автоматизоване робоче місце виробничо-технологічного призначення;

3.1 Використовувати реляційну базу даних;

3.2 Архітектура "файл/сервер";

3.3 Зберігання даних в таблицях типу Paradox 7;

3.5 Мова програмування Delphi;

3.6 Оформлення текстової документації згідно ДСТУ 3008 – 2015.

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Вступ;

4.2 Призначення пристроїв і функції автоматизованого робочого місця;

4.3 Надання інформації та способи їх передачі;

4.4 Ієрархічну модель бази даних для модуля інформаційної системи;

4.5 Розробка програмного забезпечення для модуля інформаційної системи;

4.6 Заходи і розрахунки для забезпечення безпечних умов праці;

4.7 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 12 с. формату А4

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Призначення пристроїв і функції автоматизованого робочого місця</i>	13.05 – 18.05.24	виконано
2	<i>Надання інформації та способи їх передачі</i>	19.05 – 24.05.24	виконано
3	<i>Ієрархічну модель бази даних для модуля інформаційної системи</i>	25.05 – 30.05.24	виконано
4	<i>Розробка програмного забезпечення для модуля інформаційної системи</i>	31.05 – 09.06.24	виконано
5	<i>Заходи і розрахунки для забезпечення безпечних умов праці</i>	10.06 – 12.06.24	виконано
6	<i>Оформлення пояснювальної записки</i>	13.06 – 15.06.24	виконано
7	<i>Подання роботи на перевірку Інтернет-системою StrikePlagiarism</i>	16.06 – 19.06.24	виконано
8	<i>Подання роботи на рецензію</i>	20.06 – 22.06.24	виконано
9	<i>Подання роботи на підпис зав. кафедри</i>	23.06 – 25.06.24	виконано
10	<i>Подання кваліфікаційної роботи в ЕК</i>	26.06.24	виконано

Дата видачі завдання 13.05.2024р.

Студент _____ Кравченко Д. С.
(підпис)

Керівник роботи _____ доц. Аллахверанов Р. Ю.
(підпис) (посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«13» червня 2024 р.

Кравченко Д. С.

РЕФЕРАТ

Пояснювальна записка: 77 с., 1 табл., 13 рис., 4 дод., 13 джерел.

ПРИЛАДОБУДІВНЕ ВИРОБНИЦТВО, МОДУЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ, АВТОМАТИЗОВАНІ РАБОЧІ МІСЦЯ, ГНУЧКА ВИРОБНИЧА СИСТЕМА, БАЗА ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Мета роботи – підвищення ефективності управління приладобудівним виробництвом.

Об'єктом розробки – система автоматизації виробничого процесу на приладобудівному виробництві.

Предмет розробки – програмне забезпечення.

В роботі проведено аналіз технічного завдання та літератури по темі кваліфікаційної роботи бакалавра. Проаналізовано функції інформаційних систем і підсистем управління, наведено призначення пристроїв автоматизованого робочого місця. Досліджено види надання інформації та способи їх передачі в автоматизованих підсистемах управління.

Спроектовано ієрархічну модель бази даних для модуля інформаційної системи управління. Розроблено програмне забезпечення для модуля інформаційної системи управління автоматизованого робочого місця на приладобудівному виробництві.

Проведено заходи і розрахунки для забезпечення безпечних умов праці в лабораторії де виконувалася кваліфікаційна робота.

ABSTRACT

Explanatory note: 77 pp., 1 tab., 13 figs., 4 appendices, 13 sources.

INSTRUMENT MANUFACTURING, MANAGEMENT INFORMATION SYSTEM MODULE, AUTOMATED WORKPLACES, FLEXIBLE PRODUCTION SYSTEM, DATA BASE, SOFTWARE.

The purpose of the work is the development of software for the module of the information system of the management of the automated workplace at the instrument-making industry.

The object of the development is information support of the automated production process in the instrument manufacturing industry.

The subject of development is software.

In the work, an analysis of the technical task and literature on the subject of the bachelor's qualification work was carried out. The functions of information systems and control subsystems are analyzed, the purpose of the devices of the automated workplace is given. Types of information provision and methods of their transmission in automated control subsystems were studied.

A hierarchical model of the database for the management information system module was designed. The software was developed for the module of the information system of the management of the automated workplace at the instrument-making industry.

Measures and calculations were carried out to ensure safe working conditions in the laboratory where qualification work was performed.

ЗМІСТ

Перелік скорочень	9
Вступ	10
1 Організація автоматизованого робочого місця на приладобудівному виробництві	12
1.1 Функціональна схема організації автоматизованого робочого місця	12
1.2 Структура сучасних гнучких виробничих систем	15
1.3 Будова інформаційної системи управління гнучким виробництвом	20
2 Проектування ієрархічної моделі бази даних модуля інформаційної системи управління	23
2.1 Концепція розподіленої обробки інформації	23
2.2 Нові інформаційні технології	30
2.3 Функціональне моделювання інформаційної системи управління	42
2.4 Архітектура баз даних	45
2.5 Моделювання даних	51
3 Автоматизація модуля інформаційної системи управління	54
3.1 Обґрунтування вибору мови програмування	54
3.2 Розробка програмного забезпечення інформаційної системи управління	60
3.3 Тестування і керівництво інформаційної системи управління	66
4 Заходи і розрахунки для забезпечення безпечних умов праці	70
4.1 Аналіз умов праці на робочому місці	70
4.2 Промислова безпека на робочому місці	70
4.3 Виробнича санітарія у приміщенні	71

4.4 Пожежна безпека виробничого приміщення	73
Висновки	75
Перелік джерел посилання	76
Додаток А Структура таблиць модуля інформаційної системи управління	78
Додаток Б Взаємозв'язок таблиць модуля інформаційної системи управління	79
Додаток В Лістинг програми модуля інформаційної системи управління	80
Додаток Г Демонстраційний матеріал	78

ПЕРЕЛІК СКОРОЧЕНЬ

АРМ – автоматизовані робочі місця;

БД – база даних;

ЕОМ – електронно-обчислювальна машина;

ГВС – гнучка виробнича система;

ІВС – інформаційно-вимірювальна система;

ІСУ – інформаційна система управління;

ЛОМ – локальна обчислювальна мережа;

НФ – нормальна форма;

ПВ – приладобудівне виробництво;

ПЕОМ – персональна електронно-обчислювальна машина;

ПЗ – програмне забезпечення;

РСУ – робоча станція управління;

СУБД – система управління базами даних.

ВСТУП

Автоматизація повсюдно вважається головним, найбільш важливим напрямком у розвитку промислового виробництва. В умовах автоматизованого виробництва актуальним завданням є комплекс заходів по розробці нових, прогресивних технологічних процесів, автоматизованих робочих місць (АРМ) і створення на їх основі нових високопродуктивних систем. Завдяки звільненню людини від безпосередньої участі у виробничих процесах, а також високій концентрації основних операцій значно поліпшуються умови праці і економічні показники виробництва. АРМ – програмно-технічний комплекс, призначений для автоматизації діяльності певного виду.

АРМ об'єднує програмно-апаратні засоби, що забезпечують взаємодію людини з електронно-обчислювальною машиною (ЕОМ), надає можливість введення (через пристрої введення інформації) і виведення інформації (через пристрої виведення). Відповідно до принципу системності, АРМ слід розглядати як системи, структура яких визначається функціональним призначенням. Принцип гнучкості означає пристосованість системи до можливих перебудов, завдяки модульності побудови всіх підсистем і стандартизації їх елементів.

Важливим завданням автоматизації є отримання і обробка інформації про стан всіх ланок виробничого процесу, систем управління процесами виробництва, обліку продукції та оперативного планування її випуску, яка на даному етапі вирішується за допомогою високопродуктивних ЕОМ.

Мета роботи – підвищення ефективності управління приладобудівним виробництвом.

Об'єктом розробки – система автоматизації виробничого процесу на приладобудівному виробництві.

Предмет розробки – програмне забезпечення.

Відповідно до поставленої мети в ході роботи необхідно вирішити наступні завдання:

- вивчити теоретичні аспекти створення АРМ на приладобудівному виробництві;
- проаналізувати призначення пристроїв і функції АРМ;
- дослідити види надання інформації та способи їх передачі;
- спроектувати ієрархічну модель бази даних для модуля інформаційної системи управління;
- розробити програмне забезпечення для модуля інформаційної системи управління;
- провести заходи і розрахунки для забезпечення умов безпечної праці.

Пояснювальну записку оформлено згідно з ДСТУ 3008:2015 [1], а також з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти [2-3].

1 ОРГАНІЗАЦІЯ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ НА ПРИЛАДОБУДІВНОМУ ВИРОБНИЦТВІ

Розглянемо сучасне приладобудівне виробництво як гнучку виробничу систему.

1.1 Функціональна схема організації автоматизованого робочого місця

Незупинні темпи технічного прогресу підвищують актуальність обчислювальної техніки в процесах управління. Залучення можливостей сучасної обчислювальної техніки з метою автоматизації процесу обробки інформації сприяє збільшенню продуктивності праці, підвищенню ефективності роботи, а також прискоренню обміну інформацією з управління.

У широкому сенсі АРМ набуває значення комплексу фізичних модулів, які об'єднані між собою, забезпеченого програмними засобами та здатного втілити закінчену інформаційну технологію. Даний комплекс складається з: процесора, дисплея, принтера, плотера, сканера, стримера, маніпулятора, клавіатури, спеціального обладнання. Крім того, програмних засобів, які містять операційні системи, системи керування базами даних, прикладні пакети, графічні та текстові редактори, засоби електронних таблиць, оригінальні програми, діалогові взаємодії ЕОМ та оператора. Іншими словами, мова йде про комплекс технічного та програмного забезпечення, тобто це інструменти будь-якого оператора. АРМ, яке створюється для виконання конкретних функцій керування, отримало назву функціонального (рисунок 1.1).

Роль АРМ визначається характером їхньої участі в процесі керування тієї чи іншої галузі виробничої діяльності. Функції АРМ окреслюються реалізацією предметної технології, зокрема, у послідовності етапів

модифікації первинної інформації в результат, використовуючи інформаційну технологію. Класифікація АРМ здійснюється за функціональним призначенням, за видами розв'язуваних завдань, за режимом експлуатації.

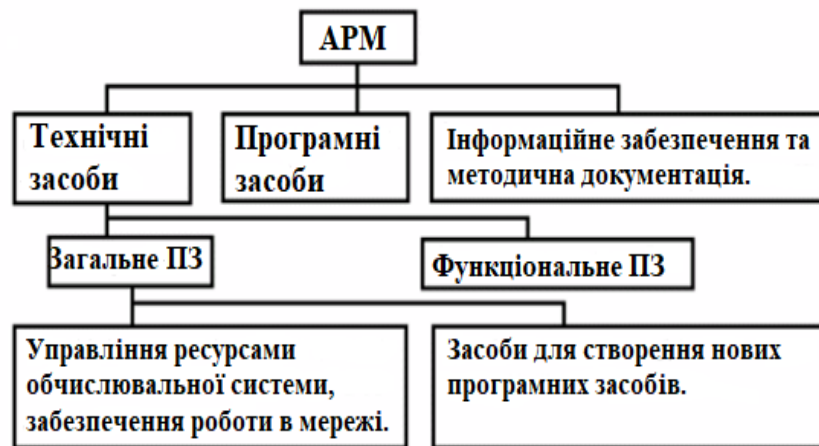


Рисунок 1.1 – Функціональна схема організації автоматизованого робочого місця

За функціональною ознакою АРМ бувають:

- адміністративно-управлінського персоналу;
- проєктувальника радіоелектронної апаратури, автоматизованих систем управління тощо;
- фахівця в галузі математики, фізики, економіки;
- виробничо-технологічного призначення.

Для кожного об'єкта керування АРМ створюються відповідно до їхнього функціонального призначення.

АРМ мають проблемно-професійну орієнтацію щодо конкретної предметної області.

Важливою класифікаційною ознакою АРМ визначають режим його експлуатації, який може бути одиночним, груповим або мережевим.

Автоматизовані робочі місця повинні створюватися суто до їхніх функціональних призначень, що є передбаченими. Однак незмінними

залишаються загальні принципи створення АРМ. До них належать:

- системність;
- гнучкість;
- стійкість;
- ефективність.

Під принципом системності мається на увазі АРМ, що визначається як система взаємопов'язаних компонентів. При цьому структура АРМ має чітко відповідати саме тим функціям, для реалізації яких готується дане автоматизоване робоче місце.

Принцип гнучкості має величезне значення під час створення сучасних та ефективних автоматизованих робочих місць, оскільки він полягає у можливості пристосування АРМ до передбачуваної модернізації не лише програмного забезпечення, але й технічних засобів. На сьогодні, коли старіння програмних та технічних засобів відбувається з дедалі більшою швидкістю, дотримання даного принципу стає однією з найважливіших умов під час створення АРМ.

З метою забезпечення принципу гнучкості на реально працюючих автоматизованих робочих місцях усі підсистеми окремо взятого АРМ реалізуються як окремі модулі, що легко замінюються. Щоб під час заміни не поставало проблем несумісності, всі елементи мають бути стандартизованими.

Велике значення покладається на принцип стійкості, що полягає у виконанні закладених функцій до АРМ, незалежно від впливу зовнішніх і внутрішніх чинників. Якщо постають проблеми з працездатністю, система повинна швидко відновлюватися, а несправності окремих елементів – легко усуватися.

Принцип ефективності полягає в тому, що витрати на створення й експлуатацію системи не мають бути більшими за економічну вигоду від її реалізації. До того ж, під час створення АРМ необхідно враховувати, що ефективність його багато в чому визначатиметься грамотним розподілом

функцій та навантаження між працівником і машинними засобами обробки інформації, ядро яких – це ЕОМ. Тільки за умови дотримання цих вимог АРМ стає засобом підвищення як продуктивності праці й ефективності керування, так і соціальної комфортності фахівців.

1.2 Структура сучасних гнучких виробничих систем

Структура гнучкої виробничої системи (ГВС) заснована на складі та взаємозв'язках її елементів і підсистем, а також на зв'язках із зовнішнім середовищем. Визначають просторову (розташування елементів системи у просторі) та тимчасову (послідовність змін у часі стану елементів і системи в цілому) структури виробничих систем. Вони є тісно взаємопов'язаними та взаємозалежними.

Структуру ГВС представлено на рисунку 1.2.

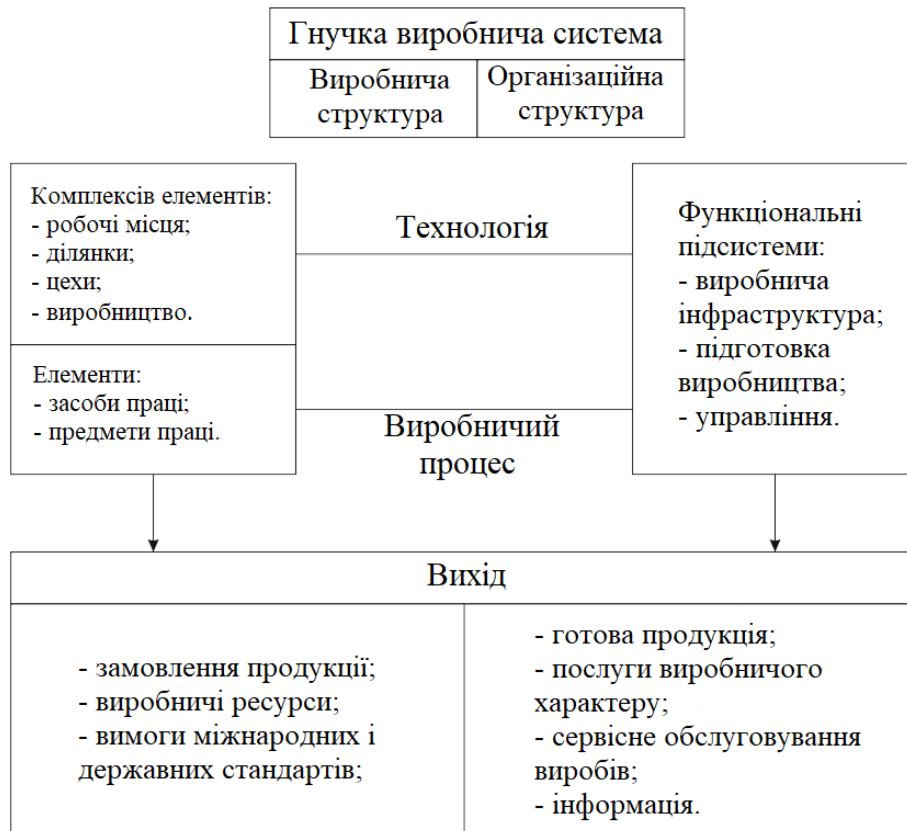


Рисунок 1.2 – Структура ГВС

Це інваріантна (незмінна) в часі фіксація елементів і зв'язків між ними. Функціонування ГВС полягає в її дії в часі. Залежно від мети й аналізу ГВС можна представляти різними структурами, зокрема, структурою основних фондів, структурою кадрів, виробничою структурою тощо. Цілісність виробничої структури вважається однією з базових властивостей.

Усі елементи ГВС функціонують з єдиною спільною метою, що полягає у розробці, проєктуванні чи виготовленні необхідної продукції. Будь-яка виробнича система складається з входу, процесу, виходу, а також зворотного зв'язку. Через пристрій входу до системи надходять вихідні ресурси (матеріали, сировина, енергія, паливо, енергія, праця тощо), вони забезпечують функціонування системи [4]. Такий процес є базовим компонентом системи, завдяки якому ресурси входу перетворюються та набувають зовсім інших нових властивостей, які вони отримують на виході. Вихід системи – це результат функціонування системи, що може бути окремим виробом, послугою, інформацією чи одночасно всім у залежності від спеціалізації виробничої системи.

Елементи, що входять до складу ГВС, різняться за своїми властивостями. Кожен з них, як структурно відокремлена частина системи, виконує функції, притаманні тільки йому. Водночас функції кожного елемента системи підпорядковуються задачам і цілям системи.

1.2.1 Інформаційні системи

На сучасному етапі розвитку науки та техніки до інформаційних систем (ІС) висуваються значно жорсткіші вимоги. Це пов'язується зі збільшенням кількості джерел інформації, а також різноманіттям задач, які ставляться під час випробувань і досліджень різних фізичних процесів та об'єктів. З огляду на це, сучасні системи збору та передачі інформації повинні відповідати як високій інформативності, так й інформаційній гнучкості. Такі характеристики дозволяють виконати програмне опитування джерел інформації та забезпечити можливість зміни частоти опитування чи точності

вимірювань за різними каналами.

Системи, що наділені найбільш високим рівнем інформаційної гнучкості, мають назву адаптивних.

Такі системи спроможні перебудувати процес вимірювань відповідно до зміни стану контролюваного об'єкта, зменшити надмірність інформації, а в залежності від стану каналу зв'язку можуть регулювати швидкість передачі інформації з метою забезпечення даного рівня достовірності.

В адаптивних системах збору та передачі інформації потрібно забезпечити адресний розподіл каналів. Для передачі інформації кожному каналу відводиться окремий часовий інтервал. До того ж, кожне джерело інформації складається як з інформаційної частини, так й адресної, в якій закодовано інформацію про номер джерела інформації.

Під час побудови моделі зручно виокремлювати в системі пристрої (підсистеми), що відповідають за певні самостійні функції, а також допускають автономне використання (за умови виготовлення різних за призначенням систем з використанням ідентичних пристроїв).

На рисунку 1.3 зображено модель ІС, що виконує функції збору, уявлення, передачі та частково обробки повідомлень.

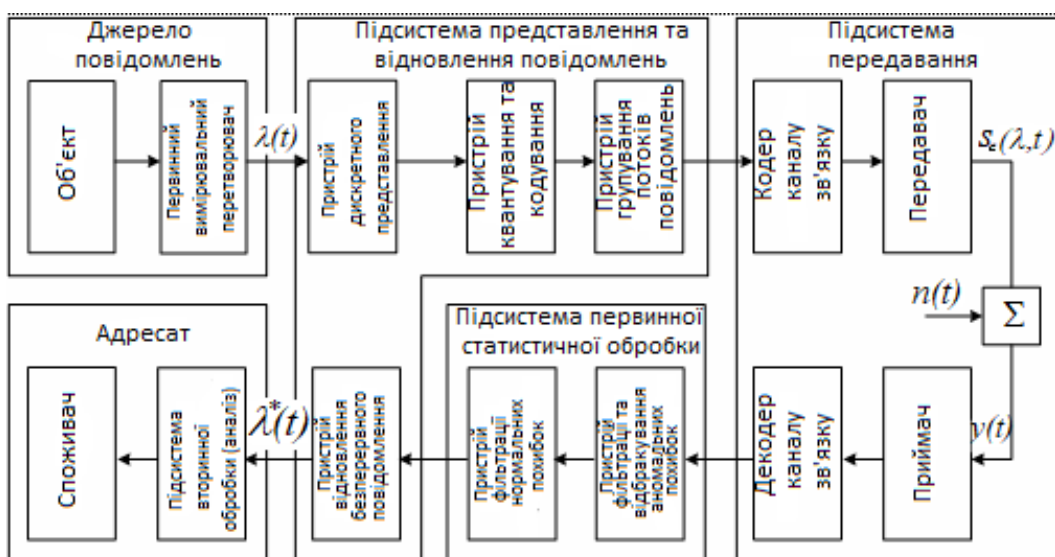


Рисунок 1.3 – Модель інформаційної системи

На вхід до системи надходить у загальному випадку аналоговий електричний сигнал $\lambda(t)$, який сформовано датчиком і первинним масштабуючим перетворювачем, а іменується він як повідомлення. Повідомлення $\lambda(t)$ досліджується як реалізація випадкового процесу $\Lambda(t)$ шляхом об'єднання безлічі повідомлень з виходу даного датчика в усіх можливих експериментах.

Ланцюг перетворення повідомлень одного датчика в багатоканальній системі створює вимірювальний канал.

У цифровій системі кожне повідомлення дискредитується за часом, а кожне дискретне значення, що номінується координатою повідомлення, квантується за рівнем та кодується. Послідовності координат, що надходять від різних вимірювальних каналів, об'єднуються в загальний потік, тобто групуються для подальшої передачі. У низці випадків можуть використовуватися пристрої стиснення даних. Вони обчислюють різні кінцеві результати, необхідні адресату. Послідовність розташування пристроїв у різних системах може відрізнитися, а функції декількох пристроїв – поєднуватися. Вони відповідають переліченим перетворенням пристрою об'єднаних підсистем подання сигналу (рисунки 1.3).

Відновлення вихідного аналогового повідомлення $\lambda^*(t)$ з допустимою похибкою за кодованими координатами реалізується на приймальній стороні. Зауважимо, що у реальних системах може не виконуватись відновлення безперервного повідомлення, оскільки реєстрація, обробка та зберігання інформації сьогодні розраховані на цифровізацію, проте можливість такого уявлення існує і повинна гарантуватися. Гарантована похибка відновлення вважається основним метрологічним показником якості системи.

Підсистема передачі містить у собі кодер і декодер каналу зв'язку. Їхня наявність полягає у передачі та прийманні обладнання й каналу зв'язку (середовищі, в якому поширюється високочастотний сигнал-переносник, разом із антенними пристроями). Кодер і декодер виконують завдастійке

кодування та декодування з метою додаткового захисту від перешкод переданих повідомлень у каналі зв'язку, однак можуть бути відсутніми, якщо похибка підсистеми передачі задовольнятиме заданим вимогам.

1.2.2 Інформаційна система управління

Сукупність об'єкта управління та пристрою управління, взаємодія яких призводить до досягнення мети називається модулем інформаційної системи управління (ІСУ). Модуль інформаційної системи управляє об'єктом за допомогою різних виконавчих механізмів. У сучасних ІСУ пристроєм управління є комп'ютер.

Іншими словами, управління технологічним процесом, обладнанням або транспортним засобом розуміється, як реалізація алгоритмів управління. На виробництві всі технологічні процеси – це об'єкти управління.

Мета управління полягає у досягненні об'єктом управління бажаного стану чи положення. Отже, незалежно від характеру технологічного процесу мета управління окреслюється як підтримка в заданих межах всіх умов, які необхідні для отримання вихідної продукції в необхідній кількості та заданої якості. Крім того, управління повинно забезпечувати технологічному процесу максимальну економічну ефективність, а разом з тим, що є не менш важливим, мінімальний негативний вплив на навколишнє середовище.

Використовуючи поняття «управління», дамо наступне визначення автоматизації виробництва.

Автоматизацією виробництва називається така його організація, при якій функції управління повністю або частково виконують технічні засоби.

Для автоматичного управління залучаються різні пристрої, зокрема, за фізичними принципами та конструкціями. Вони становлять складний комплекс взаємодіючих ланок. При цьому принципи автоматичного управління поєднуються зі спільними незалежно від того, чим і з якою метою необхідно управляти.

Будь-яка система ІСУ складається з об'єкта управління, виконавчого

механізму, датчиків і регулятора.

Об'єктом управління виступає система автоматизації, в якому відбувається процес, він підлягає регулюванню (управлінню).

Виконавчим механізмом є пристрій, який здійснює безпосередній вплив на об'єкт управління.

Регулятором є пристрій (або пристрої), за допомогою якого реалізуються керівні впливи (сигнали).

Датчиками називаються пристрої для вимірювання найважливіших параметрів технологічного процесу.

ІСУ вважаються складними системами не тільки з точки зору апаратного обладнання, їхнього конструктивного виконання, наявності значної кількості параметрів, але й з огляду на обсяги інформації. Це характеризується тим, що навіть в одному ланцюзі управління сигнал може перетворюватись багаторазово з однієї форми на іншу. Для перетворення сигналів, необхідної зміни їхніх величин, а також забезпечення заданих якостей ІСУ, застосовують узгоджувальні елементи та пристрої. Такі елементи можуть входити до найрізноманітніших ланок ланцюга управління. Під час розгляду основних принципів роботи системи автоматичного управління можна не враховувати узгоджувальні елементи, оскільки слід мати на увазі їхню обов'язкову присутність у складі основних частин системи управління. Вихідний вплив управління надходить від датчика.

1.3 Будова інформаційної системи управління гнучким виробництвом

На сучасному ГВС запроваджена ІСУ виробництвом, мета якої стежити за рухом всіх матеріальних потоків, коригувати дії окремих виробничих модулів та одиниць обладнання. Встановлення послідовності обробки об'єкта виробництва, вибору маршруту його проходження за технологічними одиницями, синхронізації безлічі матеріальних потоків, управління накопиченням і складанням, вирішення проблеми відходів та браку тощо –

все це задачі ІСУ ГВС.

Для реалізації описаної системи на практиці необхідно попередньо її змодельовати, дослідити її можливості та визначити недоліки, проаналізувати поведінку ІСУ ГВС на всіх етапах виробничого процесу. На сьогодні сучасне виробництво розглядають як комплексну інтегровану систему технологічних машин, транспортних засобів, оснащення та інших засобів виробництва, котрими повністю можна управляти за допомогою ЕОМ. У цьому випадку верстати укомплектовані обладнанням для зміни заготовок і ріжучих інструментів, до того ж, забезпечується можливість обробки різних деталей без втрати часу на переналагодження (рисунок 1.4).



Рисунок 1.4 – Модульна структура гнучкої виробничої системи

Застосовуючи гнучку виробничу систему, можна забезпечити:

- гнучкість вибору у різних заготовок для обробки протягом визначеного часу;
- можливість додавання або видалення конкретної заготовки із раніше розробленого виробничого завдання щодо обробки деталей;
- гнучкість технологічного маршруту, а саме: можливість замінити верстат для обробки конкретної деталі, наприклад, у разі зміни виробничого завдання або відмови верстата;
- можливість швидкого впровадження до виробництва конструктивних змін в деталях, які обробляються;
- можливість змін у програмі випуску конкретних деталей;
- можливість виробництва різних деталей в різних ГВС в межах одного підприємства.

Можливості ГВС є достатньо привабливими для лідерів сучасного промислового виробництва. З одного боку, вони сприяють об'єднанню високої продуктивності з малими обсягами партій деталей або виробів, з іншого – впроваджують технології без використання людської праці, що значно знижує виробничі витрати на заробітну плату. В цілому це дає можливість підприємству швидше реагувати на вимоги ринку.

Сучасне приладобудівне виробництво – це складна виробнича система, що забезпечує безперервне функціонування всіх підрозділів. Під час виготовлення приладів реалізується ланцюг технологічних процесів, у ході якого заготовки перетворюються на готові вироби.

2 ПРОЕКТУВАННЯ ІЄРАРХІЧНОЇ МОДЕЛІ БАЗИ ДАНИХ МОДУЛЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ

2.1 Концепція розподіленої обробки інформації

Концепція розподіленої обробки інформації, що реалізовано на базі сучасних персональних ЕОМ (ПЕОМ) і локальних мереж, передбачає розвиток АРМ персоналу.

АРМ називається робочим місцем персоналу автоматизованої системи управління, що обладнане засобами для забезпечення участі людини в реалізації автоматизованих функцій ІСУ. АРМ можна застосовувати під час розв'язання комплексів задач керування в різних галузях діяльності на рівні підприємства.

Створені на базі потужних ЕОМ АРМ містять у собі розвинену систему периферійного обладнання, пристрої інтерфейсу з локальними обчислювальними мережами, також центральною ЕОМ. Використовуючи АРМ, фахівець може обробляти автоматично тексти, надсилати та приймати повідомлення, що зберігаються в пам'яті ЕОМ, брати участь в онлайн-нарадах, організувати та вести особисті архіви документів на машинних носіях, реалізувати імітаційне моделювання для вивчення поведінки системи, проводити розрахунки й отримувати готові результати в табличній або графічній формі.

З огляду на те, що процес прийняття рішень у процесі керування в цілому втілюється колективом, постає потреба у проблемній спеціалізації АРМ керівного персоналу, що відповідає різним керівним ланкам і реалізованим функціям. Втілення різних фаз прийняття рішень (підготовка інформації, власне прийняття рішень, реалізація прийнятих рішень) може багато в чому співпадати з різними службами підприємства (планові служби, постачання). Це дає можливість створити гнучкі, сприятливі для перебудови

структури управління, разом з тим підвищити ефективність і оперативність роботи служб. Локальні мережі, на базі яких можуть функціонувати комплекси АРМ як у межах окремих підрозділів, так і на рівні суміжних функцій, які реалізуються різними підрозділами, є підґрунтям для взаємозв'язку окремих АРМ у системі.

Зорієнтувати АРМ на кінцевого оператора, котрий не є фахівцем у галузі обробки даних, можна забезпечити, застосувавши в АРМ алфавітно-цифрові дисплеї як основні введення та обробку інформації. Крім того, доцільно включити до складу програмного забезпечення (ПЗ) АРМ розвиненої системи діалогових процедур для спілкування з ЕОМ мову, що буде зрозумілою користувачеві.

Виникає потреба у розробці комплексу мов надвисокого рівня, що враховуватимуть специфіку об'єктів, на яких застосовуватимуться АРМ. Різні синтаксиси таких мов враховуватимуть специфіку функцій, які виконуватимуться окремими членами колективів підрозділів апарату керування.

Створені на основі ПЕОМ АРМ функціонально, органічно та фізично налаштовуються на конкретного оператора (персональний АРМ) чи групу користувачів (груповий АРМ).

Завдання організаційного керування, що розв'язуються в межах АРМ, можна умовно поділити на три класи:

– до першого класу належать повністю формалізовані задачі (бухгалтерський і складський обліки, підготовка виробництва). Для них можна розробити структуровані процедури щодо вироблення рішень. Завдання можуть розв'язуватись на ЕОМ за заздалегідь підготовленими алгоритмами без участі людини, іншими словами, алгоритм замінює людину;

– задачі другого класу характеризуються слабко структурованими процедурами вироблення рішень в умовах неповної інформації. До даного класу можна віднести завдання як поточного, так і оперативно-календарного планування виробництва, керування запасами;

– задачі третього класу вимагають виконання неструктурованих процедур вироблення рішення, зокрема, кваліфікації, творчого підходу на підставі інформованості, таланту й інтуїції людини.

Працівників організаційного керування можна розподілити за трьома групами:

- керівники (директори, головні адміністратори);
- фахівці (начальники функціональних служб, провідні фахівці);
- технічні працівники (секретарі, касири, комірники).

Керівники, зазвичай, приймають рішення стосовно завдань третього класу, рідше – першого. Фахівці розв'язують завдання другого класу, а технічні працівники – першого. Отже, за характером розв'язуваних завдань можна визначити три класи типових АРМ:

- АРМ керівника (АРМ-К);
- АРМ фахівця (АРМ-Ф);
- АРМ технічного та допоміжного персоналу (АРМ-Т).

АРМ-К може складатися з розподіленої та локальної структури. За умови розподіленої структури дисплей розташований на столі керівника, а основна функціональна частина (з додатковим дисплеєм) знаходиться у секретаря чи помічника. Локальна структура АРМ-К вимагає повної функціональної замкненості для забезпечення автономної роботи.

Вимоги до АРМ-К полягають у:

– наявності досить розвиненої БД, які постійно доповнюються оперативною достовірною інформацією. Мати доступ до бази повинен лише обмежений контингент користувачів, а тільки керівник – до окремих фрагментів бази;

– забезпеченні керівника чи його безпосередніх помічників оперативністю пошуку необхідної інформації в базі даних і наочністю подання інформації у формі, адаптованої до психологічних характеристик людини, а також високому рівні її інтеграції на екрані незалежно від джерел даних;

- наявності діалогових програмних засобів, які забезпечують прийняття рішень з максимальною адаптацією, а також регулюють організаційно-адміністративну діяльність;
- забезпеченні оперативного зв'язку з іншими джерелами інформації в межах організаційної структури разом із безпосередніми помічниками;
- забезпеченні можливості накопичення в пам'яті АРМ досвіду роботи, а також раніше прийнятих рішень.

З урахуванням особливостей та характеру роботи керівників програмне забезпечення АРМ-К містить модулі забезпечення: прийняття рішень, ділової діяльності, рутинних робіт і комунікацій.

Під час створення АРМ-Ф необхідно мати на увазі, що фахівець є професіоналом певної галузі, тому АРМ повинно дозволити йому виконувати аналітичну роботу з документами, застосовуючи різноманітну інформацію.

Його професійна орієнтація окреслює вимоги до програмного та технічного забезпечення АРМ-Ф. Фахівцю надають можливість працювати із засновницькою та персональною БД, проводити комунікаційні діалоги з додатковими джерелами інформації.

Для комп'ютерної підтримки творчого процесу система доповнена блоком моделювання аналізованих процесів з урахуванням накопиченого досвіду. Найважливішою вимогою до АРМ-Ф є забезпечення високого рівня гнучкості та багатofункціональності системи.

АРМ фахівця складається з системи забезпечення ділової та професійної діяльності, забезпечення рутинних робіт і підтримки комунікацій. За основу АРМ-Ф взято модуль забезпечення професійної діяльності, що, зазвичай, містить розвинену БД, набір програмних засобів для проведення математичних розрахунків і моделювання, засоби електронної обробки форм і ділової графіки.

База даних АРМ-Ф призначена для зберігання текстової, числової і фактографічної інформації. Зазвичай, вони ґрунтуються на типових базах даних ПЕОМ, що забезпечують мову спілкування, близьку до природньої.

Набір пакетів для математичних розрахунків і моделювання разом із експертними системами і базами знань реалізуються в АРМ-Ф для підтримки рішення формалізованих і неформалізованих задач. Найбільшою мірою вони забезпечують досягнення мети щодо підвищення ефективності та якості керування, будучи одним із нових видів програмних засобів загального призначення.

З появою більш потужних ЕОМ можливості експертних систем щодо керування виробництвом значно розширились. Вони дозволили забезпечити багато видів керівної роботи (збір інформації, загальне і календарне планування, розподіл ресурсів) шляхом комп'ютерною підтримкою прийняття рішень.

Експертна система поєднує в собі професійні знання керівників і фахівців, застосовуючи їх для формування бази знань, яка складається з набору взаємозв'язаних правил, які формалізують досвід керування. Одна з переваг експертних систем полягає у здатності щодо збирання та впорядкування даних, що відображають найбільш вагому інформацію з точки зору розв'язання завдань керування. Вона дозволяє проводити оперативний аналіз і давати експертну оцінку ситуаціям у предметній області.

АРМ технічних працівників розроблені для співробітників, які виконують рутинну роботу, що потребує певних професійних навичок. Такої категорії службовців відповідає за роботу з документами (68 % усього робочого часу) та телефонні перемовини (20 % робочого часу). Основні їхні функції полягають у введенні інформації (текстової, цифрової), оформлення документів (друк, тиражування, розсилання тощо), обробка вхідної та вихідної документації, ведення карток та архівів, контроль виконавчої діяльності. Зазначені вище види робіт можна автоматизувати з високим ступенем ефективності, застосовуючи різні АРМ-Т на базі ПЕОМ.

За умови використання АРМ-Т знижується монотонність і стомлюваність, робота стає більш різноманітною і цікавішою, розширюється функціональний діапазон виконавців, у 2-3 рази підвищується

продуктивність праці.

Особливості діяльності технічних працівників мало залежать від специфіки установи, організації, галузі. Тому активовано масовий випуск типових АРМ-Т:

- АРМ оператора текстових документів;
- АРМ архіваріуса;
- АРМ комірника;
- АРМ інспектора;
- АРМ секретаря.

Основною вимогою до програмних і технічних засобів АРМ є забезпечення максимальної «дружності» й ергономічності:

- комфортне розташування технічних засобів;
- зручна і звична клавіатура для введення інформації;
- висока якість візуальної інформації;
- простий діалог з підказками у разі неправильних дій оператора;
- наявність технічних засобів для друку та тиражування документів;
- можливість ведення архіву.

Подальший розвиток АРМ в організаційному управлінні пов'язано із застосуванням робочих станцій управління (PCY) замість ПЕОМ. Концепція PCY ґрунтується на ідеї інтегрованого «інтелектуального» призначеного для оператора інтерфейсу з маніпулювання даними та широким доступом до розвинених підсистем обслуговування. Перелічені можливості PCY дозволяють застосовувати їх як бази для створення персональних і групових АРМ. Проте, зважаючи, що PCY є універсальною системою, виникає необхідність у певній функціональній спеціалізації цих систем з її трансформацією на конкретний діловий АРМ.

PCY забезпечують 3 технологічні підсистеми:

- ділову (адміністративну) та професійну діяльності;
- прийняття рішень;
- рутинних робіт, комунікацій.

Інформаційним фундаментом цих підсистем є БД і база знань.

Технологічна основа РСУ підтримується набором таких програмних засобів:

– діловим пакетом оперативної діяльності, в якому повинні міститися різні оперативні засоби підтримки, зокрема, електронний тижневик, електронний перекидний календар, електронна записна книжка, особистий архів, картотека доручень;

– засобами пошуку та аналізу інформації, що містять різні інформаційно-довідкові системи, аналітичні картотеки, засоби екранної графіки;

– засобами моделювання керівних та економічних процесів, до складу яких входять економіко-математичні моделі, моделі прийняття рішень, електронні таблиці;

– засобами вирішення різних розрахунків – логічних задач;

– експертними системами, підтримуваними базами знань;

– засобами обробки як формалізованої, так і текстової інформації.

Традиційна технологія розв'язання задач на ЕОМ характеризується присутністю посередника між ЕОМ та кінцевим користувачем. Ним можуть бути програмісти, прикладні математики, оператори. В умовах нової інформаційної технології ЕОМ перетворюється на зручного партнера кінцевого оператора під час розв'язання завдань, які виникають в ході його професійної діяльності. Розглянемо приклад: якщо для підготовки документів за наявною технологією фахівець потребує допомоги технічних працівників з друкування, зчитування, графічних робіт, тиражування документів, то використовуючи нові інформаційні технології фахівець за допомогою інтелектуального електронного помічника може самостійно розробити документ, відтворити його та керувати тиражуванням.

Отже, обчислювальне середовище АРМ як складова людино-машинної системи за умови застосування нової інформаційної технології перетворюється на активну цілеспрямовану систему, мета якої полягає в

отриманні необхідного в постановці завдання знання, що досягається автоматичним формуванням системи, а також реалізації оптимальної в певному сенсі послідовності обчислювальних, логічних і пошукових операцій над існуючим знаннями. Крім того, забезпечується й автономна робота, й можливість зв'язку з іншими операторами в межах організаційних структур, а також з урахуванням їхніх особливостей.

2.2 Нові інформаційні технології

Проаналізувавши предметну область, можна визначити, що найбільшу увагу привертають інформаційні технології.

Вивчаючи інформаційну технологію як сукупність процесів збору, обробки, зберігання та передачі інформації, визначаємо, що предметами праці в даній технології є інформація. Інформаційній технології притаманна певна організованість та впорядкованість. Аналогічні вимоги (організованість та впорядкованість) висуваються і до інформації – об'єкта переробки.

Таким засобом організованості та впорядкованості інформації як ресурсу виступають БД, котрі становлять певним чином організовану і підтримувану мовними і програмними засобами сукупність взаємопов'язаних між собою даних, які зберігаються на технічних носіях, а також описують певну предметну область. За основу організації бази даних обрано модель даних. Завдяки моделі даних представляються безлічі даних і описуються взаємозв'язки між ними. Серед взаємозв'язків між даними можна визначити такі види. Зв'язок «один до одного» (1:1) ґрунтується на тому, що в будь-який момент часу кожному значенню (екземпляру) елемента даних *A* відповідає одне і тільки одне значення, пов'язаного з ним елемента *B*. Зв'язок «один до багатьох» (1:Б) між елементами даних *A* і *B* вказує на те, що будь-якому значенню елемента *A* відповідає більше одного значення пов'язаного з ним елемента даних *B*. Зв'язок «багато до багатьох» (Б:Б) позначає, що безлічі

значень елемента даних *A* відповідає кілька значень елемента даних *B*. Як приклад такого типу взаємозв'язків можна навести зв'язок між елементами даних «код операції» і «табельний номер робочого», враховуючи те, що безліч операцій технологічного процесу можуть виконувати різні робочі та навпаки.

Взаємозв'язки між даними повинні відображатися в БД, а засобом їхнього відображення є модель даних. Іншими словами, модель даних визначає правила, відповідно до яких дані структуруються. Найбільш поширеними є такі моделі даних: ієрархічні, мережеві та реляційні. Ієрархічна модель будується за принципом підпорядкованості між елементами даних і має вигляд деревоподібної структури, що складається з вузлів (сегментів) і дуг (гілок).

Кожен вузол дерева – це набір логічно взаємопов'язаних елементів даних, який описує конкретні об'єкти предметної області. В ієрархічній моделі дерево впорядковано за певними правилами розташування його вузлів і гілок, а саме:

- на найвищому рівні ієрархії розташований один вузол, який іменується кореневим;
- взаємозв'язки в ієрархічній моделі будуються за принципом «вихідний – породжений». Оскільки вузол другого рівня ієрархії залежить від першого, то перший рівень вважається вихідним, а другий – породженим;
- кожен вихідний рівень може налічувати кілька породжених;
- в ієрархічній моделі реалізуються типи взаємозв'язків між елементами даних (1:1) і (1:Б);
- доступ до кожного вузла (за винятком кореневого) реалізується через його вихідний вузол. З огляду на це, шляхи доступу до кожного вузла в ієрархічній моделі унікальні та лінійні за своєю структурою;
- кожен вузол може мати кілька екземплярів конкретних значень елементів даних. Кожен екземпляр породженого вузла пов'язаний з екземпляром вихідного вузла. Кожен екземпляр кореневого сегмента разом із

безліччю взаємопов'язаних примірників породжених вузлів утворює один логічний запис;

- примірник породженого вузла не може існувати без примірника вихідного вузла;

- за умови видалення примірника вихідного вузла також видаляються всі взаємопов'язані з ним екземпляри породжених сегментів.

Незважаючи на простоту проєктування та розуміння, ієрархічні моделі БД мають низку недоліків. Основним з них є обмеження форм представлення взаємозв'язків між даними. Таким чином, у моделі зазначеного типу взаємозв'язки (Б:Б) і (Б:1) не допускаються.

Другим істотним недоліком можна назвати значний час пошуку даних, оскільки доступ виконується від кореневого вузла – по ланцюжку «вихідний – породжений». Через ієрархічні впорядкованості вузлів є досить складними операції коригування бази даних, які пов'язані з додаванням і видаленням даних.

Якщо під час аналізу взаємозв'язків ми розуміємо, що породжений елемент може мати більше одного вихідного, то такі взаємозв'язки не можна описати як ієрархію. Наприклад, до виробу можуть входити різні деталі, а матеріал поставлятися кількома постачальниками тощо. Такі взаємозв'язки доцільно представляти як мережеву структуру. У мережевих моделях, на відміну від ієрархічних, можна застосувати всі типи взаємозв'язків між елементами даних. Мережева модель – це орієнтований граф з пойменованими вершинами і дугами. Вершинами графа називаються записи, що являють собою пойменовану сукупність логічних взаємопов'язаних елементів даних або агрегатів даних. Так, агрегатом даних вважається пойменована сукупність елементів даних всередині запису. Для кожного типу запису може існувати кілька екземплярів конкретних значень його інформаційних елементів. Два записи, що взаємозв'язані дугою, утворюють набір. Запис, з якої виходить дуга, йменується власником набору, а запис, до якої вона спрямована, – членом набору. Суттєва відмінність мережевої

моделі від ієрархічної полягає в тому, що кожен запис може брати участь в будь-якому наборі і виступати, як власником, так і членом набору.

Перевагою мережевої моделі можна назвати підтримування будь-яких типів взаємозв'язків, якими характеризуються об'єкти предметної області. Проте мережева модель має і низку недоліків, наприклад, складності реорганізації БД за умови появи нових елементів даних. Зауважимо, що не всі наявні системи управління БД (СУБД) можуть підтримувати всі типи взаємозв'язків. Зокрема, СУБД, які реалізують концепцію CODASIL, можуть підтримувати лише прості відношення типу (1:1) або (1:Б).

Реляційні моделі БД – це подальший розвиток теорії відносин. Реляційна модель даних виражається набором двомірних плоских таблиць, що складаються з рядків і стовпців. Будь-який первинний документ або лінійний файл можна представити як плоску двомірну таблицю. У термінології реляційних БД така таблиця позначає відношення, де кожний стовпець є атрибутом, сукупність значень одного типу (одного стовпчика) – доменом, а рядок – кортежем. Відповідно до традиційної термінології виходить, що стовпці таблиці є елементами даних, а рядки – записами. Кожній таблиці та стовпцям, відповідно, обов'язково присвоюється ім'я. Кожен атрибут таблиці повинен бути атомарним (неподільним). Порядок рядків не є суттєвим.

Кожен кортеж (запис) відносин обов'язково повинен мати ключ. Один або кілька атрибутів, за допомогою яких можна однозначно ідентифікувати кортеж відносин, називається основним (первинним) ключем. Ключі бувають простими та складеними. Простим називається ключ, який складається з одного атомарного атрибута, значення якого є унікальним. Складеним називається ключ, який складається з двох і більше атрибутів.

Для зв'язку відносин (1:1) у БД застосовують зовнішні ключі. Атрибут або комбінація атрибута відношення R_1 є зовнішнім ключем, якщо він не є основним (первинним) ключем цього відношення, але є первинним ключем для деякого відношення R_2 .

Реляційна модель даних, у порівнянні з мережевими й ієрархічними, має низку переваг. Головні переваги полягають у простоті та наочності БД для проєктування, а також теоретичних обґрунтуваннях основних принципів побудови реляційних БД. Під час проєктування БД використовується строгий формалізований метод, який ґрунтується на теорії нормалізації відносин та дозволяє будувати раціональну реляційну модель даних.

Будь-який первинний документ, у тому числі і будь-який лінійний масив, можна представити у вигляді відношення. БД певної предметної області, що реалізується через сукупність первинних документів або лінійних масивів, перетворених на відношення, може підтримуватися за рахунок реляційної СУБД. Проте у такій БД виявляється низка небажаних властивостей:

- надмірність та дублюванням інформації;
- суперечливість;
- аномалія включення нових даних і видалення старих після оновленні інформації.

Для ліквідації зазначених недоліків під час проєктування реляційних БД реалізується теорія нормалізації відносин. Теорія нормалізації відносин розуміється як набір формальних процедур, які дозволяють формувати такі набори відносин для зберігання в БД, та наділені найкращими властивостями під час редагування (оновлення, включення, видалення), до того ж, інформаційно менш надлишкові, ніж інші відношення, що несуть у собі таку ж інформацію. Основи теорії нормалізації відносин заклав Е. Ф. Кодд.

Перед початком розгляду теорії нормалізації визначимо поняття функціональної залежності між атрибутами відносин. Атрибут B зв'язку R функціонально залежить від атрибута A , якщо в кожен момент кожному значенню атрибута A відповідає не більше одного значення атрибута B . Даному твердженню відповідає такий запис математичної логіки: $A \rightarrow B$. У відносинах реляційної моделі БД виконується аналіз функціональних залежностей ключових атрибутів від ключових.

Великого значення в реляційних моделях БД набуває поняття повної функціональної залежності. Аналіз щодо повноти функціональних залежностей здійснюється для відносин зі складеними ключами.

Повною функціональною залежністю називається залежність, в якій атрибут B зв'язку R функціонально залежить від усієї множини атрибута A , проте не залежить від жодної підмножини A .

Розглянуте поняття функціональної залежності дозволяє перейти до викладу теорії нормалізації відносин. Ненормалізованими відношеннями називають відношення, що представляються в тому вигляді, в якому були отримані під час вивчення предметної області, а також виділенні її інформаційних об'єктів. Нормалізація реалізується крок за кроком.

Першим кроком нормалізації вважається подання інформаційних характеристик об'єктів предметної області як двомірні плоскі таблиці без повторюваних стовпців і рядків із атомарними атрибутами.

Другим кроком нормалізації можна трактувати як виділення ключових атрибутів відносин, аналіз функціональних залежностей та усунення неповної функціональної залежності не ключових атрибутів від складових первинного ключа шляхом виділення їх в окремі відношення. Звернемо увагу, що не ключовим атрибутом виступає атрибут, який не є складовою частиною первинного ключа. Відношення в другій нормальній формі (2НФ) – це відношення в (1НФ), де будь-який не ключовий атрибут функціонально повністю залежить від первинного ключа.

Перетворення 2НФ на третю здійснюється шляхом ліквідації транзитивних залежностей. Транзитивним зв'язком між елементами одного відношення називаються зв'язки виду: $A \rightarrow B$ і $B \rightarrow C \Rightarrow A \rightarrow C$. У даному прикладі елемент B залежить від A , а елемент C залежить від B , відповідно, кажуть, що C транзитивний залежить від A .

Отже, наступним етапом є аналіз відносин у другій нормальній формі щодо наявності транзитивних залежностей. Якщо відношення, що знаходиться в другій нормальній формі, є транзитивними залежностями не

ключових атрибутів від можливих ключів, то ліквідація таких залежностей виконується шляхом розщеплення вихідного відношення на кілька інших, в яких відсутні транзитивні залежності [10].

Узагальненням третьої нормальної форми у разі, коли відношення має більше одного потенційного ключа, виступає нормальна форма Бойса-Кодда.

Відношення знаходиться в нормальній формі Бойса-Кодда лише в тому випадку, коли детермінанти всіх функціональних залежностей є потенційними ключами.

Нормалізація відношень аж до нормальної форми Бойса-Кодда базується на понятті функціональної залежності, а також теоремі Хеза, що гарантує реалізацію декомпозиції буде без втрат інформації.

Подальша нормалізація пов'язується вже з узагальненням поняття функціональної залежності.

Атрибути (безлічі атрибутів) Y і Z багатозначно залежать від X , ($X \twoheadrightarrow Y/Z$), лише в тому разі, коли в відношенні R наявні кортежі $r_1 = (x, y, z_1)$ та $r_2 = (x, y_1, z)$ слідує, що відповідно R міститься також і кортеж до $r_3 = (x, y, z)$.

Коректність подальшої декомпозиції базується на теоремі Фейджин, в якій мова йде про те, що декомпозиція відношень на дві проекції є декомпозицією без втрат лише в тому випадку, коли у відношенні існує деяка багатозначна залежність. Якщо в плані наявна функціональна залежність, то автоматично існує й тривіальна багатозначна залежність, яка окреслюється цією функціональною залежністю.

Багатозначна залежність $X \twoheadrightarrow Y/Z$ називається нетривіальною багатозначною залежністю, якщо функціональних залежностей $X \rightarrow Y$ і $X \rightarrow Z$ не існує.

Відношення R знаходиться в четвертій нормальній формі лише в тому разі, коли відношення не має нетривіальних багатозначних залежностей.

Актуальними є і залежності спеціального виду, коли відношення не може піддаватися декомпозиції на дві проекції без втрат, проте може бути

декомпозовано на більшу кількість проєкцій. Наведені залежності називаються залежностями з'єднання і позначають узагальнення поняття багатозначної залежності.

Відношення R знаходиться в п'ятій нормальній формі лише у тому разі, коли будь-яка існуюча залежність з'єднання – тривіальна.

Дослідження та аналіз досвіду розробки інформаційного забезпечення, що орієнтоване на використання баз даних, показують, що мережеві й ієрархічні моделі більш орієнтовані на централізовану обробку даних. Реляційні моделі можна з користю застосовувати в системах централізованої та децентралізованої обробки даних. Вибір реляційних моделей для побудови розподілених баз даних у системах децентралізованої обробки інформації пояснюється тим, що реляційні бази є дуже гнучкими стосовно їхньої декомпозиції. До того ж, реляційні оператори сприяють щодо реалізації динамічної композиції децентралізованої інформації, отже, дуже зручні для організації передачі даних у мережі на відміну від інших мов не реляційного типу.

Система децентралізованої обробки інформації на промислових підприємствах орієнтується на створення АРМ планово-управлінського персоналу ґрунтуючись на використанні ПЕОМ. Для реалізації інформаційного зв'язку та взаємодії АРМ повинна бути об'єднаною обчислювальна мережа. Мова йде не про взаємодію пов'язаних мережевих засобів обчислювальної техніки, що розділені великими відстанями, а взаємодію ЕОМ, що розміщені в одній будівлі чи віддаленні одні від інших на невеликі відстань. Такі мережі отримали найменування локальних обчислювальних мереж (ЛОМ). Під поняттям ЛОМ розуміється відкрита система, пов'язаних між собою за допомогою єдиного передавального середовища ЕОМ, терміналів, периферійних пристроїв, що розташовані один від одного на невеликій відстані (кілька десятків кілометрів), а також мають високу швидкодію, низьку вартість, невеликий відсоток помилок та стандартну комунікацію даних.

Додатковими умовами до ЛОМ є:

- наявність у мережі НЕ менше двох ЕОМ;
- обробка мережею кількох вхідних інформаційних потоків, а також відсутність комунікаційної операційної системи;
- самостійне виконання кожної з ЕОМ мережі строго певного набору функцій з обслуговування та управління виробництвом незалежно від інших ЕОМ.

ЛОМ спроможні забезпечити більш надійну обробку інформації за великі мережі. У ЛОМ створюються умови для наближення кінцевого оператора до засобів обчислювальної техніки і закладаються передумови обробки інформації в момент її виникнення. Це сприяє підвищенню оперативності та обґрунтованості прийняття керівних рішень. ЛОМ, зазвичай, об'єднують однорідні персональні ЕОМ чи мініЕОМ. Вони застосовують як моноканал, так і поліканал, а їхня топологія будується за принципом загальної шини або кільця, що не вимагають створення центрів комунікації.

Однією з важливих проблем створення ЛОМ можна назвати створення розподілених баз даних. Іноді в літературних джерелах не розділяють такі поняття, як розподілена обробка даних і розподілених баз даних. Можна погодитися з думкою, що ці системи повинні містити чітке розходження. Системи розподілених БД характеризуються наявністю розподілених даних між декількома ЕОМ, а також відповідними засобами для керування цими даними. У свою чергу системи розподіленої обробки даних містять централізовану базу, де відбувається розподілення обчислювальних потужностей та програмних засобів між кількома взаємопов'язаними ЕОМ.

Вирізняють кілька типів розподілених баз даних:

- централізовану БД, в якій всі дані зосереджені в одному вузлі мережі;
- розподілену (децентралізовану) БД, в якій підмножини даних розподілені між різними вузлами мережі;
- розподілену базу з дублюванням, зокрема, БД складається з декількох

копій, які дублюються в кожному вузлі мережі;

– змішану (комбіновану) базу даних, тобто такий варіант організації, в якому існує кілька копій підмножин БД, а в кожному з вузлів розташовано її довільний фрагмент.

Централізована БД налічує низку переваг: простоту побудови БД, максимальну уніфікацію методів зберігання, зведення до мінімуму дублювання інформації, а також коригування та пошук інформації. Проте зосередження БД в одному вузлі мережі має цілу низку недоліків: за умов централізації значно збільшується час для передачі інформації, за рахунок чого різко зростає час реакції системи, до того ж, централізована структура має обмеження щодо обсягу пам'яті ЕОМ.

Розподілена база даних наділена незаперечними перевагами в порівнянні з централізованою:

– обсяг пам'яті обмежено пам'яттю не однієї ЕОМ, а сумарною пам'яттю ЕОМ, що знаходяться у всіх вузлах зв'язку;

– зменшуються витрати на передачу інформації, оскільки в кожному вузлі розташована та інформація, що необхідна для конкретного оператора і задовольняє всі його інформаційні потреби по можливості.

Проте розподілена база даних призводить до неминучого дублювання інформації, безконтрольності її зростання, суттєво ускладнюється проблема збереження несуперечності інформації. Таким чином, найбільше вона підходить для предметної області, в якій майже немає необхідності дублювання інформації в різних вузлах. У зв'язку з цим потрібна лише мінімальна кількість логічних посилань для інформаційного зв'язку різних вузлів (1:1).

При розподіленій базі з дублюванням зберігання в кожному вузлі мережі всієї БД або її дублювання розв'язують проблему надійності, доступності та ефективності вибірки даних. Проте дублювання бази даних у кожному з вузлів не завжди реалізується через обмеженість пам'яті. До того ж, за умови такого підходу не завжди узгоджувати стан багатьох копій даних

просто. Побудова розподілених баз даних із використанням стратегії дублювання є доцільною для предметної області, в якій потрібно зберігати невеликі обсяги інформації з малою інтенсивністю їхнього поновлення, проте з високими вимогами до надійності.

Змішана або комбінована БД ґрунтується на використанні комбінації двох підходів: стратегії розподілу та дублювання. Комбінована структура передбачає розчленування БД на окремі фрагменти з метою зберігання в кожному вузлі мережі в поєднанні з дублюванням окремих фрагментів. Здійснюється дублювання лише тих фрагментів бази даних, до інформації яких виконується найчастіший доступ у різних вузлах мережі. Змішаний підхід до побудови розподілених баз даних забезпечує найбільшу гнучкість і є найбільш доцільним.

Під час організації децентралізованої обробки інформації, що зорієнтована на ЛОМ, поряд із завданнями вибору оптимальної топології мережі, вибору апаратури також ключовим є завдання зі створення розподілених баз даних. Розподіл інформації вузлами обчислювальної мережі є базовим моментом у створенні ЛОМ, оскільки впливає на швидкодію, надійність, обсяги даних і обсяги пам'яті, вартість і інтенсивність потоків інформації в мережі.

Значна кількість робіт, що належать до проблеми оптимального розподілу інформації в вузлах мережі, присвячена створенню й експлуатації великих територіально розподілених мереж. Зауважимо, що в цих роботах розв'язуються загальні (глобальні) завдання проектування мережі, іншими словами містить рішення всієї сукупності проблем, вибору топології і створення розподіленої БД. За критерій оптимального проектування береться мінімум вартості створення і функціонування обчислювальної мережі. Такий критерій оптимізації є цілком виправданим під час побудови дорогих мереж великих ЕОМ, віддалених одна від одної на сотні кілометрів, причому призначених для обслуговування інформаційних потреб галузі або регіону.

До того ж, спроби постановки глобального завдання призводять до

моделі настільки великого розміру, що її розв'язання не є можливим при наявному рівні розвитку математичних методів та обчислювальної техніки.

Для формування компактної моделі, котру можна розв'язати в прийнятні терміни, слід виходити з передумови, що топологія мережі є окремим завданням, розв'язання якого вже існує, тому треба розв'язати окреме завдання формування розподілених баз даних.

Процеси проєктування розподілених баз даних можна розподілити на два етапи: розчленування та розміщення. Поняття розчленування полягає у виділенні з глобальної логічної моделі, до складу якої входить логічна структура всієї БД, окремих розділів БД з урахуванням переліку призначених для оператора застосунків, обсягів пам'яті кожної ЕОМ, часу реакції системи, надійності її функціонування, а також частоти застосування. За результатами розчленування виходять набори файлів, які підлягають зберіганню в вузлах ЛОМ. Розміщення даних по вузлах мережі залежить від обраної стратегії побудови розподілених баз даних. Якщо перевага надається централізованій стратегії, то задача розміщення зводиться до вибору вузла, в якому буде зберігатися БД. У разі вибору стратегії дублювання для кожного вузла мережі може виникнути два альтернативних варіанти: розміщувати чи не розміщувати в даному вузлі повну копію БД. Розміщення розподілених баз даних за умови використання стратегії розчленування чи змішаного підходу вважається багатоваріантною задачею.

Розподілені бази даних і побудовані на їхній базі системи децентралізованої обробки інформації – це новий ефективний засіб автоматизації керівної праці, що спрощує систему керування, підвищує продуктивність керівного персоналу, а також достовірність й обґрунтованість прийнятих управлінських рішень.

2.3 Функціональне моделювання інформаційної системи управління

Різні програмісти потребують різних логічних файлів, які виходять з однієї і тієї ж сукупності даних. Між елементами запам'ятовування даних можуть існувати різні зв'язки. Деякі бази даних міститимуть складні переплетення взаємозв'язків. Метод організації даних повинен забезпечувати можливість зручного подання таких взаємозв'язків і швидко узгоджувати внесені до них зміни. Система управління БД повинна забезпечувати можливість отримання необхідних логічних файлів з існуючих даних і чинних між ними зв'язків. При цьому висувається вимога щодо існування хоча б невеликої схожості між поданням логічного файлу в прикладній програмі та способом фізичного зберігання даних.

БД, які розроблені спеціально для використання їх оператором терміналу, забезпечують час відповіді, необхідний для діалогу “людина – термінал”. До того ж, система БД має забезпечувати відповідну пропускну здатність. У системах, які розраховані на невеликий потік запитів, пропускну здатність накладає незначні обмеження на структуру БД. У системах із великим потоком запитів, на кшталт систем резервування авіаквитків, пропускну здатність спричиняє вирішальний вплив на вибір організації фізичного зберігання даних.

У системах, які спрямовані тільки на пакетну обробку, час відповіді не є таким важливим і метод фізичної організації може вибиратися з урахуванням забезпечення ефективної пакетної обробки.

Для економії витрат на створення й експлуатацію БД перевага надається таким методам організації, що мінімізують вимоги до зовнішньої пам'яті. За умови використання цих методів фізичне подання даних у пам'яті може сильно відрізнитися від того уявлення, котре застосовує прикладний програміст. Перетворення одного подання до іншого виконується програмним забезпеченням або, якщо можливо, апаратними або мікропрограмними засобами. У таких ситуаціях доводиться вибирати між

витратами на алгоритм перетворення й економією пам'яті.

У системах обробки, що функціонували до появи систем управління БД, інформаційні фонди мали дуже високий рівень надмірності, зокрема, велику кількість надлишкових даних містила більшість стрічкових бібліотек. Навіть під час використання баз даних по мірі зростання інформації, що об'єднувалась в інтегровані БД, потенційна можливість виникнення надлишкових даних поступово збільшується. Надлишкові дані є дорогими в тому сенсі, що вони займають більше пам'яті, ніж контроль за суперечностями, що виникають з появою надлишкових даних.

Користувач БД може звертатися до неї з різноманітними питаннями щодо збережених даних. У більшості сучасних комерційних застосунків типи запитів зумовлені, це необхідно фізично, і потребують більше однієї операції оновлення. Мета організації БД повинна полягати у знищенні надлишкових даних там, де це вигідно, отже, організація даних розробляється для їхньої обробки з необхідною швидкістю. Збільшення вимог до систем пояснюються забезпеченням обробки таких запитів або формування таких відповідей, які не заплановані заздалегідь.

Якщо БД містить дані, якими послуговується значна кількість операторів, стає важливим, щоб елементи даних і зв'язку між ними не руйнувалися. Необхідно враховувати ймовірність виникнення помилок, також випадкових збоїв різного характеру. Зберігання даних, їхнє оновлення, процедури включення даних має реалізуватись таким чином, щоб система, в разі виникнення збоїв, змогла відновити ці дані без втрат, тобто обчислювальна система повинна гарантувати цілісність збережених у ній даних.

Дані в системах БД повинні зберігатися в таємниці та безпеці. Інформація, що запам'ятовується, іноді дуже важлива для її використання установою. Вона не повинна бути загубленою чи викраденою. З метою вдосконалення життєстійкості інформації в БД важливо захищати її від апаратних або програмних збоїв, кримінальних і катастрофічних ситуацій,

некоректного або некомпетентного використання особами, що можуть неправильно нею розпоряджатися.

Поняття безпеки даних – це захист даних від навмисного або випадкового доступу до них осіб, які не мають на це право, від неавторизованої модифікації даних або їхнього знищення.

Зазвичай секретність визначають як право окремих осіб або організацій визначати, як, коли і скільки відповідної інформації дозволяється передавати іншим особам або організаціям.

Організації, що експлуатують системи обробки даних протягом певного часу, витрачають значні кошти на написання програм, процедур та організацію зберігання даних. У разі, коли підприємство починає застосовувати нове ПЗ управління базами даних на обчислюваному пристрої, важливо, щоб паралельно вона могла працювати з вже встановленими програмами на даному пристрої, а опрацьовані дані можна було б перетворювати відповідним чином. Така умова потребує реалізації програмної та інформаційної сумісності, а відсутність її може стати головним стримуючим фактором під час переходу до нових систем управління БД. Проте важливо, щоб проблема зв'язку з минулим не гальмувала розвиток засобів управління БД.

Важливим є і зв'язок із майбутнім. У майбутньому дані та середовище їхнього зберігання зміняться за багатьма напрямками. З часом будь-яке підприємство зазнає змін. Особливо суттєвими ці зміни будуть для користувачів системами обробки даних.

Засоби, що застосовуються для подання загального логічного опису даних, повинні бути простими та досконалими.

Інтерфейс ПЗ повинен орієнтуватися на кінцевого оператора, а також, враховувати ймовірність того, що користувач не володіє необхідною базою знань з теорії БД.

2.4 Архітектура баз даних

Для вивчення способів організації потоків у БД потрібно визначимо кілька понять.

Ядро БД відповідає за керування даними у зовнішній пам'яті, керування буферами оперативної пам'яті, керування транзакціями і журналізацією. З огляду на зазначене вище, можна виділити такі компоненти ядра (принаймні, логічно, хоча в деяких системах ці компоненти явно присутні), зокрема, менеджер даних, менеджер буферів, менеджер транзакцій. Ядро БД містить власний інтерфейс, який не є доступним безпосередньо операторам і застосовується в програмах, розроблених компілятором SQL та утилітою БД. Ядро БД – це основна резидентна частина СУБД. Під час використання архітектури "клієнт-сервер" основною складовою серверної частини системи є ядро. Основна функція компілятора мови БД полягає в компіляції операторів мови БД на деяку виконувану програму.

Загальний перелік засобів, необхідних для роботи готового застосунку з БД, подано на рисунку 2.1.

Відповідно до поданої загальної схеми отримуємо ланцюжок «Застосунок → Ядро БД → БД». У структурі застосунку існує ланцюжок «Невізуальні компоненти → Візуальні компоненти». Невізуальні компоненти надають програмісту деякі функції щодо керування як ядром БД, так і власне даними. Використовуючи візуальні компоненти, виконується відображення даних на екрані (таблиці, розкриті списки, графіки тощо). Місцезнаходження ядра БД та власне БД в цьому ланцюжку не відображається. Місцезнаходження ядра БД і власне БД залежить від застосованої архітектури. Виділяють чотири різновиди архітектур БД:

- локальні бази даних;
- архітектура "файл-сервер";
- архітектура "клієнт-сервер";

– багатоланкова (триланкова N-tier або multi-tier) архітектура.



Рисунок 2.1 – Загальний склад засобів для роботи готового застосунку з БД

Використання тієї чи іншої архітектури має значний вплив на загальну ідеологію роботи програми, на програмний код у застосунку, на склад компонентів для роботи з БД, зокрема, це стосується невізуальних компонентів.

Під час роботи з локальними БД власне БД розташовані на тому ж ПЕОМ, що і застосунки, що виконують доступ до них. Ядро БД розміщено на ПЕОМ оператора. Застосунок відповідає за підтримку цілісності БД, а також виконання запитів до БД.

Загальну схему одного оператора архітектури продемонстровано на рисунку 2.2.

Під час роботи в архітектурі "файл-сервер" БД і застосунок розташовані на файловому сервері мережі (на кшталт, Novell NetWare).

Можливо розрахувати роботу з однією і тією ж БД на багато користувачів, коли кожен користувач зі своєї ПЕОМ активізує застосунок, розташований на мережевому сервері. Тоді на ПЕОМ оператора запускається копія застосунку. З кожним запитом до БД дані з таблиць БД переганяються на ПЕОМ оператора, незалежно від того, яка реальна кількість даних необхідна для виконання запиту. Після цього запит виконується.

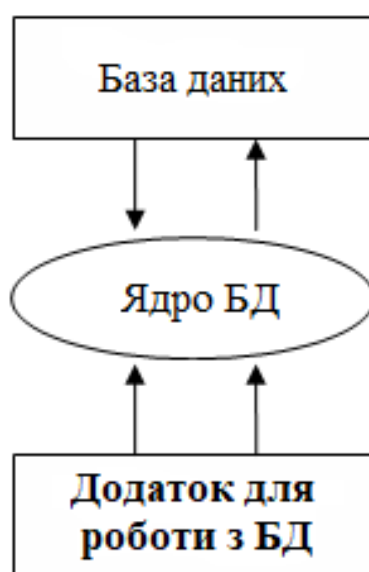


Рисунок 2.2 – Однокористувальницька архітектура під час роботи з локальними БД

Кожен користувач володіє на своєму комп'ютері локальною копією даних, які час від часу оновлюються з реальної БД, що розташована на мережевому сервері. При цьому зміни до БД, що вносить кожен користувач, до певного моменту можуть бути невідомими іншим операторам. Це робить актуальним завдання систематичного оновлення даних на ПЕОМ оператора з реальної БД. Ще одним актуальним завданням виступає блокування записів, які змінюються одним з користувачів. Це створюється для того, щоб у цей час інший користувач не зробив зміни у тих же даних. В архітектурі "файл-сервер" весь тягар реалізації запитів до БД, керування цілісністю БД лягає на застосунок оператора. БД та сервер – це пасивні джерела даних.

Загальну схему архітектури «файл-сервер» представлено на рисунку 2.3.

З точки зору архітектури кардинальних відмінностей між однокористувальницькою та архітектурою «файл-сервер» немає. В обох випадках у ролі СУБД використовуються так звані "персональні" (або "локальні") СУБД, на кшталт Paradox, dBase тощо. Власне БД у даному разі представляє собою набір таблиць, індексних файлів, файлів полів коментарів (memo-полів) тощо, котрі зберігаються в одному каталозі на диску як окремі файли.

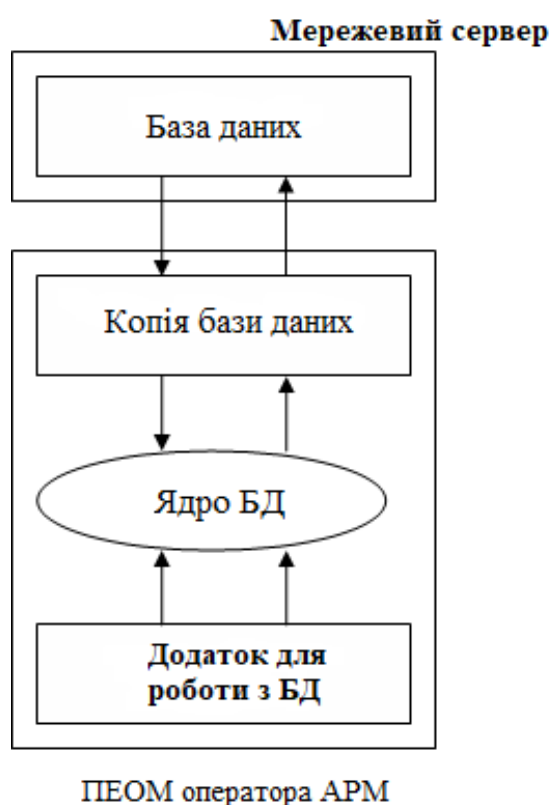


Рисунок 2.3 – Архітектура "Файл-Сервер"

Архітектура «файл-сервер» є неефективною, принаймні, в двох випадках:

– під час виконання запиту до бази даних, яка розміщена на файловому сервері. Сутність полягає у запиті до локальної копії даних на ПЕОМ

оператора. Тому перед здійсненням запиту дані в локальній копії оновлюються в повному обсязі з реальної БД. Наприклад, якщо таблиця БД містить 1000 записів, а для реалізації запиту (наприклад, видати суму премій за жовтень у відділі Y) реально потрібно 10 записів, все одно переганяються всі 1000 записів. Отже, не потрібно залучати занадто багато користувачів і запитів від них, щоб серйозно "забити" мережу. А це, як правило, не може не позначитися на її швидкодії;

– забезпечення цілісності БД проводиться із застосунків. Вони є потенційним джерелом помилок, які порушують фізичну та логічну цілісність БД, тому що різні застосунки можуть виконувати контроль цілісності БД по-різному: взаємовиключними способами чи зовсім не здійснювати такого контролю.

Набагато ефективніше керувати БД з одного місця і за єдиними законами, ніж з різних застосунків і за потенційно різними законами (все залежить від того, як застосунок написано). Таким чином, під час роботи в архітектурі «файл-сервер» безпека не є високою, до того ж, завжди присутній елемент невизначеності. Також важко забезпечити секретність і конфіденційність під час роботи з БД в архітектурі «файл-сервер», оскільки будь-хто з тих, хто має доступ до каталогу мережевого сервера, в якому зберігається БД, може змінювати таблиці БД будь-яким чином, наприклад, копіювати їх чи замінювати.

Архітектура «клієнт-сервер» розмежовує функції програми оператора, тобто клієнта і сервера.

Загальну схему архітектури «клієнт-сервер» представлено на рисунку 2.4.

Застосунок «клієнт» формує запит до сервера, на якому розміщена БД, структурною мовою запитів SQL. Віддалений сервер приймає запит, після чого переадресовує його до SQL-серверу БД. SQL-сервером називається спеціальна програма, що керує віддаленою базою даних. SQL-сервер відповідає за інтерпретацію запиту, його реалізацію в базі даних, формування

результату виконання запиту та надсилання його застосунку «клієнт». Зауважимо, що ресурси клієнтського комп'ютера не беруть участі у фізичному виконанні запиту; клієнтський комп'ютер лише надсилає запит до серверної БД і отримує результат, після чого інтерпретує його необхідним чином і надсилає користувачу.

З огляду на те, що клієнтському застосунку надсилається результат виконання запиту, мережею «подорожують» лише ті дані, що необхідні клієнту. Таким чином, знижується навантаження на мережу. У зв'язку з тим, що реалізація запиту відбувається там же, де зберігаються дані (на сервері), отже, необхідності в пересиланні великих пакетів даних немає. До того ж, SQL-сервер, за можливістю, оптимізує отриманий запит таким чином, щоб він виконувався за мінімальний час із найменшими накладними витратами.

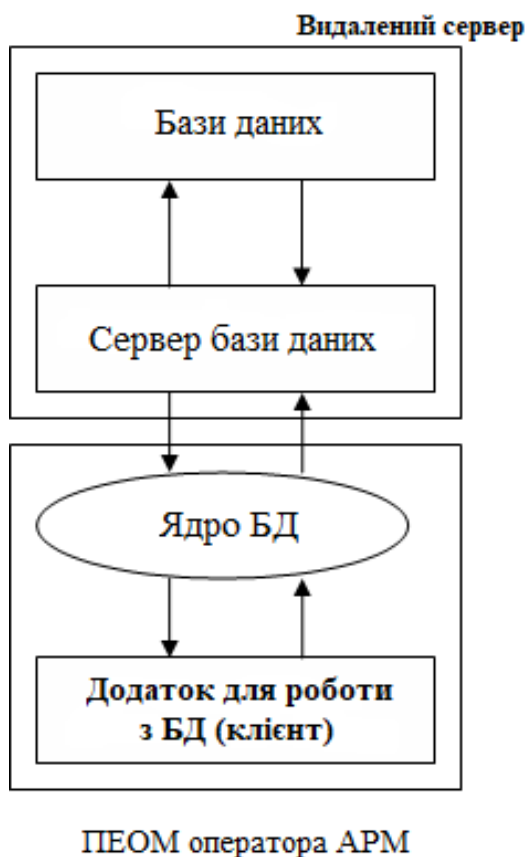


Рисунок 2.4 – Архітектура "клієнт-сервер"

Усе це підвищує швидкодію системи і знижує час очікування на результат запиту.

2.5 Моделювання даних

У розробляемій інформаційній системі підтримки використаємо реляційну БД.

Реляційною БД називається сукупність відношень, що містять всю інформацію, котра має зберігатися в БД. Проте користувачі можуть сприймати таку базу даних як сукупність таблиць. Реляційні бази даних визначаються такими характеристиками:

- кожна таблиця складається з однотипних рядків і має унікальне ім'я;
- рядки містять фіксовану кількість полів (стовпців) і значень (множинні поля і повторювані групи є неприпустимими). Іншими словами, в кожній позиції таблиці на перетині рядка і стовпця завжди в точності є одне значення або нічого;
- рядки таблиці обов'язково відрізняються один від одного хоча б одним значенням, яке дає можливість однозначно ідентифікувати будь-який рядок такої таблиці;
- стовпцям таблиці однозначно присвоюються імена, а в кожному з них розміщуються однорідні значення даних (дати, прізвища, цілі числа або грошові суми);
- повний інформаційний зміст бази даних подається у вигляді явних значень даних, і такий метод подання єдиний. Отже, не існує будь-яких спеціальних "зв'язків" або покажчиків, які з'єднують одну таблицю з іншою;
- під час виконання операцій з таблицею її рядки і стовпці можна опрацьовувати в будь-якому порядку, незважаючи на їхній інформаційний зміст. Цьому сприяє наявність імен таблиць та їхніх стовпців, а також можливість виділення будь-яких рядки або будь-якого набору рядків із зазначеними ознаками.

Структуру виробничої інформації подано на рисунку 2.5.

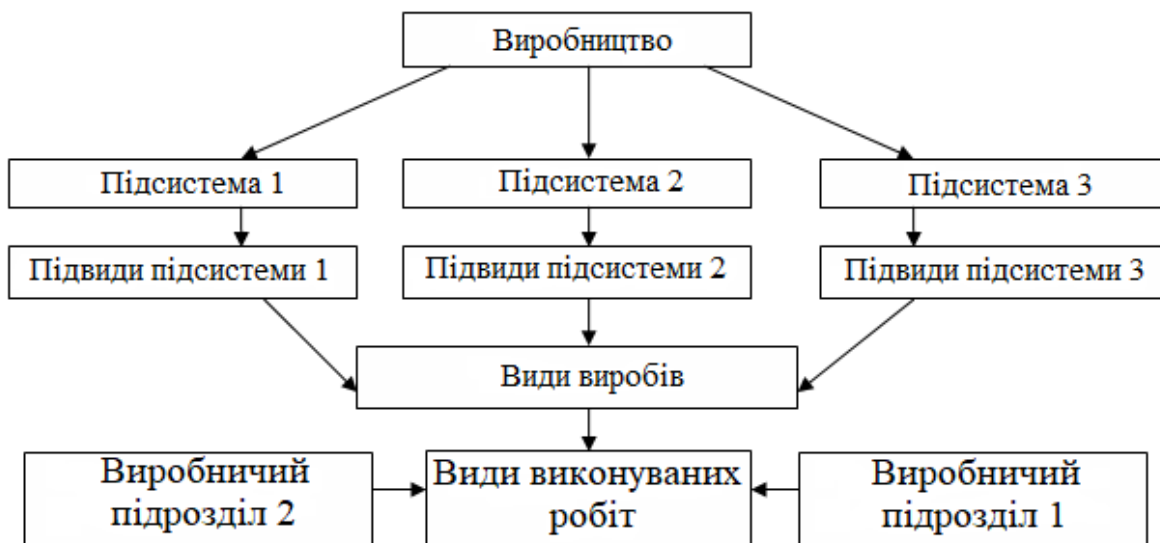


Рисунок 2.5 – Структура виробничої інформації

Програма, що розробляється, застосовує шість таблиць загальної інформації про процес виробництва, калькуляції, контроль та збут у гнучкій виробничій системі. БД у програмі містить 6 пов'язаних таблиць:

- перша – зберігає інформацію про модуль;
- друга – види підсистем;
- третя – підвиди підсистем;
- четверта – види виробів;
- п'ята – види робіт;
- шоста – зберігає інформацію про виробничі підрозділи, що цим займаються.

Структуру БД і їхній зв'язок наведено в додатку А.

Інформаційна система легко опановуватиметься користувачами, з нею зможуть працювати люди, що володіють різними рівнями знання комп'ютера.

Усі дані зберігатимуться в таблицях типу Paradox 7, це дозволяє створювати та редагувати їх у різних редакторах таблиць. Зв'язок таблиць подано в додатку Б.

В інформаційній системі використовуватиметься архітектура «файл-

сервер», оскільки вона призначена для зберігання та застосування необхідної інформації інженером-технологом під час роботи на ділянці гнучкого виробничого комплексу чи у відділі нормування на індивідуальному ПК. У свою чергу застосування архітектури «клієнт-сервер» у даному разі є недоцільним, бо з огляду на справжній стан промислової галузі нашої держави, мова йде про те, що не кожне підприємство устатковано локальною комп'ютерною мережею, а на деяких обмежуються тільки кількома ПК без будь-якого зв'язку (мережі) між собою. Проте за необхідності архітектуру «файл-сервер», змінивши деякі процедури самої програми, можна реалізувати архітектуру «клієнт-сервер» із заданими параметрами її роботи в мережі. Але в нашому випадку такої необхідності не передбачається.

3 АВТОМАТИЗАЦІЯ ІНФОРМАЦІОННОЇ СИСТЕМИ УПРАВЛІННЯ

3.1 Обґрунтування вибору мови програмування

У зв'язку з тим, що використання БД є одним з наріжних каменів, на яких базується існування різних організацій, пильну увагу розробників застосунків баз даних викликають інструменти, завдяки яким такі застосунки можна було б створювати. Вимоги, що висуваються до них, у загальному вигляді можна сформулювати як: "швидкість, простота, ефективність, надійність".

Серед широкої різноманітності продуктів для розробки застосунків Delphi – одне із простих. Delphi надають перевагу розробники з різним досвідом, звичками, професійними інтересами. За допомогою Delphi написано колосальну кількість застосунків.

За основу береться той факт, що Delphi, як жодна інша система програмування, задовольняє викладеним вище вимогам. Справді, застосунки, що базуються на Delphi, розробляються швидко, до того ж, взаємодія розробника з інтерактивним середовищем Delphi не спричиняє внутрішнього відторгнення, а навпаки, надає відчуття комфорту. Застосунки на Delphi є ефективними, якщо розробник дотримується певних правил (і часто – якщо не дотримується). Такі застосунки вважаються надійними, а під час експлуатації характеризуються передбачуваною поведінкою.

Пакет Delphi є продовженням лінії компіляторів мови Pascal корпорації Borland. У класі інструментальних засобів для програмістів-початківців продуктам компанії Borland довелося конкурувати з середовищем Visual Basic корпорації Microsoft, в якому краще були вирішені питання інтеграції та зручності роботи. Коли на початку 70-х років Н. Вірт опублікував повідомлення про Pascal, це була компактна, з невеликою кількістю

зарезервованих слів та основних понять мова програмування, спрямована на навчання студентів. Мова, на якій можуть працювати користувачеві Delphi, повинна відрізнятися від вихідної не тільки наявністю безлічі нових понять і конструкцій, але й ідейно: замість мінімізації кількості понять і використання найпростіших конструкцій (що, безумовно, добре для навчання, проте не завжди виправдано на практиці) у ній перевага надається зручності роботи професійного оператора. Наприклад, мову Turbo Pascal природно порівнювати з її найближчими конкурентами численними варіаціями на тему мови Basic (у першу чергу з Visual Basic корпорації Microsoft), а також з C++. Існує думка, що Turbo Pascal істотно перевершує Basic за рахунок повноцінного об'єктного підходу, ґрунтується на розвинених механізмах інкапсуляції, успадкування та поліморфізму. Остання версія мови, що використовується в Delphi, за своїми можливостями наближається до C++. З основних механізмів, які властиві C++, відсутнє лише множинне спадкування. Проте цим потужним механізмом породження нових класів послуговується лише невелика частина програмістів, що пишуть на C++. Переваги використання мови Pascal є очевидними: з одного боку, на відміну від Visual Basic, що засновано на інтерпретації проміжного коду, наявний компілятор, який генерує машинний код, це дозволяє отримувати значно швидші програми. З іншого боку, на відміну від C++, синтаксис мови Pascal дає можливість будувати дуже швидкі компілятори.

Середовище програмування схоже на пакет Visual Basic. У ньому є кілька окремих вікон:

- меню й інструментальні панелі;
- Object Inspector (окреслює властивості об'єкта і пов'язані з ним події);
- вікна візуального будівника інтерфейсів (Visual User Interface Builder);
- Object Browser (дозволяє вивчати ієрархію класів і переглядати списки їхніх полів, методів і властивостей);
- вікна керування проектом (Project Manager) і редактора.

Delphi має повноцінний текстовий редактор типу Brief, в якому призначення клавіш відповідають стандартам, прийнятим у Windows, а глибина ієрархії операцій Undo необмежена. Крім того, реалізовано колірне виділення різних лексичних елементів програми (це стало вже обов'язковим). Процес побудови програми є доволі простим. Потрібно вибрати форму (до поняття форми належать звичайні, діалогові, батьківські і дочірні вікна MDI), задати їй властивостей і включити до неї необхідні компоненти (видимі і, за потреби, ті, що не відображаються): меню, інструментальні панелі, рядок стану тощо, задати їх властивості, а далі написати (за допомогою редактора вихідного коду) оброблення подій.

Object Browser. Вікна типу Object Browser стали невід'ємною частиною систем програмування на об'єктно-орієнтованих мовах. Працювати з ними можна відразу після того, як буде скомпільовано застосунок.

Project Manager є окремим вікном, в якому перелічуються модулі і форми, складові проекту. При кожному модулі зазначається маршрут до каталогу, в якому розташовано вихідний текст. Жирним шрифтом виділяються змінені, проте ще не збережені частини проекту. У верхній частині вікна є набір кнопок: «додати», «видалити», «показати вихідний текст», «показати форму», «задати опції» і «синхронізувати вміст вікна з текстом файлу проекту», іншими словами, з головною програмою на мові Pascal.

Опції, разом із режимом компіляції, задаються для всього проекту в цілому. В цьому випадку традиційні make-файли, що використовуються в компіляторах мови C, є значно гнучкішими.

Різноманіття палітри об'єктів Visual Component Library (VCL) для побудови інтерфейсу, призначеного для оператора, – один з ключових чинників під час вибору інструмента візуального програмування. До того ж, для оператора має значення як кількість елементів, включених безпосередньо до середи, так і доступність елементів відповідного формату на ринку. Під час виконання запитів сервером ступінь безпеки даних підвищується суттєво,

тому що правила цілісності даних окреслюються в базі даних на сервері і є єдиними для всіх застосунків, які, використовують цю БД. З огляду на сказане вище, виключається можливість визначення суперечливих правил підтримки цілісності. Потужний апарат транзакцій, що підтримується SQL-серверами, дозволяє виключити одночасну зміну одних і тих же даних різними операторами і надає можливість відкатів до первинних значень під час внесення змін до БД, що закінчилися аварійно. Отже, функції програми-клієнта полягають у:

- надсиланні до сервера запитів;
- інтерпретації результатів запитів, отриманих від сервера, та їхнє подання користувачу в необхідній формі;
- реалізації інтерфейсу оператора.

SQL-сервер є програмою, що розташована на комп'ютері мережного сервера. На момент прийняття запиту від клієнта SQL-сервер повинен бути завантаженим. Функції сервера БД:

- приймання запитів від програм-клієнтів, інтерпретація запитів, виконання запитів у БД, відправка результату виконання запиту застосунку-клієнту;
- управління цілісністю БД, забезпечення системи безпеки, блокування невірних дій програм-клієнтів;
- зберігання бізнес-правил, найчастіше використовуваних запитів у вже інтерпретованому вигляді;
- забезпечення одночасно безпечності та відмовостійкості, що розраховані на роботу багатьох користувачів з одними і тими ж даними. В архітектурі «клієнт-сервер» застосовуються так звані «віддалені» (або «промислові») СУБД.

Промисловими їх називають через те, що саме СУБД цього класу можуть забезпечити роботу інформаційних систем підприємства середнього і великого масштабу. Локальні СУБД призначені для роботи одного оператора чи для забезпечення роботи інформаційних систем, які розраховані

на невеликі групи користувачів.

Промисловими СУБД є: Oracle, Gupta, Informix, Sybase, MS SQL Server, DB2, InterBase і низка інших.

Зазвичай, SQL-сервером керує окремий співробітник або група співробітників (адміністратори SQL-сервера). Вони керують фізичними характеристиками БД, виконують оптимізацію, налаштування та перевизначення різних компонентів БД, створюють нові БД, змінюють існуючі тощо, а також надають привілеї (дозвіл на доступ певного рівня до конкретних БД, SQL-серверу) різним операторам. До того ж, існує окрема категорія співробітників, які є адміністраторами БД. Здебільшого, це адміністратори сервера, розробники БД або користувачі, що мають привілеї зі створення, зміни, налаштування оптимальних параметрів окремих серверних БД, а також відповідають за надання прав на доступ за різними рівнями до супроводжуваних ними БД для інших користувачів/операторів.

Застосування архітектури "клієнт-сервер":

а) різко зменшується мережевий трафік:

1) знижується складність програм-клієнтів (оскільки їм вже немає необхідності забезпечувати цілісність і безпеку БД і стежити за параметрами роботи з БД, розрахованої на багато користувачів);

2) знижуються вимоги до апаратних засобів, на яких функціонують дані застосунки (до комп'ютерів користувачів-клієнтів);

б) підвищується надійність БД, її цілісність, безпека та секретність.

Отже, для створення системи інформаційної підтримки як засобу автоматизації доцільно використовувати мову програмування Delphi.

Потужність і гнучкість Delphi під час роботи з БД ґрунтується на низькорівневому ядрі – процесорі БД Borland Database Engine (BDE). Його інтерфейс з прикладними програмами має назву Integrated Database Application Programming Interface (IDAPI). На даний момент ці дві назви (BDE і IDAPI) не розрізняють, тому вважають їх синонімами.

BDE дозволяє виконувати доступ до даних як із використанням

традиційного record-орієнтованого (навігаційного) підходу, так і за допомогою set-орієнтованого підходу, що застосовується у SQL-серверах БД. Крім BDE, Delphi дозволяє виконувати доступ до БД, застосовуючи технологію (і, відповідно, драйвери) Open DataBase Connectivity (ODBC) фірми Microsoft. Проте, як показує практика, продуктивність систем з використанням BDE є набагато вищою за ті, що застосовують ODBC. ODBC драйвера працюють через спеціальний "ODBC socket", який дозволяє вбудовувати їх до BDE [6].

Усі інструментальні засоби БД Borland – Paradox, dBase, Database Desktop – застосовують BDE. Всі особливості, що містяться в Paradox або dBase, "успадковуються" BDE, тому і Delphi має такі ж особливості.

Таблиці розроблено за допомогою утиліти Database Desktop, яка входить до набору Delphi. Хоча для створення таблиць можна послуговуватись різними засобами (SQL – компонент TQuery і WISQL, компонент TTable), використання даної утиліти дозволяє створювати таблиці в інтерактивному режимі і відразу ж переглядати їхній вміст – і все це для значної кількості форматів. Це дуже зручно для локальних баз даних, зокрема Paradox і dBase. Як засіб автоматизації будемо застосуємо середу програмування Delphi.

Основний акцент цієї моделі в Delphi наголошується на максимальному використанні коду. Це дозволяє розробникам досить швидко створювати застосунки зі заздалегідь підготовлених об'єктів, а також дає їм можливість реалізовувати власні об'єкти для середовища Delphi. Жодних обмежень за типами об'єктів, які можуть створювати розробники, не існує. Справді, все в Delphi написано на ньому ж, тому розробники мають доступ до тих же об'єктів та інструментів, які також є основою для створення середовища розробки. В результаті ніякої різниці між об'єктами, що поставляються Borland або третіми фірмами, й об'єктами, що ви можете створити, немає.

Перш за все Delphi призначений для розробників-професіоналів корпоративних інформаційних систем. Тут доцільно пояснити, що конкретно

мається на увазі. Не є секретом, що деякі вдалі продукти, призначені для швидкісної розробки застосунків (RAD – rapid application development) чудово працюють під час створення доволі простих застосунків, проте розробник стикається з непередбаченими труднощами, коли намагається зробити щось насправді складне. Буває, що в продукті проявляються властиві йому обмеження тільки після певного часу.

У Delphi такі обмеження не спостерігаються. Як доказ тому наведемо той факт, що сам Delphi розроблений на Delphi. Проте Delphi реалізовано не тільки для програмістів-професіоналів.

3.2 Розробка програмного забезпечення для інформаційної системи управління

Інформаційна система підтримки написана на мові програмування Delphi 7.0 і містить п'ять модулів:

- Unit1 – головний модуль програми;
- Unit2 – модуль модульного вікна для пошуку затребуваних параметрів;
- Unit3 – модуль запиту по базі даних;
- Unit4 – модуль редагування параметрів комп'ютеризованої системи;
- Unit5 – діалогове вікно «Про програму».

Головний модуль програми – це Unit1. Після запуску програми активізуються дії щодо ініціалізації програми та створення форми головного вікна. Ці дії виконує процедура TForm1.FormCreate (Sender: Object).

Процедура combo заносить до компонента TComboBox, який має вигляд списку, значення всіх записів поля «Name» вказаної таблиці і переводить курсор на зазначену позицію в даному компоненті. Наприклад, процедура combo (MainForm.ComboBox4, DM.T4,0) заносить до компонента ComboBox4 форми MainForm значення поля «Name» таблиці DM.T4 (таблиця «Види виробів») і переводить курсор на першу позицію в компоненті

ComboBox4 (оскільки нумерація позицій у цьому компоненті починається з нульової):

```

procedure combo(CB:TComboBox;T:TTable;k:integer);
var i:integer;
begin
  CB.Items.Clear;
  with T do
    begin
      First;
      for i:=1 to RecordCount do
        begin
          if FieldbyName('Name').AsString<>"
            then
              CB.Items.Append(FieldbyName('Name').AsString);      (3.1)
          Next;
        end;
      First;
    end;
  if k>=0
    then
      CB.ItemIndex:=k;
end;

```

Процедура EditLE відображає зміни значень у компонентах EditName, EditCeh і EditPower, які виникають внаслідок переходу до різних записів таблиці «Підрозділ» (таблиця DM.T6), і відображає зміну кількості записів цієї таблиці:

```

procedure EditLE;
begin
  with MainForm do
    begin

```

```

EditName.Text:=DM.T6.FieldByName('Name').AsString;
EditCeh.Text:=DM.T6.FieldByName('Ceh').AsString;           (3.2)
EditPower.Text:=DM.T6.FieldByName('Power').AsString;
Label5.Caption:=IntToStr(DM.T6.RecordCount);
end;
end;

```

Функція `change_pr` виконує перевірку правильності введення значень під час редагування видів робіт та обладнання. У випадку, якщо вибрано поле, значення якого необхідно змінити, і введено його нове значення, то результатом виконання функції є «Істина», в зворотному випадку – «Брехня». Якщо поле не вибрано чи його нове значення не введено, на екрані з'являється повідомлення:

```

function change_pr(RB:TRadioButton;CB:TComboBox;E:TEdit):boolean;
begin
  if CB.Text=""
  then
    begin
      ShowMessage('Необхідно вибрати '"+RB.Caption+"'");
    end
  else
    begin
      if E.Text=""
      then
        begin
          ShowMessage('Необхідно ввести нове значення!');
          E.SetFocus;
        end;
      end;
    end;
  if (CB.Text<>"")and(E.Text<>"")
  then

```

```

    result:=true
else
    result:=false;
end;

```

Така процедура реалізується при натисканні на кнопку N9 (кнопка «Видалити підрозділ») головного меню форми MainForm і здійснює видалення підрозділу зі списку підрозділів. Якщо порожній список підрозділів, то з'являється відповідне повідомлення. Якщо кількість підрозділів не дорівнює нулю, відповідно, з'являється запит на видалення: «Ви дійсно хочете видалити підрозділ?». При натисканні на кнопку «Yes», підрозділ буде видалено, в зворотному випадку – підрозділ залишиться:

```

procedure TMainForm.N9Click(Sender: TObject);
begin
    if MessageDlg('Ви дійсно хочете видалити підрозділ?', mtConfirmation,
[mbYes, mbNo], 0)=mrYes
    then
        begin видалено
            with DM.T6 do
                begin
                    if RecordCount=0
                    then
                        begin
                            ShowMessage('Список підрозділ порожній!');
                            exit;
                        end;
                    Delete;
                    EditLE;
                end;
            end;
        end;
end;
end;

```

(3.4)

Відкриття закладки редагування щодо обробки дій оператора реалізується процедурою TForm5.PageControl1Changing(Sender: TObject; var AllowChange: Boolean):

```

procedure TForm5.PageControl1Changing(Sender: TObject;
  var AllowChange: Boolean);
begin
  if PageControl1.ActivePageIndex=2
  then
    Combo(Form5.ComboBox7,DM.T4,0);
  if PageControl1.ActivePageIndex<>1
  then
    begin
      vkl(6);
      combo(ComboBox1,DM.T1,0);
      combo(ComboBox2,DM.T2,0);
      combo(ComboBox3,DM.T3,0);
      combo(ComboBox4,DM.T4,0);
      combo(ComboBox5,DM.T5,0);
    end;
end;

```

Обробка помилок під час некоректних дій оператора:

```

function change(RB:TRadioButton;CB:TComboBox;E:TEdit):boolean;
begin
  if CB.Text=""
  then
    begin
      ShowMessage('Необхідно вибрати '"+RB.Caption+"'");
      result:=true;
    end
  else

```

```

begin
  if E.Text=""
  then
    begin
      ShowMessage('Необхідно ввести нове значення!');
      result:=true;
    end;
  end;
end;

```

Запит для пошуку за деяким параметром виконується процедурою TForm5.BitBtn1Click (Sender: TObject) за вибором RadioButton1 оператором:

```

procedure TForm5.BitBtn1Click(Sender: TObject);
var s_code:string;
begin
  if RadioButton1.Checked
  then
    begin
      otkl(2);
      DM.T1.Locate('Name',ComboBox8.Text,[]);
      s_code:=DM.T1.FieldName('Code').AsString;
      with DM.T6 do
        begin
          First;
          Memo1.Lines.Clear;
          k:=0;
          for i:=1 to RecordCount do
            begin
              if copy(FieldName('Code').AsString,1,1)=s_code
              then
                begin

```

(3.7)

```

        k:=k+1;
        Memo1.Lines.Add(IntToStr(k)+'
'+FieldName('Name').AsString);
    end;
    Next;
end;
end;
end;

```

Пункт меню «Допомога» відображає діалогове вікно «Про програму». У ньому виводиться версія продукту та ім'я автора програми. Повні тексти програм усіх модулів інформаційної системи подано в Додатку В.

3.3 Тестування і керівництво для інформаційної системи управління

Для тестування БД було розроблено тестовий набір даних. З огляду на те, що за умови введення даних в таблиці всі значення вважаються коректними (всі стовпці таблиць символьного типу і обмежені тільки за довжиною), під час тестування програми перевіряється правильність виконання SQL-запиту. Результат виконання тестового прикладу залежить від інформації, що розміщена в БД. Якщо під час перевірки тестовий набір реалізується правильно, то програма працює нормально.

Інформаційну систему підтримки ГВС розроблено для використання її оператором АРМ на приладобудівному підприємстві. Вона містить інформацію, з якою найчастіше мають справу фахівці в даній області.

Інформаційна система підтримки (рисунок 3.1) призначена для зберігання та використання інформації необхідної оператора АРМ під час роботи на ділянці ГВС.

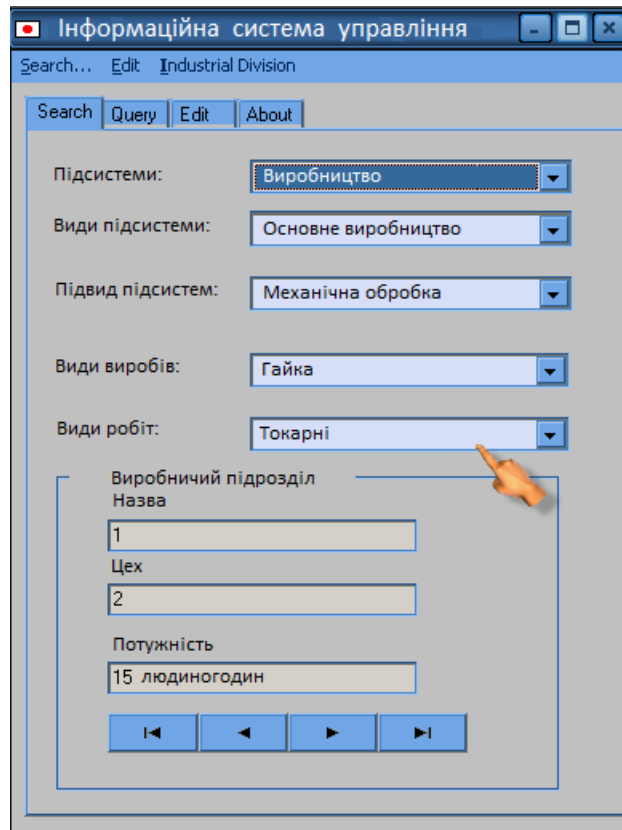


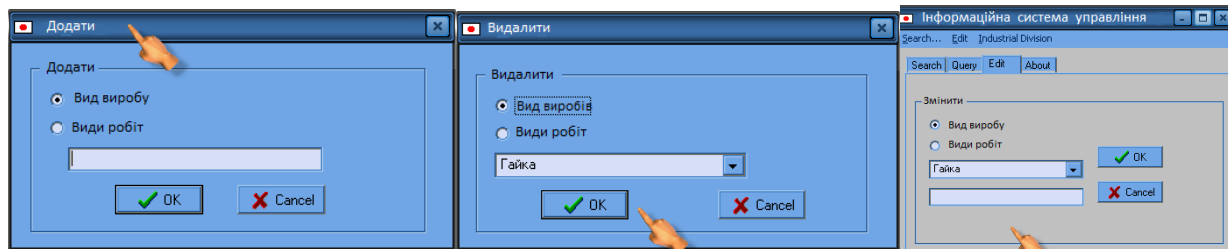
Рисунок 3.1 – Головне вікно інформаційної системи управління

Для початку роботи з підсистемою необхідно завантажити файл Kompys.exe. Поле програми розділене на дві частини, зокрема, у верхній частині відображаються елементи пошуку, а в правій частині – інформаційне вікно, в якому виводяться дані про виробничі підрозділи та їхню кількість.

Програма є дуже простою у використанні. Для відображення закладок, що зацікавили, у верхній частині вікна достатньо клікнути мишею на назву закладки. Потім відобразиться її зміст і функції, які вона здійснює в цій системі.

Усі закладки складаються з пошукових елементів та інформаційних вікон. Програма дозволяє додавати\видаляти записи за допомогою верхнього меню для видів виробів і видів робіт, які необхідні для виготовлення даного виробу, а також редагувати їхній список. Аналогічні дії можна застосувати й для самих підрозділів, а також надавати інформацію, що відповідає запиту, наприклад, висновок усіх наявних підрозділів.

У програмі є меню "Edit", до якого входять пункти "Add", "Delete", "Change", – результатом виконання яких, є відповідні дії для обраного конкретного параметра (рисунок 3.2).



а)

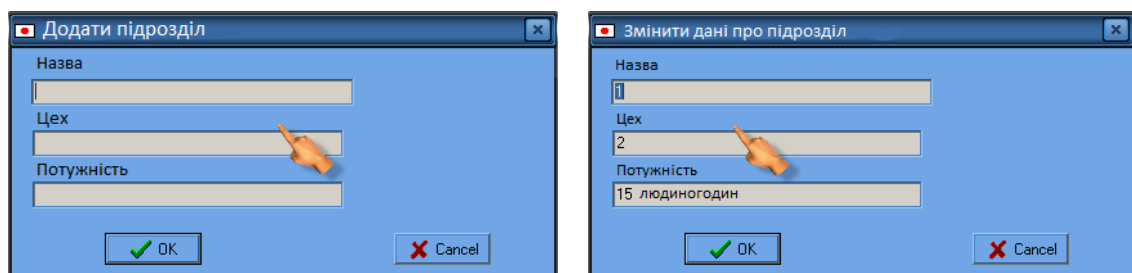
б)

в)

а – додавання; б – видалення; в – редагування

Рисунок 3.2 – Редагування даних про виробів

Для маніпулювання даними щодо власне підрозділів використовується пункт меню „Industrial Division” (додати/видалити запис, перейти до першого/останнього або до наступного/попереднього запису, редагувати запис, підтвердити або скасувати редагування запису), що розташовані безпосередньо у діалоговому вікні (рисунок 3.3).



а)

б)

а) додати підрозділ; б) змінити дані про підрозділ

Рисунок 3.3 – Редагування даних про підрозділи

Під час натискання закладки "Query" пропонується вибрати пошукові параметри, що входять до результату запиту. При натисканні кнопки "OK" будуть виведені результати запиту (рисунок 3.4).

До аварійних ситуацій у програмі можна віднести відсутність пам'яті

або спробу запуску програми не в 32-розрядній системі. В обох випадках програма не запуситься, а інформаційне повідомлення з відповідною інформацією буде виведено на екран.

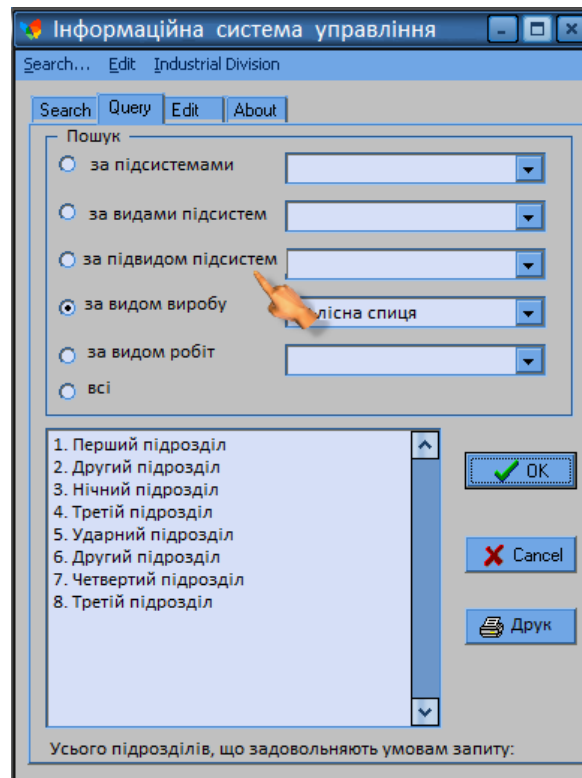


Рисунок 3.4 – Результат запиту

Якщо запустити програму без BDE, то програма не стартує, в результаті чого буде видано попередження про критичну помилку, тому для запуску програми необхідно встановити на EOM BDE, що входить до стандартного пакету Delphi.

Крім того, може виникнути аварійна ситуація у програмі, якщо один з файлів, необхідний для нормальної роботи пошукової системи, буде відсутній. У всіх зазначених вище ситуаціях відсутності файлів програма не припиняє своєї роботи, однак попередження буде виведено. Програма не вимагає особливих знань та умінь під час роботи з EOM. З програмою зможе працювати людина, котра не отримала спеціальних знань у галузі інформатики. Програма розрахована на роботу на окремому ПЕОМ.

4 ЗАХОДИ І РОЗРАХУНКИ ДЛЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕЧНИХ УМОВ ПРАЦІ

4.1 Аналіз умов праці на робочому

На робочому місці оператора ПК згідно виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, шкідливі речовини, підвищений рівень електромагнітних випромінювань радіочастот, висока напруга електричної мережі, статична електрика та інші. Робота з ПК супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці оператора ПК є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

Організація робочого місця. Приміщення, в якому знаходиться робоче місце оператора ПК, загальною площею 48 м², і висотою стелі 3,5 м. У приміщенні знаходиться 6 робочих місць з ПК. Кожне робоче місце обладнане робочим столом, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші.

4.2 Промислова безпека на робочому

Живлення ПК здійснюється від трифазної чотирьох електричної мережі змінного струму з глухо-заземленою нейтраллю і напругою 220 В, частотою 50 Гц. Згідно НПАОП 40.1-1.21-98 приміщення можна віднести до

категорії без підвищеної небезпеки, так як в приміщенні відсутні чинники, які викликають підвищену або особливу небезпеку.

Для створення безпечних умов праці необхідно провести ряд організаційних і технічних заходів. Згідно НПАОП 40.1-1.32-01 для запобігання ураження людини електричним струмом в приміщенні застосовується система занулення.

4.3 Виробнича санітарія у приміщенні

Робота оператора ПК за енерговитратами відноситься до категорії легких робіт. В таблиці 4.1 наведені оптимальні параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу [13].

Таблиця 4.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	22 – 24 °С; 40 – 60 %; до 0,1 м/с
Теплий	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	23 – 25 °С; 40 – 60 %; 0,1 – 0,2 м/с

Виміряні за допомогою приладів температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціонування повітря, або забезпечити подачу свіжого повітря системами вентиляції.

Лабораторія, де виконується розробка конструкції модуля, має наступні характеристики:

- площа приміщення 48 м² (8 м × 6 м);

- висота 3,5 м;
- кількість робочих місць – 6 шт.;
- обладнання – стіл з ПК і периферією – 6 шт.

Приміщення, відповідно до ДНАОП 0.00-1.31-99, має забезпечувати 6 м² площі та 20 м³ обсягу на одне окреме робоче місце з ПК [13]. Площа приміщення 48 м² та об'єм 168 м³, на кожне робоче місце приходиться 8 м² площі і об'єм 28 м³, тобто вимога виконана.

Приміщення з ПК повинні мати природне і штучне освітлення відповідно до ДБН В.25-28-2006 «Природне і штучне освітлення». Природне світло повинно проникати через бічні світлові прорізи, зорієнтовані, як правило, на північ або північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5 %.

Рівень загального штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності, викладеної в [13].

Розрахункова формула методу:

$$W = \frac{W_{\Sigma}}{S}, \quad (4.1)$$

де W – питома потужність, Вт/м²;

S – площа приміщення, м²;

W_{Σ} – загальна потужність освітлювальної установки Вт, яка розраховується за формулою:

$$W_{\Sigma} = W_{ce} \cdot n_{ce}, \quad (4.2)$$

де W_{ce} – потужність одного світильника, Вт;

n_{ce} – кількість світильників в приміщенні.

$$W_{\Sigma} = 100 \cdot 4 = 400 \text{ Вт}, \quad (4.3)$$

$$W = \frac{400}{48} = 8,33 \text{ Вт/м}^2. \quad (4.4)$$

Питомої потужності 8,33 Вт/м² по таблиці Б.3 із [13] відповідає освітленість в 250 лк при мінімальній допустимій освітленості 300 лк.

Отже, для створення сприятливих зорових умов в лабораторії необхідно збільшити кількість світильників або замінити лампи в світильниках на більш потужні.

4.4 Пожежна безпека виробничого приміщення

Пожежна безпека – стан об'єкта, при яким виключається можливість пожежі, а у випадку його виникнення запобігає вплив на людей небезпечних факторів пожежі й забезпечується захист матеріальних цінностей.

Пожежна безпека забезпечується системою запобігання пожежі й системою пожежного захисту. У всіх службових приміщеннях обов'язково повинен бути «План евакуації людей при пожежі», що регламентує дії персоналу у випадку виникнення вогнища загоряння, що й указує місця розташування пожежної техніки.

Горючими компонентами у виробничому приміщенні є: перегородки, двері, підлоги, ізоляція кабелів і ін.

Протипожежний захист – це комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожежі, обмеження його поширення, а також на створення умов для успішного гасіння пожежі.

Джерелами запалювання у виробничому приміщенні можуть бути електронні схеми від ПК, прилади, застосовувані для технічного обслуговування, пристрою електроживлення, кондиціонування повітря, де в

результаті різних порушень утворюються перегріті елементи, електричні іскри й дуги, здатні викликати загоряння горючих матеріалів.

У сучасних ПК дуже висока щільність розміщення елементів електронних схем. У безпосередній близькості друг від друга розташовуються сполучні проведення, кабелі. При протіканні по них електричного струму виділяється значна кількість теплоти. При цьому можливо оплавлення ізоляції. Для відводу надлишкової теплоти від ПК служать системи вентиляції й кондиціонування повітря. При постійній дії ці системи являють собою додаткову пожежну небезпеку.

Енергопостачання виробничого приміщення здійснюється за допомогою трансформаторної станції та за допомогою двигун-генераторних агрегатів. На трансформаторних підстанціях особливу небезпеку представляють трансформатори які мають масляне охолодження. У зв'язку із цим перевагу слід віддавати сухим трансформаторам.

ВИСНОВКИ

Під час роботи над кваліфікаційною роботою було проаналізовано теоретичні засади застосування інформаційних систем управління на приладобудівному виробництві. Були визначені можливості та переваги сучасних інформаційних систем управління приладобудівним виробництвом.

У результаті проведеної роботи створено і досліджено модуль інформаційної системи управління виробничого процесу на приладобудівному виробництві. Їого основним завданням є робота з БД.

Реалізація даної роботи була проведена без залучення потужних засобів роботи з БД, які дуже громіздкі, оскільки носять універсальний характер і до того ж вимагають необхідну базу знань з теорії БД.

Використання потужних засобів Delphi 7.0 зі створення додатків працюючих в операційній системі Windows і зокрема додатків БД, дозволило створити програмний продукт максимально орієнтований на кінцевого користувача, який не досвідчений в питаннях теорії БД.

Вся необхідна робота по здійсненню методів доступу до інформації, що зберігається в БД, її модифікації, підтримці БД в цілісному вигляді прихована всередині і користувачеві не потрібно знати про неї, щоб успішно вирішувати весь коло виникаючих завдань пов'язаних з використанням інформації, що зберігається в БД. Більш того, програмний інтерфейс максимально полегшує роботу щодо поводження з БД.

Всі функції виконуються СУБД були ретельно перевірені і протестовані в процесі розробки.

Розраховано штучне освітлення в дослідницькій лабораторії з розробки модуля інформаційної системи приладобудівного виробництва.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. – Введ. 2015-06-22. – К. Держстандарт України, 2017. – 29 с.
2. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології»: довід. / І. Ш. Невлюдов, А. О. Андрусевич, О. В. Токарєва, Г. В. Пономарьова. – Київ, 2018. – 320 с.
3. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарєва, С.П. Новоселов, О.В Сичова. – Харків: ХНУРЕ, 2022. – 55 с.
4. Діагностика та контроль робочих процесів: навч. посібник для студентів спеціальності «Прикладна механіка» денної та дистанційної форм навчання / В. М. Доля – Харків: НТУ «ХПІ», 2019. – 129 с.
5. Невлюдов І.Ш. Людино-машинний інтерфейс в технічних засобах автоматизації: Навчальний посібник / І.Ш. Невлюдов, О.І. Филипенко, Б.О. Шостак. – Харків : «ХТМТ», 2019. – 244 с.
6. Технічні засоби автоматизації: Підручник / І. Ш. Невлюдов, А. О. Андрусевич, О. І. Филипенко, Н. П. Демська, С. П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.
7. Mouna Fradi, Faïda Mhenni, Raoudha Gaha, Abdelfattah Mlika, Jean-Yves Choley, Category theory-based collaborative design methodology for mechatronic systems, Advanced Engineering Informatics, Volume 55, 2023, 101865, ISSN 1474-0346.

8. Craig J.J. Introduction to Robotics: Mechanics and Control / 2nd edition. – Addison-Wesley, 2019. – 450 p.— ISBN 0-201-09528-9
9. Chol-Guk Choe, Jong-Hyon Pak, Chol-Song Rim, Joint near-isometry and optimal sparse recovery: Nonuniform recovery from multi-sensor measurements, Signal Processing, Volume 208, 2023, 108980, ISSN 0165-1684.
10. Джосепс Ховс. Android Application Programming with OpenCV / Джосепс Ховс. O'Reilly Media, – 190 с.
11. Джон Кенні. A Computational Approach to Edge Detection / Джон Кенні. O'Reilly Media, – 305 с.
12. Гэри Брадскі. Learning OpenCV. Computer Vision with the OpenCV Library / підручник / Гэри Брадскі, Адріан Кехлер; – O'Reilly Media, 2018. – 580 с.
13. Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т.Є. Стищенко, Г.В. Пронюк, Н.М. Сердюк. – Харків, 2017. – 108 с.