

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

"_____" _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

«Розробка АРМ керівника маркетингового відділу»

Виконав: студент групи 2147ст

_____ Залітко А.С.
(підпис)

Керівник роботи:

к.т.н, доцент
(посада, науковий ступінь вчене звання)

_____ Гайдаєнко О.В.
(підпис)

м. Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.

" ____ " _____ 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "бакалавр"

Студенту Залітко Антону Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка АРМ керівника маркетингового відділу

Керівник роботи

Гайдаєнко Оксана Володимирівна

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем,

4. Перелік питань, що належать до розробки (найменування розділів):

Аналіз предметної області та постановка задачі, проектування АРМ, розробка АРМ, техніко-економічне обґрунтування, охорона праці

5. Перелік презентаційних матеріалів: аналіз предметної області: недоліки в системі, мета та завдання проекту, порівняльна таблиця існуючих рішень, діаграма варіантів використання, потоки даних (0-го та 1-го рівнів), потоки даних (2-го рівня), концептуальна модель БД, логічна модель БД, фізична модель БД, діаграма розгортання, результати розробки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Гайдаєнко О.В., доцент, к.т.н.		
2	Гайдаєнко О.В., доцент, к.т.н.		
3	Гайдаєнко О.В., доцент, к.т.н.		
4	Гайдаєнко О.В., доцент, к.т.н.		

7. Дата видачі завдання 14 лютого 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області		
2	Розробка концепції системи		
3	Проектування системи		
4	Реалізація проекту		
5	Вирішення питань охорони праці		
6	Виконання графічних документів		
7	Оформлення пояснювальної записки		
8	Подання дипломної роботи (ДР) на кафедральний захист		
9	Подання ДР рецензенту		
10	Подання ДР на державний захист		

Студент

(підпис)

Залітко А.С.

(ПІБ)

Керівник роботи

(підпис)

Гайдаєнко О.В.

(ПІБ)

АНОТАЦІЯ

У даній дипломній роботі представлено розробку АРМ керівника маркетингового відділу. За допомогою даної АРМ керівник зможе автоматизувати обробку статистичних даних попиту, оцінювати витрати на продукцію, створювати звіти з продажів продукції. Розробка допоможе знизити трудомісткість і підвищити достовірність, оперативність отримання результатної інформації.

В дипломній роботі виконано аналіз предметної області, техніко-економічне обґрунтування, технічний та робочий проекти, розділ з охорони праці.

Програмне забезпечення для АРМ розроблено за допомогою мови програмування Delphi, для створення бази даних було використано мову запитів SQL в СУБД MS Access.

Робота виконана на 104 сторінках машинописного тексту і містить: 31 рисунок, 27 таблиць, 2 додатки і список використаних джерел із 16 найменувань.

Робота викладена українською мовою.

ANNOTATION

In this diploma work the development of the workstation of the head of the marketing department is presented. With the help of this workstation, the manager will be able to automate the processing of demand statistics, estimate the cost of products, create reports on product sales. Development will help reduce the complexity and increase the reliability, efficiency of obtaining the resulting information.

In the diploma work the analysis of the subject area, the technical and economic substantiation, technical and working projects, the section on labor protection are executed.

The software for the workstation was developed using the Delphi programming language, and the SQL query language in the MS Access database was used to create the database.

The work was performed on 104 pages of typewritten text and contains: 31 figures, 27 tables, 2 appendices and a list of used sources from 16 names.

The work is presented in Ukrainian.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
1. ЗАГАЛЬНА ЧАСТИНА.....	11
1.1 Опис предметної області.....	11
1.2 Технічне завдання.....	12
1.3 Аналіз готових рішень	16
1.4 Організація виробництва.....	17
1.4.1 Впровадження в експлуатацію.....	17
1.4.2 Сутність, напрямки, цілі і завдання організації праці.....	20
1.4.3 Технолого-нормувальна карта розробки програмного забезпечення	22
1.4.4 Розрахунок чисельності розробників.....	24
1.5 Економічна частина	25
1.5.1 Розрахунок вартості матеріально – енергетичних витрат	25
1.5.2 Розрахунок фонду оплати праці за виконану роботу	27
1.5.3. Розрахунок ціни розробки.....	30
1.5.4 Розрахунок економічної ефективності та строку окупності програмного забезпечення	33
2. ПРОЕКТУВАННЯ АРМ.....	35
2.1 Моделювання та конструювання	35
2.1.1 Розробка моделі системи	35
2.1.2 Дослідження потоків даних.....	37
2.1.3 Концептуальна модель	43
2.1.4 Розробка діаграми станів програми.....	44
2.2 Технічний проект.....	45
2.2.1 Логічна модель даних	45
2.2.2 Розробка архітектури програми	49
2.2.3 Специфікація модулів	50
2.2.4 Алгоритм роботи програми.....	53
2.3 Робочий проект.....	54
2.3.1 Вибір засобів розробки програми.....	54
2.3.2 Фізична модель бази даних.....	56
2.3.3 Кодування	59
2.3.4 Тестування та налагодження програми	61
3. РОЗРОБКА АРМ	63
4 ОХОРОНА ПРАЦІ	67
4.1 Вступ.....	67

4.2 Основні законодавчі і нормативні документи, які регламентують Охорону праці України	67
4.3 Вимоги до охорони праці користувачів ПК	69
4.4 Загальні вимоги до виробничих приміщень ЕОМ	71
4.5 Санітарно-гігієнічні вимоги виробничого середовища ЕОМ	71
4.6 Перелік шкідливих та небезпечних факторів, які супроводжують роботу програміста	73
4.7 Ергономічні вимоги до робочого місця програміста	73
4.8 Висновки	75
ВИСНОВОК	76
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
Додаток А – Інструкція користувача	78
Додаток Б – Код розробки	85

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

UML (Unified Modeling Language) – уніфікована мова моделювання;

БД – база даних;

MS Access – система управління базою даних

SQL – (Structured query language) – мова структурованих запитів) –
декларативна мова програмування для взаємодії користувача з базами даних

IBM Rational Rose – об'єктно-орієнтований CASE-засіб проектування
інформаційних систем

APM – автоматизоване робоче місце

Borland Delphi 7 – IDE розробки програмного забезпечення

ПЗ – програмне забезпечення

ТЗ – технічне завдання

ПК – персональний комп'ютер

ВСТУП

Виробництво - це процес створення матеріальних і суспільних благ, необхідних для існування і розвитку. Створюючи певні блага люди вступають у зв'язки і взаємодію - виробничі відносини. Тому виробництво є завжди суспільним.

Сучасне виробництво розвивається в умовах науково-технічної революції. Важлива роль належить науці, яка стає безпосередньо продуктивною силою. В умовах розвинутого суспільного поділу праці виробництво складається з великої сукупності різних спеціалізованих галузей.

Зміни зовнішнього середовища та нестабільність ринкових умов стимулюють підприємства до пошуку інформації з метою зменшення невизначеності, забезпеченості стабільності та підвищення конкурентоспроможності на цій основі. Маючи достовірну, точну та, головне, вчасно отриману інформацію про стан зовнішнього і внутрішнього середовища, можна вчасно реагувати на ринкові зміни та підвищувати конкурентоспроможність підприємства.

Стрімке входження вітчизняних підприємств у ринкові умови вимагає від них мобільності та оперативності при прийнятті маркетингових рішень. Затримка необхідної інформації або її невірогідність можуть поставити підприємство на межу краху. У цих умовах інформація та інформаційні системи починають відігравати провідну роль на підприємстві і вносити істотний вклад у процес прийняття рішень. Потреба в інформації для прийняття ефективних маркетингових рішень значно зросла за умов становлення в Україні ринкової економіки, особливо в умовах невизначеності, пошуку перспективних шляхів розвитку, зниження рівня кон'юнктурних ризиків, необхідності оперативного виявлення змін у зовнішньому середовищі та своєчасного реагування на них шляхом відповідної адаптації стратегії, організаційної структури та бізнес-процесів підприємства.

Метою дипломного проекту є створення АРМ-системи що дозволить автоматизувати обробку статистичних даних попиту на продукцію та створити на

їх основі план виробництва, що в свою чергу дозволить більш точно та краще розуміти, які саме товари та в який період часу потрібно виготовляти для замовників. Дана система дозволяє оцінити затрати на продукт в процесі виготовлення та розрахувати собівартість. Також автоматизація обробки статистичних даних кондитерської фабрики призведе до скорочення витрат на облік і збільшення замовлень.

1. ЗАГАЛЬНА ЧАСТИНА

1.1 Опис предметної області

Основним завданням кондитерської фабрики є виробництво і збут кондитерських виробів. Для цього фіксується дата виробництва та дата поставки продукту, кожен продукт має свій унікальний код, назву, одиницю виміру. Продукція класифікується за типами. Виробництво забезпечується ресурсами, які поставляються постачальниками. Поставка ресурсів супроводжується накладною надходження, яка має номер, перелік ресурсів, загальну суму поставки, дату прибуття на фабрику та кількість даної сировини (в кг). Дані накладних надходження оброблюються та аналізуються.

В свою чергу поставки готової продукції в магазини (тобто до споживачів) оформляються також накладною, яка має унікальний номер, кількість товару, що поставляється, ціну за одиницю товару, дату поставки, ПІБ особи, яка відповідає за прийом товару.

Обробка даних надходження ресурсів та відвантаження готової продукції проводиться вручну. Дані обробки застосовуються для прийняття управлінських рішень направлених на підвищення ефективності діяльності підприємства.

Нагальним є питання впровадження інформаційних технологій в процеси обробки вказаної інформації. Збереження інформації в базі даних дозволить систематизувати роботу фабрики, оперативно отримувати необхідну інформацію про дату випуску, обсяги випуску, обсяги збуту продукції за низкою критеріїв, як то тип, назва, код та інші. Буде забезпечена можливість легко отримати дані про споживачів, постачальників. На основі даних про поставки готової продукції за минулі періоди можна буде скласти план на поточний рік.

З цією метою необхідно автоматизувати процес обробки накопичених даних за вказаний період з метою отримання інформації про обсяги попиту на продукцію у розрізі кожного виду, проведення аналізу статистичних даних.

Перелік основних реквізитів накладної, які підлягають статистичній обробці включає:

- 1) кількість товару відправленого на продаж;
- 2) сума товару;
- 3) дата поставки;
- 4) назва товару.

1.2 Технічне завдання

Вступ

Завданням проекту є «Розробка АРМ менеджера керівника маркетингового відділу»

Функціональне призначення

Основним завданням є створення АРМ-системи для автоматизації операцій по збору, накопиченню та обробленню інформації по відвантаженню продукції. В результаті обробки формуються статистичні дані, які використовуються при плануванні роботи фабрики на майбутній період.

Експлуатаційне призначення

Використання АРМ дозволить скоротити витрати ручної праці, підвищити точність, оперативність та достовірність даних про збут товару, скоротити час одержання аналізу даних.

Вимоги до складу виконуваних функцій

Програмний продукт повинне виконувати наступні функції:

- підтримку в актуальному стані нормативно-довідникової інформації БД;
- накопичення та упорядкування оперативної інформації;
- архівування;
- ведення журналу внесення змін в БД;
- оброблення даних поставки продукції з можливістю добору, пошуку за критеріями необхідних даних;
- занесення в БД даних з товарних накладних;

- формування даних у вигляді діаграм для звітів про результати діяльності кондитерської фабрики за певний період;
- формування статистичних даних по збуту товару за вказаний період;
- формування плану виробництва;
- формування запитів та звітів за критеріями користувача.

Вимоги до організації вхідних та вихідних даних

Вхідні дані надходять в систему у вигляді повідомлень, документів (накладних відвантаження) про кількість проданої та виготовленої продукції.

Вихідна інформація формується за результатами обробки оперативних даних, які нагромаджуються в БД і відображається у вигляді запитів та звітів на екрані та, за рішенням користувача, у друкованому форматі. План виробництва на вказаний період подається у електронній та друкованій формі.

За рішенням користувача, система архівує дані.

Основний режим використання системи - щоденна робота.

Вимоги до часових характеристик і розміру пам'яті, необхідної для роботи програми

Час реакції програми на натискання клавіш або маніпуляцій мишею не повинен перевищувати 0,25с. Реакція на команди меню не повинна перевищувати 1с.

Об'єм вільної оперативної пам'яті, необхідний для роботи програми повинен бути не менш ніж 100 Мбайт.

Час запуску програми не повинен перевищувати 5 секунд.

Дисковий простір, необхідний для збереження програми і файлу бази даних повинен бути хоча б 200 Мбайт. В архівному вигляді файли початкового коду програми, порожньої бази даних і додаткових службових файлів не повинні перевищувати 20 Мбайт.

Вимоги до надійності

Програма повинна стабільно функціонувати. При виникненні збою в роботі апаратної частини, відновлення нормальної роботи програми повинне виконуватися після перезавантаження операційної системи, повторного

виконання дій, втраченої до останнього збереження інформації на магнітному диску. Програма повинна забезпечувати коректну обробку виняткових ситуацій.

Для забезпечення надійної роботи програмного забезпечення також необхідно передбачити можливість резервного копіювання інформації. Для виконання резервного копіювання мають бути використані вбудовані можливості операційної системи, що дозволяє створювати резервні копії файлів вручну або автоматично за заздалегідь розробленим планом.

Для забезпечення надійності, необхідно перевіряти права доступу до БД та повноту і коректність вхідних даних, забезпечити захист програми від некоректних дій користувача. Для виконання робіт по актуалізації БД системи повинен бути виділений відповідальний користувач з правами адміністратора.

Вимоги до контролю вхідної і вихідної інформації

Програма повинна забезпечувати правильне введення інформації за рахунок використання, там де це доцільно, шаблонів введення, процедурного блокування введення некоректної інформації, списків та автопідстановки. Обробка виняткових ситуацій, пов'язаних з доступом до дисків, пристроїв введення-виведення інформації, повинна оброблятися програмно з виведенням відповідних інформаційних повідомлень, не призводити до блокування роботи програми.

Вимоги до часу відновлення після відмови

Час відновлення після відмови, не пов'язаної з роботою програми, повинен складатися з: часу перезавантаження операційної системи; часу повторного введення або зчитування з носіїв втрачених даних.

Вимоги до умов експлуатації та зберігання

Програмне забезпечення не потребує спеціальних навичок для роботи із ним. Програма призначена для автоматизації операцій збору та накопичення та оброблення даних на кондитерській фабриці. ПЗ повинно зберігатися у виді двох копій на флеш-накопичувачах – еталонної і робочої, на яких вказується номер версії програми, дата запису програми і дата її наступного перезапису. Періодичний перезапис інформації повинен здійснюватися відповідно до нанесеного маркування.

Умови збереження носіїв повинні відповідати загальним вимогам.

Вимоги до складу й параметрів технічних засобів

Для роботи програмного забезпечення необхідна наявність у користувача персонального комп'ютера, що має такі системні характеристики:

- IBM-сумісний комп'ютер з тактовою частотою процесора 1300МГц, та вище (Intel Pentium 4);
- об'єм оперативної пам'яті не нижче 512 МБ, та вище;
- не менше 100 МБ вільного дискового простору жорсткого диску;
- монітор;
- клавіатура та миша;
- миша;
- принтер (опціонально).

Вимоги до інформаційної і програмної сумісності

Для роботи з програмним забезпеченням необхідно встановити на комп'ютер наступні програмні продукти:

- операційну систему Windows 2000, XP, 7, 8, 8.1, 10.
- пакет Microsoft Office (MS Access).
- середовище візуального програмування Borland Delphi 7.0.

Вимоги до маркування та пакування

Для зберігання даного програмного продукту необхідно використовувати флеш-накопичувачі, отже, вимоги до маркування та упаковки пред'являлися відповідно аналогічних вимог для даних носіїв.

Вимоги до транспортування та зберігання

Програма має зберігатися і транспортуватися як всі флеш-накопичувачі згідно ДСТ. Не можна допускати деформації носія.

Дана програма встановлюється на флеш-накопичувач. Для конкретної роботи температура в приміщенні повинна бути в межах від +15 до +35 С. Відносна вологість не більше 70%.

Вимоги до програмної документації

Документація до програмного засобу повинна містити наступні документи:

- технічне завдання;
- опис програми;
- текст програми;
- програма і методика випробувань;
- інструкція користувача.

1.3 Аналіз готових рішень

В якості готових рішень для підприємств існують готові ERP-системи, які призначені для автоматизації обліку та керування. До таких систем відносяться Oracle, SAP, Microsoft Dynamics, 1C та інші.

Але не всім підприємствам потрібно використання таких систем, хоч вони й мають певні переваги, такі як:

- модульний принцип, охоплення всіх ключових процесів на підприємстві;
- доступність та узгодженість даних;
- набір потужних інструментів управління;
- контроль роботи співробітників.

Незважаючи на перераховані вище переваги, впровадження даних систем управління на малих виробництвах буде не таким доцільним, дані ERM, в більшості випадків, призначені для великих підприємств. Недоліками даних систем в такому випадку являються:

- специфіка документообігу на підприємстві, під який неможливо або важко адаптувати ERM;
- складність в навчанні персоналу для користування завпровадженою системою;
- БД з системою породжують велику кількість зв'язків, тим самим навантаження на апаратну частину ПК (або сервера) зростає, для вирішення потрібно потужна апаратна частина, що вимагає додаткових витрат;
- помилки в одній частині системи можуть призвести до втрат в іншій;

- на впровадження та супроводження потрібна велика кількість коштів.

Так наприклад мінімальний річний пакет послуг з обслуговування ІС:Підприємство коштує в середньому від 10 до 20 тисяч гривень. Причому даний пакет не дає можливості консультацій робочих на підприємстві, для цього потрібно викликати співробітника, якому теж потрібно платити. Також для використання послуг з обслуговування потрібно укласти контракт.

1.4 Організація виробництва

1.4.1 Впровадження в експлуатацію

Повний спектр робіт відповідно до побажань замовника, починаючи від інсталяції, адаптації та налагодження програмного забезпечення і до інтеграції з пристроями і передачі в експлуатацію, називається впровадженням ПЗ в систему. Час і вартість комплексу робіт залежать від безлічі факторів і критеріїв виконання, зазначених замовником або необхідних для стабільності, таких як:

- готовність персоналу компанії до переходу на нове ПЗ або його освоєння;
- наявність необхідних для виконання апаратних засобів;
- особливостей виконання роботи;
- масштабу передбачуваних дій;
- стану баз даних на поточний момент, наявності резервних копій на крайній випадок;
- наявності і працездатності каналів зв'язку.

Оскільки процедура впровадження ПО може викликати перебої в роботі компанії, процес поділяється на кілька етапів, кожен з яких має свої нюанси і здійснюється після суворого узгодження з замовником.

Обстеження компанії

Перед створенням проекту виконується дослідження поточної роботи компанії професіоналами. Після закінчення попереднього обстеження і аудиту замовник отримує рекомендації, пов'язані з розробкою технічного завдання на

проведення робіт. У ньому приділяється увага кожній деталі, детально описані вимоги по:

- підготовці та вимогам до техзасобів;
- формату зберігання і передачі даних і резервних архівів;
- складу і виконанню підготовчих робіт для об'єкта;
- конфігурації системи передачі інформації;
- роботі загального і прикладного програмного забезпечення.

Якісно складене ТЗ гарантує точність виконання робіт.

Складання контракту на виконання робіт

Контракт на проведення робіт складається за спільною згодою замовника і компанії після виконання аналізу ТЗ.

Цей період - оцінний. Оскільки план робіт призначений і терміни визначені, компанія-виконавець може оцінити всю процедуру в комплексі і визначитися з ціною. Найчастіше первинний етап проводиться безкоштовно або стає таким на підставі подальшого замовлення. Ціна на виконання робіт по інтеграції програмного забезпечення може залежати від наступних факторів:

- складу і кількості робочих місць, підсистем і модулів;
- проведення додаткових робіт по інтеграції з іншими підсистемами і системами, а також складності її виконання;
- обсягу інформації, що зберігається в БД інформації та її стану (працездатності та наявності резервних копій).

Створення групи по впровадженню ПЗ

Третій період також входить в підготовчі роботи. Компанією-виконавцем формується група впровадження програмного забезпечення і призначаються відповідальні.

Інсталяція та налагодження ПЗ

В цей період проводиться інсталяція програмного забезпечення на серверах і клієнтських машинах, підключення зв'язку, а також перевірка і налагодження робочого стану системи і її тестування під навантаженням. В стандартний перелік робіт четвертого етапу входять:

- встановлення і підготовка загальносистемного ПЗ сервера;
- інсталяція та налагодження компонентів і функцій серверної платформи;
- створення таблиць баз даних, завантаження інформації та інтеграція;
- перенесення БД (при необхідності), конвертація в потрібний формат, налагодження та створення робочих копій ПЗ, підготовка програм;
- установка і підготовка клієнтських машин (загально-прикладне і прикладне ПЗ);
- інтеграція і адаптація з уже наявними системами і платформами;
- перевірка працездатності всієї системи, тестування функціонування комплексу програмного забезпечення;
- остаточне налаштування за результатами тестування з метою отримання максимальної продуктивності і оптимізації роботи.

Завершення впровадження ПО включає виконання наступних робіт:

- навчання групи фахівців з боку замовника роботі з новим ПЗ - може проводитись віддалено або на території замовника;
- внесення змін відповідно до досвіду експлуатації замовником нового ПЗ;
- після закінчення внесення змін і усунення зауважень підписується акт здачі робіт і приймання проекту згідно ТЗ, після чого система передається замовнику і операція по впровадженню вважається завершеною.

Після інтеграції програмного забезпечення з боку замовника можуть виникнути проблеми. Це може бути людський фактор або недостатня оптимізація і інтеграція з незаявленими в ТЗ системами, які стосуються впровадженого ПО. У зв'язку з цим компанії надають технічну підтримку як своїх, так і інтегрованих сторонніми компаніями систем. Підтримка і супровід роботи серверів не входить в оплату за основні роботи, вказаними в технічному завданні.

1.4.2 Сутність, напрямки, цілі і завдання організації праці

Організація праці – це впорядкована система взаємодії працівників із засобами виробництва та один з одним у єдиному виробничому процесі.

Значення організації праці зростає в міру розвитку ринкових відносин, що сприяють відродженню конкуренції, при якій велику вагу набуває результативність праці, що робить вплив на ефективність виробництва. У міру технічного вдосконалення виробництва зростає ціна одиниці робочого часу.

Рівень використання результатів науково-технічного прогресу (зміни в техніки, технології виробництва) залежить від взаємозв'язку і пропорційності між вдосконаленням техніки і відповідними змінами в організації праці.

Праця стає складною і відповідальною, а її недосконала організація призводить до негативних наслідків.

На рівні підприємства, організація праці - це система раціональної взаємодії працівників із засобами виробництва та один з одним, заснована на певному порядку побудови і здійснення трудового процесу, спрямована на досягнення кінцевих соціально-економічних результатів.

Організація праці на підприємстві заснована на здійсненні процесу виробництва.

Процес виробництва - це органічна сполука знарядь, предметів праці і самої праці.

Процес виробництва включає в себе наступні основні елементи:

- конструкторсько-технологічну підготовку виробництва, яка полягає в розробці конструкції виробу, технології його виготовлення, підборі обладнання, інструментів;
- матеріальну підготовку виробництва, що забезпечує постачання робочих місць необхідною сировиною, напівфабрикатами, комплектуючими виробами, інструментом;
- організаційно-технологічну підготовку виробництва (розміщення засобів виробництва, послідовність здійснення технологічних процесів, операцій і робіт);

- організацію праці (організація і обслуговування робочих місць, використання раціональних методів і прийомів праці, встановлення міри праці для кожного працівника, створення сприятливих умов праці);

- організація управління виробництвом - раціональна організаційна структура управління, облік і контроль за ходом виконання технологічного процесу.

Таким чином, організація виробництва являє собою єдність всіх основних елементів, що становлять процес виробництва на підприємстві.

Головна проблема організації праці на підприємстві - це скорочення часу виробництва продукції, часу протікання виробничого процесу.

Завдання і напрямки організації праці:

- техніко-технологічні - висловлюють вплив, який чиниться організацією праці на вдосконалення структури підприємства, спеціалізацію виробництв, вибір технологічних процесів (наприклад, проектування цехів, розробка нових технологічних процесів, конструювання обладнання);

- економічні - висловлюють спрямованість організації праці на створення такої системи взаємозв'язку людини із засобами виробництва, яка забезпечує максимальну продуктивність праці, мінімальну собівартість виробництва продукції, максимальну рентабельність виробництва;

- психофізіологічні - створення сприятливих умов праці на робочих місцях, що забезпечують високу працездатність людини протягом тривалого часу (наприклад, за рахунок обґрунтованих режимів праці і відпочинку, створення нормальних умов праці);

- соціальні - їх основною метою є забезпечення змістовності і привабливості праці, забезпечення поєднання фізичних і розумових зусиль працюючих при виконанні певних робіт і функцій.

Перераховані завдання взаємопов'язані між собою і з вдосконаленням техніки і технології виробництва. На практиці неможлива ефективна реалізація рішення одного з перерахованих завдань, відокремленого від інших. Найбільша

ефективність досягається при комплексному та взаємопов'язаному вирішенні перерахованих завдань.

Основні напрямки організації праці:

- розробка раціональних форм поділу і кооперації праці заснована на відокремленні частин виробничого процесу, їх технологічних особливостей і кваліфікаційних вимог до робітників;
- організація робочих місць - оснащення засобами виробництва, предметами праці, інструментом, їх раціональне розміщення, що сприяє застосуванню раціональних методів і прийомів праці;
- обслуговування робочих місць - пошук оптимальних варіантів взаємодії основних і допоміжних робітників, що дозволяє забезпечити безперебійне постачання робочих місць матеріалами, послугами;
- оптимізація трудового процесу шляхом впровадження раціональних методів і прийомів праці, що забезпечують максимальну економію робочого часу при мінімальних витратах фізичної енергії;
- створення сприятливих умов праці на робочих місцях, раціоналізація режимів праці та відпочинку;
- атестація і раціоналізація робочих місць - періодичний облік і оцінка, атестація робочих місць на предмет їх відповідності сучасним вимогам;
- нормування праці - діяльність з управління працею і виробництвом, спрямована на встановлення необхідних витрат і результатів праці, співвідношення між чисельністю працівників і кількістю одиниць обладнання.

1.4.3 Технолого-нормувальна карта розробки програмного забезпечення

Технолого-нормувальна карта (табл.1.1) розглядається як основний документ щодо поліпшення як в цілому організації праці на робочому місці, так і його нормування. При такому варіанті нормування праці значення норм часу пов'язують з технологією виконання робіт. У додатку до технолого-нормувальної карти наводиться рекомендована оснащеність робочого місця обладнанням,

машинами, механізмами, технологічним оснащенням, ручними і механізованими інструментами, контрольно-вимірювальними приладами та іншими засобами, а також планування, система обслуговування, умови і безпеку праці на робочому місці. Трудовий процес, затверджений в технологічній, нормувальній карті, повинен стати еталоном при виконанні роботи, виключивши її виконання по інтуїції, досвіду. Пропонується повне охоплення даними картами всіх робочих місць.

Таблиця 1.1 Технологічна – нормувальна карта

Найменування операції	Тривалість виконання в нормо-годинах
1) Дослідження предметної області	20
2) Визначення вимог користувача до розробки	30
3) Розробка технічного завдання та його узгодження з замовником	79
4) Проектування АРМ	
4.1) Розробка документу моделювання та конструювання	92
4.2) Розробка документу технічного проекту	146
4.3) Розробка документу робочого проекту	100
5) Тестування та налагодження ПЗ	80
6) Підготування інструкції користувачу АРМ	75
7) Внесення змін за результатами досліджень експлуатації ПЗ	95
8) Супроводження	60
Всього	777

1.4.4 Розрахунок чисельності розробників

У плануванні діяльності підприємств важливе місце належить визначенню потреби в трудових ресурсах. Планування трудових ресурсів на підприємствах починається з аналізу забезпеченості робочих місць працівниками відповідної кваліфікації та рівня їх використання. Чисельність персоналу організації залежить від характеру, масштабу, складності, трудомісткості виробничих (чи інших) і управлінських процесів, ступеня їхньої механізації, автоматизації, комп'ютеризації. Ці фактори визначають її нормативну (планову) величину. Більш об'єктивно персонал характеризується обліковою (фактичною) чисельністю, тобто кількістю співробітників, що офіційно працюють в організації в даний момент. Порядок обліку персоналу визначений Інструкцією зі статистики персоналу і заробітної плати робітників та службовців на підприємствах, в установах і організаціях. Інструкція регламентує методи визначення облікового складу, середньооблікової чисельності працівників, перелік зайнятих в основній і неосновній діяльності, категорії персоналу, порядок обліку прийому і вибуття кадрів та інші питання.

Розрахунок необхідної чисельності робітників здійснюється за формулою:

$$N = \frac{T}{F_{\text{еф.}} \cdot K.V.H.} \quad (1.1)$$

Де T - трудомісткість виготовлення продукції або виконання робіт в нормогодинах (в нормо годинах)

$F_{\text{еф.}}$ – ефективний фонд робочого часу в плановому періоді (год.)

$K.V.H$ – коефіцієнт виконання норм, від 1.11 – 1.15

$$F_{\text{еф.}} = D \cdot \varphi \cdot \eta \quad (1.2)$$

Де φ – тривалість робочої зміни (в годинах)

η – коефіцієнт, що враховує втрати робочого часу, варіюєця 0.93 – 0.96

D – кількість робочих днів

D - 35 днів

T – 777 нормо-годин

φ = 8 годин

$$N = ?$$

$$F_{\text{эф.}} = 35 \times 8 \times 0.96 = 268,8 \text{ год}$$

$$N = \frac{777}{268,8 \times 1,14} = 2,536 = 3 \text{ чол.}$$

Для розробки програмного забезпечення нам необхідно три працівники за таким фахом:

- 1) Інженер – програміст
- 2) Тестувальник програмного забезпечення
- 3) Дизайнер

Інженер – програміст – фахівець, що застосовує принципи програмної інженерії до проектування, розробки, тестування та оцінки програмного забезпечення та комп'ютерних систем.

Тестувальник програмного забезпечення - фахівець, який займається перевіркою працездатності, якості, безпеки та доступністю використання програмного забезпечення, сайтів, додатків тощо.

Дизайнер – фахівець, який проектує дизайн проекту: розміщення компонентів на формі, кольорову гамму програми чи веб – сайту, створює правильне оформлення для зручного використання продукту.

1.5 Економічна частина

1.5.1 Розрахунок вартості матеріально – енергетичних витрат

Кошторис виробництва - це витрати підприємства, зв'язані з основною його діяльністю за певний період, незалежно від того, відносять їх на собівартість продукції в цьому періоді чи ні. Отже, кошторис виробництва і собівартість загального обсягу продукції, як правило, не збігаються. Кошторис виробництва складають за економічними елементами.

Матеріальні витрати як елемент кошторису складаються з витрат на: сировину й основні матеріали, які є матеріальною субстанцією продукції; вироби, що їх треба купити для укомплектування продукції (двигуни, прилади, тощо); покупні напівфабрикати (штампування, відливки, поковки та ін.);

виробничі послуги сторонніх підприємств і організацій, необхідні для виготовлення продукції; допоміжні матеріали, які використовуються в технологічному процесі (кріпильні деталі, фарби, інструмент і т.п.) або потрібні для його обслуговування (ремонт, експлуатації устаткування та ін.), на господарські та управлінські потреби (утримання будівель, канцелярські товари тощо); паливо та енергію зі сторони (електроенергію, пар, газ тощо). Витрати на власне виробництво енергії включаються в кошторис за окремими елементами; пошук і використання природної сировини (відрахування на геологорозвідувальні роботи, рекультивацію землі, плата за деревину та ін.). Витрати на матеріали обчислюються на підставі норм їхнього витрачання та цін з урахуванням транспортно-заготівельних витрат, які не є складовими інших елементів кошторису (плата за транспортування, вантажно-розвантажувальні роботи, комісійні заготівельним організаціям та ін.). Із вартості матеріалів віднімають вартість відходів за ціною використання чи продажу.

Таблиця 1.2 Розрахунок вартості матеріальних витрат

Найменування матеріалу	Одиниці виміру	Ціна за одиницю	Загальна вартість
Флеш - накопичувач	Штук	150, 00 грн.	300, 00 грн.
Інтернет	Мбіт/сек	80, 00 грн.	80, 00 грн.
Роутер	Штук	400, 00 грн.	400, 00 грн.
Папір	Штук	1, 00 грн.	350, 00 грн.
Картриджі	Штук	300, 00 грн.	900, 00 грн.
Всього			2030, 00 грн.

Транспортно - заготовельні витрати складають 20% від суми матеріальних витрат, тобто $2030,00 \times 20:100 = 406,00$ грн.

Разом матеріальні витрати з ТЗВ складають:

$$Ц_{заг} = Ц_{м} + Ц_{тзр} \quad (1.3)$$

$$Ц_{заг} = 2030,00 + 406,00 = 2436,00 \text{ грн}$$

Розрахунок вартості електричної енергії

Кількість комп'ютерів – 3 шт.

Кількість принтерів – 1 шт.

Вартість електрики розраховується за формулою:

$$V_{\text{ел.}} = N \times D \times \gamma \times C \quad (1.4)$$

Де N – потужність ПК/Принтера (кВт)

D – кількість робочих днів (35)

γ – тривалість робочої зміни (8 год)

C – вартість 1 кВт/год. (1,78 грн.)

$$N_{\text{ПК}} = 0,22 \times 3 = 0,66 \text{ кВт}$$

Принтер працює 3 години в день.

$$N_{\text{принт.}} = 0,55 \times 3 = 1,65 \text{ кВт}$$

Відповідно розрахуємо вартість споживання електроенергії комп'ютерами:

$$V_{\text{ел.ПК}} = 0,66 \times 35 \times 8 \times 1,78 = 328,94 \text{ грн.}$$

Вартість споживання електроенергії принтера:

$$V_{\text{ел.Принт.}} = 1,65 \times 3 \times 1,78 = 102,79 \text{ грн.}$$

Сума вартості електроенергії дорівнює:

$$328,94 \text{ грн.} + 102,79 \text{ грн.} = 431,73 \text{ грн.}$$

Загальна вартість матеріально – енергетичних витрат становить:

$$2436 \text{ грн.} + 431,73 \text{ грн.} = 2867,73 \text{ грн.}$$

1.5.2 Розрахунок фонду оплати праці за виконану роботу

Заробітна плата включає всі форми оплати праці штатного й позаштатного виробничого персоналу підприємства, тобто персоналу, що зайнятий виробництвом продукції, обслуговуванням виробничого процесу та управлінням. Не включаються в собівартість виплати працівникам, що фінансуються із прибутку або з інших джерел спеціального призначення.

Відповідно до Закону України “Про оплату праці” заробітна плата - це винагорода, обчислена, як правило, у грошовому виразі, яку за трудовим договором власник або уповноважений ним орган виплачує працівникові за виконану ним роботу. Розмір заробітної плати залежить від складності та умов

виконуваної роботи, професійно - ділових якостей працівника, результатів його праці та господарської діяльності підприємства .

В ринковій економіці заробітна плата як соціально-економічна категорія повинна виконувати такі функції:

- відтворювальну – передбачає встановлення норм оплати праці на такому рівні, який забезпечує відтворення робочої сили відповідної кваліфікації;
- стимулюючу – стимулювання кількості, якості та результатів праці;
- регулюючу – як засіб регулювання ринку праці, розподілу і перерозподілу по регіонах країни, галузям економіки з урахуванням ринкової кон'юнктури;
- соціальну – забезпечення соціальної справедливості, однакової винагороди за однакову працю.

Заробітна плата складається з основної (постійної) і додаткової (змінної) частин, а також заохочувальних та компенсаційних виплат.

Основна заробітна плата - це винагорода за виконану роботу відповідно до встановлених норм праці (норми часу, виробітку, обслуговування, посадові обов'язки). Вона встановлюється у вигляді тарифних ставок (окладів) і відрядних розцінок для робітників та посадових окладів для службовців.

Додаткова зарплата - це винагорода за працю понад встановлені норми, за трудові успіхи та винахідливість і за особливі умови праці. Вона включає доплати, надбавки, гарантії і компенсаційні виплати, передбачені чинним законодавством.

Інші компенсаційні та заохочувальні виплати - це винагорода за підсумками роботи за рік, премії за спеціальними системами і положеннями, компенсаційні та інші грошові й матеріальні виплати, які не передбачені актами чинного законодавства або які здійснюються понад встановлені зазначеними актами норми.

Особливе місце в системі оплати праці посідає мінімальна заробітна плата, яка є державною соціальною гарантією, обов'язковою на всієї території України для підприємств всіх форм власності і господарювання.

Мінімальна заробітна плата - це законодавчо встановлений розмір заробітної плати за просту, некваліфіковану працю, нижче якого не може проводитися оплата праці за виконану працівником місячну, годинну норму праці (обсяг робіт). До мінімальної заробітної плати не включаються доплати, надбавки, заохочувальні та компенсаційні виплати.

1. Трудомісткість (Т) - показник, що характеризує витрати робочого часу на виробництво певної споживної вартості або на виконання конкретної технологічної операції.

Трудомісткість становить 777 годин.

2. $C_{ч1/2/3/сер.роб}$ – середня годинна тарифна ставка, встановлюється керуючим, в залежності від професійності працівників.

$$C_{ч1.роб} = 110,00 \text{ грн.}$$

$$C_{ч2.роб} = 99,00 \text{ грн.}$$

$$C_{ч3.роб} = 100,00 \text{ грн.}$$

$$C_{ч.сер} = \frac{C_{ч1.роб} + C_{ч2.роб} + C_{ч3.роб}}{3} \quad (1.5)$$

$$C_{ч.сер} = \frac{110,0 + 99,0 + 100,0}{3} = 103,00 \text{ грн.}$$

3. Фонд оплати праці (ФОП) - основний економічний показник з оплати праці, що широко використовується у господарській практиці, статистичній звітності, економічному аналізі.

$$\text{ФОП} = Z_{осн.} + Z_{дод.} \quad (1.6)$$

Де: $Z_{осн}$ – основна заробітна плата.

$$Z_{осн} = C_{ч.сер} \times T \quad (1.7)$$

$$Z_{осн} = 103,0 \times 777 = 80\,031,00 \text{ грн.}$$

$Z_{дод.}$ – додаткова заробітна плата, (премія) що становить 25 – 40% від основної заробітної плати, в нашому випадку премія – 27%.

$$Z_{дод.} = \frac{\Phi_{прем} \cdot Z_{осн}}{100} \quad (1.8)$$

$$Z_{дод.} = \frac{27 \cdot 80\,031,0}{100} = 21\,608,37 \text{ грн.}$$

$$\text{ФОП} = 80\,031,0 + 21\,608,37 = 101\,639,37 \text{ грн.}$$

4. Єдиний соціальний внесок (ЄСВ) - обов'язковий платіж до системи загальнообов'язкового державного соціального страхування, що справляється в Україні з метою забезпечення страхових виплат за поточними видами загальнообов'язкового державного соціального страхування (становить 22% від ФОП).

$$\text{ЄСВ} = \frac{22 \cdot \text{ФОП}}{100} \quad (1.9)$$

$$\text{ЄСВ} = \frac{22 \cdot 101639,37}{100} = 22360,661 \text{ грн.}$$

5. $Z_{\text{сер.}}$ - Середня заробітна плата одного працівника за день

$$Z_{\text{сер.}} = \frac{\text{ФОП}}{D \cdot N}$$

$$Z_{\text{сер.}} = \frac{101\,639,37}{35 \cdot 3} = 967,994 \text{ грн.}$$

N - Число робітників – 3

D - Тривалість виконання – 35 днів

$Z_{\text{сер.міс}}$ – середня зарплата одного працівника за місяць

$$Z_{\text{сер.міс}} = Z_{\text{сер.}} \times 22$$

$$Z_{\text{сер.міс}} = 967,994 \times 22 = 21\,295,868 \text{ грн.}$$

1.5.3. Розрахунок ціни розробки

Собівартість продукції - це виражені в грошовій формі сукупні витрати на підготовку і випуск продукції. В більш узагальненому вигляді собівартість можна визначити як грошовий вираз величини ресурсів, використаних з конкретною метою. Таке визначення собівартості містить у собі три важливих моменти:

- собівартість відображає, скільки і яких ресурсів було використано у виробництві;
- величина використаних ресурсів представлена в грошовому виразі, що дозволяє розраховувати загальну вартість ресурсів;
- конкретна мета використання ресурсів зумовлює необхідність чітко встановити об'єкт собівартості (виробництво чи реалізація).

Виробнича собівартість продукції - це виражені у грошовій формі поточні витрати підприємства на виробництво продукції, виконані роботи та надані послуги.

Таблиця 1.3 Розрахунок ціни розробки програмного забезпечення

Статті калькуляції	Методика розрахунку	Сума в гривнях
1) Основні і додаткові матеріали	За розрахунками в пункті 5.1	2030, 00
2) Транспортно -заготівельні витрати	За розрахунками в пункті 5.1	406,00
3) Енергетичні витрати	За розрахунками в пункті 5.1	431,73
4) Основна заробітна плата	$Z_{\text{осн}} = C_{\text{ч.сер}} \times T$	80 031,00
5) Додаткова заробітна плата	27% від основної заробітної плати	21608, 37
6) ЄСВ	22% від ст. 3+4	22360, 66
7) Амортизація відрахувань	За розрахунком кумулятивним методом	2620,36
8) Загальні виробничі витрати	90% від основної заробітної плати	72027, 90
9) Виробнича собівартість	Сума статей 1 - 7	201 516,02
10) Адміністративні витрати	73% від основної заробітної плати	58 422, 63
11) Повна виробнича собівартість	Сума статей 8 - 9	259 938, 65
12) Рівень рентабельності	31% від статті 10	80580, 98
13) Ціна розробки	Сума статей 10 – 11	340 519,63

--	--	--

Використовуючи кумулятивний метод, річну суму амортизації визначають як добуток вартості, яка амортизується, та кумулятивного коефіцієнта. Кумулятивний коефіцієнт розраховується діленням кількості років, що залишаються до кінця терміну корисного використання об'єкта основних засобів, на суму кількості років його корисного використання.

Кумулятивний метод належить до групи прискорених методів амортизації. Отже, всі принципові особливості цієї групи подібні та полягають у тому, що значна частка амортизаційних відрахувань накопичується протягом перших років експлуатації. Тому кумулятивний метод застосовують у тих самих випадках, що й інші два методи прискореної групи.

Кількість комп'ютерів – 3 шт.

Кількість принтерів – 1 шт. (не враховується)

Вартість комп'ютера – 10000, 0 грн.

Ліквідаційна вартість комп'ютера – 500, 0 грн.

Враховуючи те, що комп'ютер використовується перший рік:

$$BA = BP - BL \quad (1.10)$$

BA – вартість, яка амортизується

BP – первісна вартість

BL – ліквідаційна вартість

$K_{\text{кум 1 рік}}$ – кумулятивний коефіцієнт

$$K_{\text{кум 1 рік}} = \frac{2}{1+2} = \frac{2}{3} = 0,67$$

$$BA = 3 \times 10000, 00 - 500, 00 = 29500, 00 \text{ грн.}$$

A - сума амортизаційних відрахувань

$$A_{\text{рік}} = 29500, 00 \times 0,67 = 19765, 00 \text{ грн.}$$

$$A_{\text{міс}} = 19765, 00 : 12 = 1647, 08 \text{ грн.}$$

$$A_{\text{день}} = A_{\text{міс}} : 22$$

$$A_{\text{день}} = 1647,08 : 22 = 74, 87 \text{ грн.}$$

$$A_{\text{заг.}} = A_{\text{день}} \times 35$$

$$A_{\text{заг.}} = 74,87 \times 35 = 2620,36 \text{ грн.}$$

1.5.4 Розрахунок економічної ефективності та строку окупності програмного забезпечення

Ефективність - це економічна категорія, що відображає співвідношення між одержаними результатами і витраченими на їх досягнення ресурсами, причому при вимірюванні ефективності ресурси можуть бути представлені або в певному обсязі за їх первісною вартістю (застосовувані ресурси), або частиною їх вартості у формі виробничих витрат (виробничо спожиті ресурси). Якщо при цьому врахувати, що результати виробництва не лише є різноманітними, але й можуть бути представлені в різних формах: вартісній, натуральній, соціальній, то стає очевидною необхідність в ідентифікації категорії ефективності відповідно до тих аспектів діяльності підприємства, які важливо проаналізувати й оцінити.

Період окупності проекту – це час, який потрібен для того, щоб сума надходжень від реалізації проекту відшкодувала суму витрат на його впровадження. Період окупності звичайно вимірюється в роках або місяцях.

Економічна ефективність розраховується за формулою:

$$ЕФ = \Delta C - K_{\text{еф.}} * BC \quad (1.11)$$

Де: ΔC – вивільнені кошти після впровадження програмного продукту

$K_{\text{еф.}}$ – коефіцієнт економічної ефективності, що становить 0,6

BC – виробнича собівартість (з табл. №3 дорівнює 201 516,02 грн.)

$$\Delta C = \Phi ЗП_{\text{1 рік}} - EB \quad (1.12)$$

Де: $\Phi ЗП_{\text{1 рік}}$ – сума коштів, яка вивільнюється на підприємстві після впровадження програмного продукту

$$EB = A_{\text{відр.}} + MB + B_{\text{ел.}} \quad (1.13)$$

Де: $A_{\text{відр.}}$ – сума амортизаційних відрахувань (становить 2620,36 грн.)

MB – вартість матеріальних витрат (становить 2436,00 грн.)

$B_{\text{ел.}}$ – вартість електроенергії (становить 431,73 грн.)

$$EB = 2620,36 + 2436,00 + 431,73 = 5488,09 \text{ грн.}$$

Пряму економічну ефективність визначають ґрунтуючись на тому, що за експертною оцінкою фахівців підприємства впровадження програмного продукту вивільняє одного працівника, середня зарплата якого складає:

$$\Phi ЗП_{\text{м}} = 11\,600,00 \text{ грн./міс}$$

Вивільнені кошти за рік складають:

$$\Phi ЗП_{\text{1 рік}} = \Phi ЗП_{\text{м}} \times 12 \quad (1.14)$$

$$\Phi ЗП_{\text{1 рік}} = 11\,600,00 \times 12 = 139\,200,0 \text{ грн.}$$

Вивільнені кошти після впровадження програмного продукту за один рік становлять 139 200, 00 грн.

Отже у перший рік витрати на створення і експлуатацію ПЗ складуть:

$$\Delta C = 139\,200,00 - 5488,09 = 133\,711,91 \text{ грн.}$$

Строк окупності програмного продукту визначається як відношення витрат на розробку продукту до вивільнених коштів від впровадження цього продукту:

$$T_{\text{ок}} = BC : \Delta C$$

$$T_{\text{ок}} = 201\,516,02 : 133\,711,91 = 1,5 \text{ роки}$$

Отже, термін окупності програмного продукту становить 1,5 роки.

2. ПРОЕКТУВАННЯ АРМ

2.1 Моделювання та конструювання

2.1.1 Розробка моделі системи

Підхід до виявлення системних вимог заснований на використанні варіантів використання системи (Use Cases), які охоплюють як функціональні, так і нефункціональні вимоги, які специфічні для конкретного варіанту використання.

Для користувача важливо, щоб система виконувала певні дії для нього, причому користувач певним чином взаємодіє з системою, використовує її для своїх цілей. Таким чином, якщо визначити всі можливі варіанти використання системи користувачами, або іншими зовнішніми процесами, то ми отримаємо функціональні вимоги до неї.

Однак кожен конкретний користувач працює з системою по-своєму, тому для визначення дійсних варіантів використання системи необхідно досконально знати потреби всіх зацікавлених користувачів, провести аналіз отриманої інформації і систематизувати її в дійсні варіанти використання системи, тобто абстрагуватися від конкретних користувачів і виходити від бізнес-завдань.

На основі вимог визначається склад та вигляд користувацького інтерфейсу для реалізації того чи іншого прецеденту.

Для визначення вимог до програмного продукту будуємо Use-Case діаграму в уніфікованій мові моделювання Rational Rose (рис.2.1).

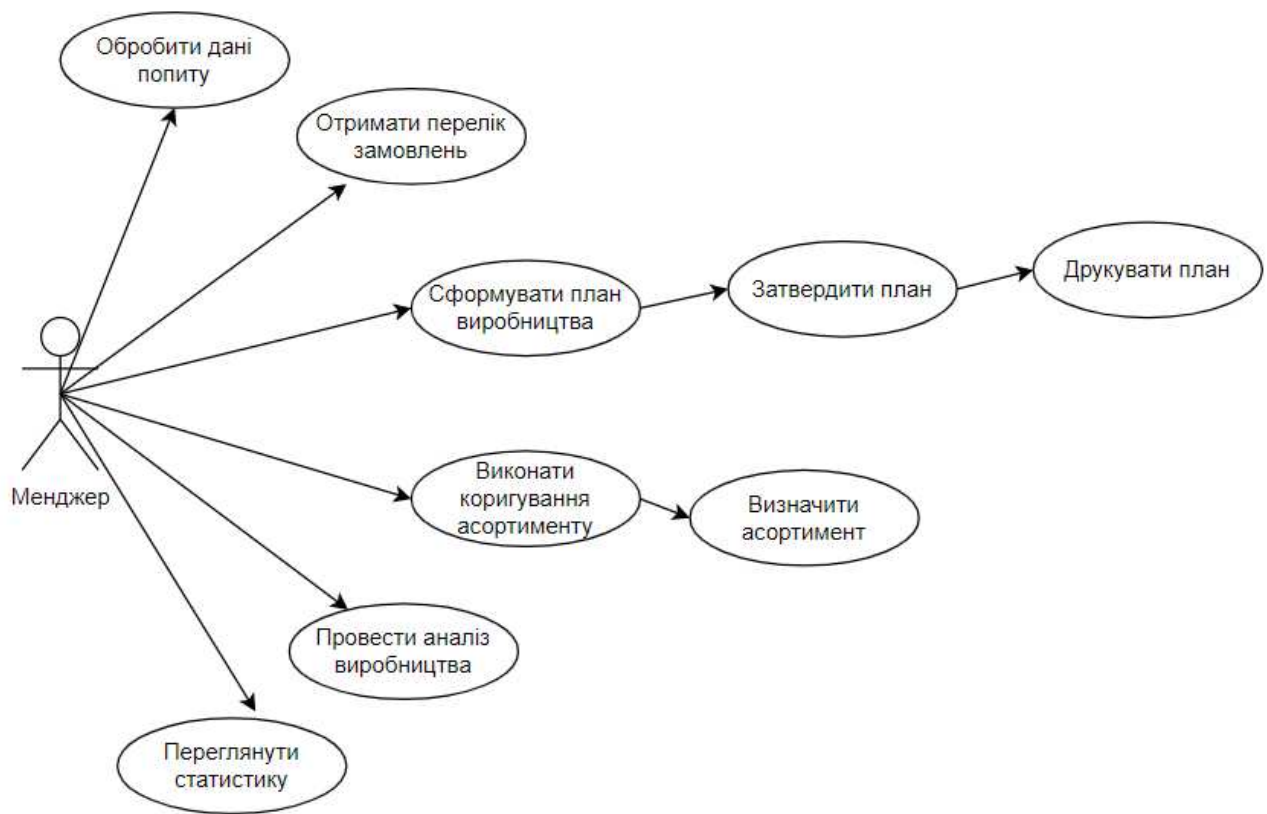


Рисунок 2.1 Діаграма прецедентів

Користувачем є менеджер. Він (актор) виконує певні функції в системі. Use-case діаграма дозволяє в подальшому нам визначити архітектуру програми та інформаційні потоки даних.

Варіанти використання менеджера включають:

- Обробку даних попиту;
- Отримання переліку замовлень;
- Формування плану виробництва;
- Затвердження плану;
- Друкування плану;
- Коригування асортименту;
- Визначення асортименту;
- Проведення аналізу виробництва;
- Перегляд статистики

Use case діаграма дозволяє узгодити із замовником функції майбутнього програмного продукту. Переглядаючи діаграму, Замовник може надати пропозиції по включенню додаткових функцій або розширенню представлених. В результаті уточнень на наступних етапах розробки програмного продукту ймовірність виникнення змін у вимогах зменшиться, що зменшує ризики розробки проекту. Use-case діаграма є основою для розробки технічного завдання.

2.1.2 Дослідження потоків даних

Дослідження потоків даних проведемо з використанням діаграм потоків даних, які є моделлю графічного представлення «потоків» даних в інформаційній системі. Діаграма потоків даних також може використовуватись для візуалізації процесів обробки даних (структурне проектування). Діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати розлого розроблювану систему.

На рисунку 2.2 приведена діаграма нульового рівня. Сутність «Менеджер» взаємодіє з процесом АРМ

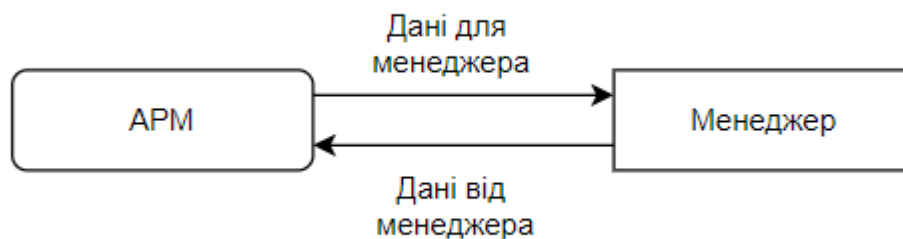


Рисунок 2.2 Діаграма потоків даних нульового рівня

Зміст інформаційних потоків ДПД нульового рівня наступний:

Дані для менеджера:

- Перелік замовлень
- Обсяги поставок
- Асортимент

- Замовники
- Статистичні дані
- Виробничі потужності
- План виробництва

Дані від менеджера:

- Статистичні дані попиту
- Пропозиції до плану
- Оперативна інформація про поставки
- План виробництва
- Асортимент продукції
- Вартість продукції
- Затверджений план
- Коригування асортименту
- Виробничі потужності

Побудуємо діаграму першого рівня, яка дає уявлення про функціонування системи. Процес першого рівня містить в собі наступні процеси: «Формувати/коригувати довідники», «Обробити дані відвантаження», «Обробити дані поставок ресурсів».

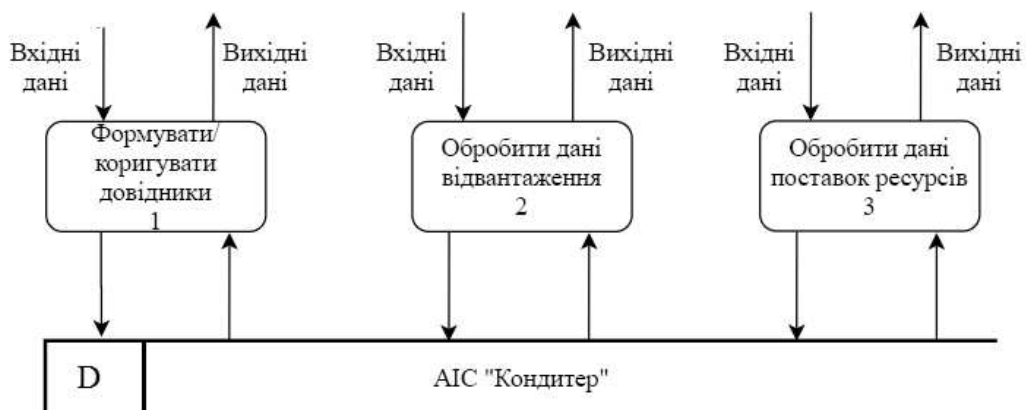


Рисунок 2.3 Діаграма потоків даних 1-го рівня

Таблиця 2.1 Словник даних ДПД 1-го рівня

ДПД 0-го рівня	ДПД 1-го рівня
Інформація від менеджера	Вхідна інформація
Інформація для менеджера	Вихідна інформація

Процес «Обробки даних відвантаження» представлено на рисунку 2.4 дає основне уявлення про функціонування процесу. Процес складається з під процесів: «Обробки даних накладної відвантаження», «Формування статистики відвантаження», «Формування плану виробництва».

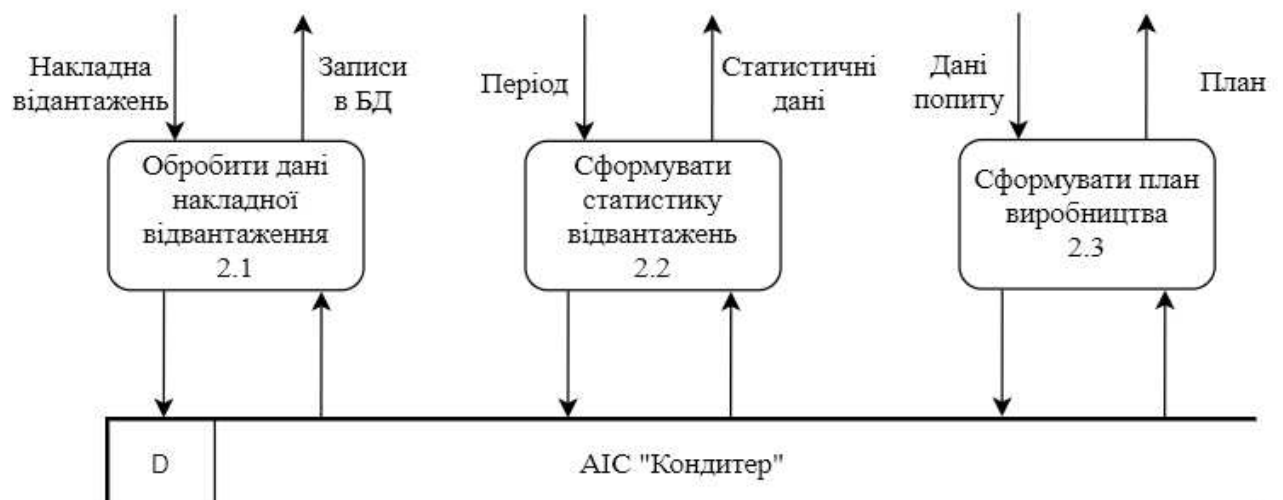


Рисунок 2.4 ДПД 2-го рівня задачі 2

Таблиця 2.2 Словник даних ДПД 2-го рівня задачі 2

ДПД 1-го рівня	ДПД 2-го рівня	Атрибути
1	2	3
Вхідні дані	Накладна відвантаження	Номер накладної Дата Замовник Назва товару Кількість
	Період постачання	Дата Продукція Замовник
	Дані попиту	Період Код товару Назва товару Обсяг проданого Запланований обсяг Різниця
Вихідні дані	Накладна-заголовок	Номер накладної Дата Замовник
	Накладна-зміст	Номер накладної Назва товару Кількість
	Статистичні дані	Назва товару Обсяг Період
	План на продукцію	Період Код товару Назва товару Запланований обсяг

Процес аналізу фактичного продажу представлено на рисунку 2.5 дає основне уявлення про планування календарного випуску продукції. Цей процес складається з процесу «Аналізу даних фактичного продажу» та «Формування календарного плану випуску продукції».

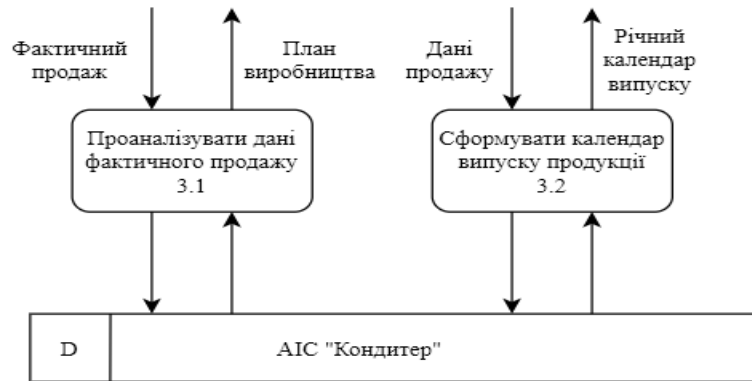


Рисунок 2.5 ДПД 2-го рівня задачі 3

Таблиця 2.3 Словник даних ДПД 3-го рівня задачі 3

ДПД 2-го рівня	ДПД 3-го рівня	Атрибути
1	2	3
Вхідні дані	Фактичний продаж	Код товару Назва товару Період
	Дані з продажу	Назва товару Кількість проданого за місяць
Вихідні дані	План	Код товару Назва товару Запланований випуск на рік
	Календар випуску продукції	Назва товару Заплановий випуск на місяць

Побудуємо ДПД 2-го рівня для процесів ведення довідників (рис. 2.6)



Рисунок 2.6 ДПД 2-го рівня задачі «Формування та ведення довідників»

Таблиця 2.4 Словник даних ДПД 2-го рівня задачі «Формування та ведення довідників»

ДПД 1-го рівня	ДПД 2-го рівня	Атрибути
1	2	3
Вхідні дані	Список товару	Номер товару Наявність Тип товару
	Список ресурсів	Номер ресурсу Дані поставок Дата поставки
Вихідні дані	Довідник товару	Код товару Група Назва товару Одиниця виміру
	Довідник ресурсів	Код ресурсу Назва ресурсу Одиниця виміру Ціна

2.1.3 Концептуальна модель

Побудова концептуальної моделі – процес побудови інформаційної моделі предметної області, яка відображує її з урахуванням вимог висунутих до цього процесу.

Результати дослідження інформаційних потоків предметної області надають можливість створити концептуальну модель даних, яка є першим кроком при створенні будь – якої БД. При створенні концептуальної моделі ми використовуємо визначені на попередньому етапі сутності та їх атрибути. В концептуальній моделі відобразимо зв'язки між сутностями, визначимо ключові атрибути та типи зв'язків.

Сутність – інформаційний об'єкт, який зберігається в БД.

Атрибут – представляє собою властивість сутності.

Зв'язки сутностей – залежність між атрибутами цих сутностей. Зв'язок може бути між двома або більше сутностями.

Після дослідження даної предметної області було виявлено такі сутності:

- 1) Продукція
- 2) Постачальники ресурсів
- 3) Ресурси
- 4) Замовники
- 5) Накладна відвантаження продукції
- 6) Накладна надходження ресурсів
- 7) Норми витрат ресурсів на одиницю продукції

Сутності пов'язані між собою зв'язками один – до – багатьох, багато – до – одного та один – до – одного (рис. 2.7).

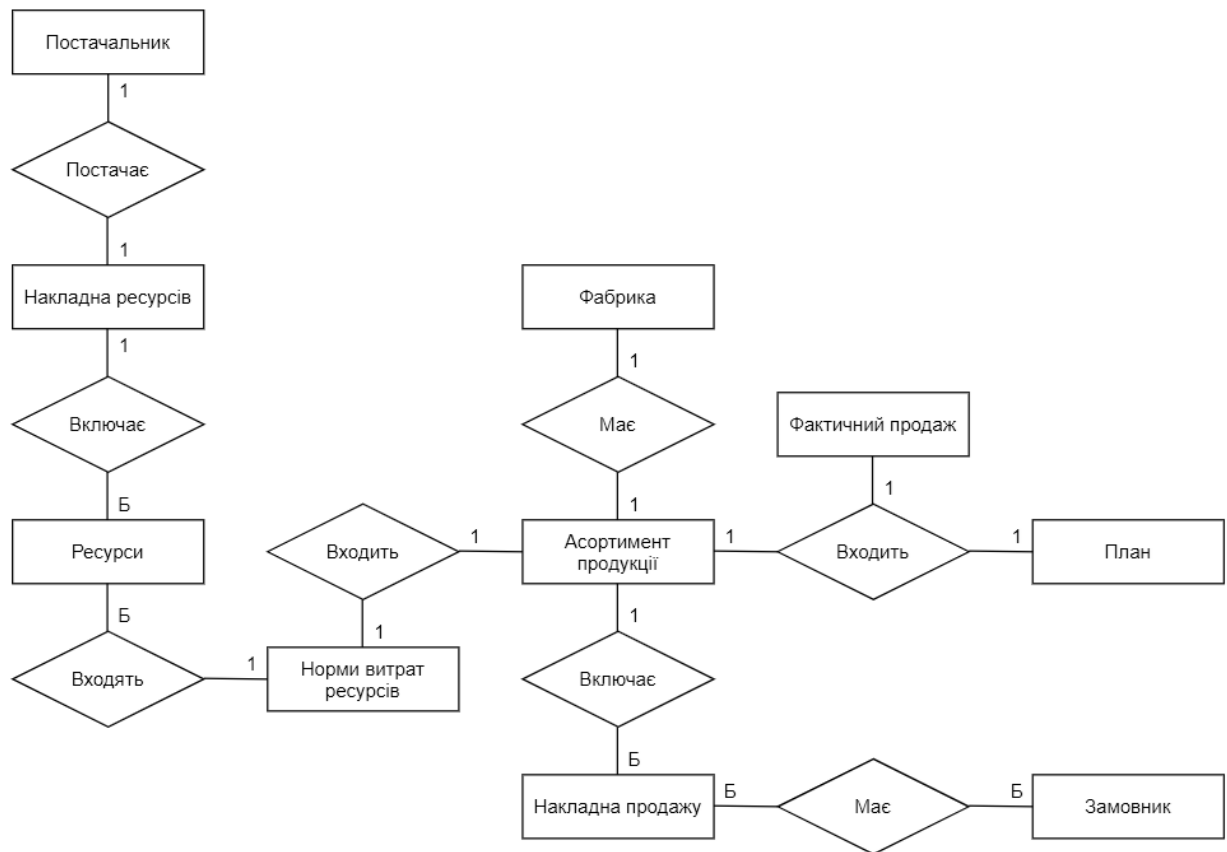


Рисунок 2.7 Концептуальна модель даних

2.1.4 Розробка діаграми станів програми

Діаграма станів дозволяє описати можливі послідовності станів і переходів, які в сукупності характеризують поведінку елемента моделі на протязі його життєвого циклу. Вона представляє динамічну поведінку сутностей, на основі специфікації їх реакції на сприйняття деяких конкретних подій.

Хоча діаграми станів часто використовуються для опису поведінку окремих екземплярів класів (об'єктів), але вони також можуть бути застосовані для специфікації функціональності інших компонентів моделей, таких як варіанти використання, актори, підсистеми, операції і методи.

Діаграма станів по суті є графом спеціального виду, який представляє деякий автомат. Поняття автомата в контексті UML має доволі специфічну семантику, яка базується на теорії автоматів. Вершинами цього графа є стани і деякі інші типи елементів автомата (псевдостани), які зображуються відповідними

графічними символами. Дуги графа служать для позначення переходів із стану в стан. Діаграми станів можуть бути вкладені одна в одну, утворюючи вкладені діаграми більш детального представлення окремих елементів моделі.

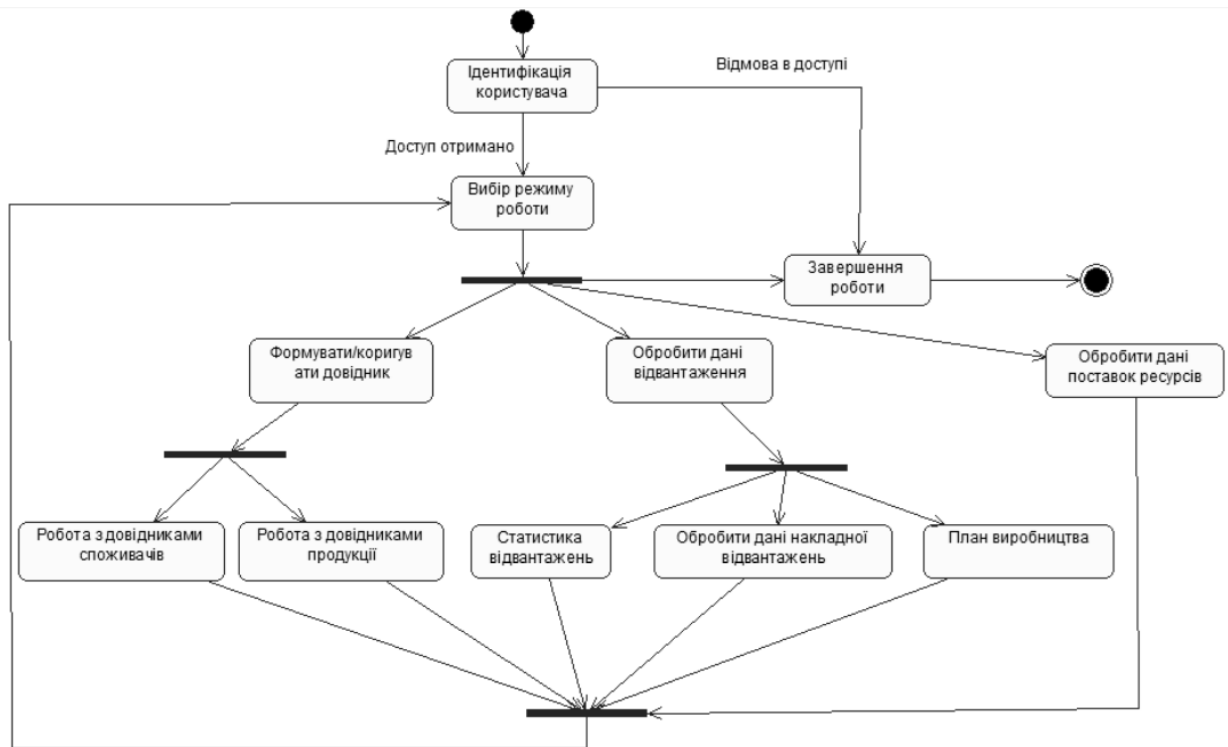


Рисунок 2.8 Діаграма станів програми

2.2 Технічний проект

2.2.1 Логічна модель даних

Процес розробки інформаційно логічної моделі вимагає знання предметної області й розуміння її логічних взаємозв'язків, тому процес виділення ІО буде ефективніше якщо даний етап виконується разом з користувачем і розроблювачем додатка.

Дані про об'єкти предметної області перебувають у деяких документах, які є основними носіями інформації представлені друкованим або рукописним способом на бланку встановленої форми і юридичну чинність, що має.

Логічна модель базується на концептуальній. Вона є більш детальним представленням даних, одна сутність може передаватися більш ніж одною таблицею, атрибути сутностей стають полями таблиць, відношення між таблицями стають більш точними, виділяються первинні ключі.

Логічна модель даних наведена в таблицях 2.5-2.13

Таблиця 2.5 Логічна модель таблиці «Асортимент» (Assortment)

Поле	Тип даних	Ключ
Код продукту	Числовий	+
Тип продукту	Числовий	
Назва продукту	Текстовий	
Одиниця виміру	Текстовий	

Таблиця 2.6 Логічна модель таблиці «Тип продукції» (ProdGroup)

Поле	Тип даних	Ключ
Код типу	Числовий	+
Назва типу	Текстовий	

Таблиця 2.7 Логічна модель таблиці «Ресурси» (Resource)

Поле	Тип даних	Ключ
Код ресурсу	Числовий	+
Назва ресурсу	Текстовий	
Одиниця виміру	Текстовий	
Ціна	Числовий	

Таблиця 2.8 Логічна модель таблиці «Нормо-витрати» (CostRate)

Поле	Тип даних	Ключ
Код продукту	Числовий	
Код ресурсу	Числовий	
Кількість	Числовий	

Таблиця 2.9 Логічна модель таблиці «Замовники» (Users)

Поле	Тип даних	Ключ
Код замовника	Текстовий	+
Назва/ім'я замовника	Текстовий	
Адреса	Текстовий	
Телефон	Текстовий	

Таблиця 2.10 Логічна модель таблиці «Накладна постачання_заголовок» (NaklPostZag)

Поле	Тип даних	Ключ
Код накладної	Текстовий	+
Дата відвантаження	Дата	
Код замовника	Текстовий	

Таблиця 2.11 Логічна модель таблиці «Накладна постачання_зміст» (NaklPostZm)

Поле	Тип даних	Ключ
Код накладної	Текстовий	
Код продукту	Числовий	
Кількість продукції	Числовий	
Ціна відвантаження	Числовий	

Таблиця 2.12 Логічна модель таблиці «План» (Plan)

Поле	Тип даних	Ключ
Код продукту	Числовий	
План січня	Числовий	
План лютого	Числовий	
План березня	Числовий	

Продовження таблиці 2.12

План квітня	Числовий	
План травня	Числовий	
План червня	Числовий	
План липня	Числовий	
План серпня	Числовий	
План вересня	Числовий	
План жовтня	Числовий	
План листопада	Числовий	
План грудня	Числовий	
Рік	Числовий	

Таблиця 2.13 Логічна модель таблиці «Фактичний продаж» (fact)

Поле	Тип даних	Ключ
Код продукту	Числовий	
Продаж за січень	Числовий	
Продаж за лютий	Числовий	
Продаж за березень	Числовий	
Продаж за квітень	Числовий	
Продаж за травень	Числовий	
Продаж за червень	Числовий	
Продаж за липень	Числовий	
Продаж за серпень	Числовий	
Продаж за вересень	Числовий	
Продаж за жовтень	Числовий	
Продаж за листопад	Числовий	
Продаж за грудень	Числовий	
Рік	Числовий	

Результат відображення побудованої ІЛМ у логічну структуру реляційної БД можна виразити графічно у вигляді даних. Кожний ІО (таблиця БД) зображується прямокутником, куди вписуються назви ІО, а потім після горизонтальної риси всі його реквізити (поля таблиці). Ім'я ключового поля для наочності доцільно виділити, і привести серед реквізитів першим. Між зв'язаними полями на схемі проводяться стрілки, спрямовані від ключового поля в головному ІО до цього ж поля в залежному ІО.

На рисунку 2.9 наведено схему даних бази даних, яка була побудована базуючись на представлений логічній моделі.

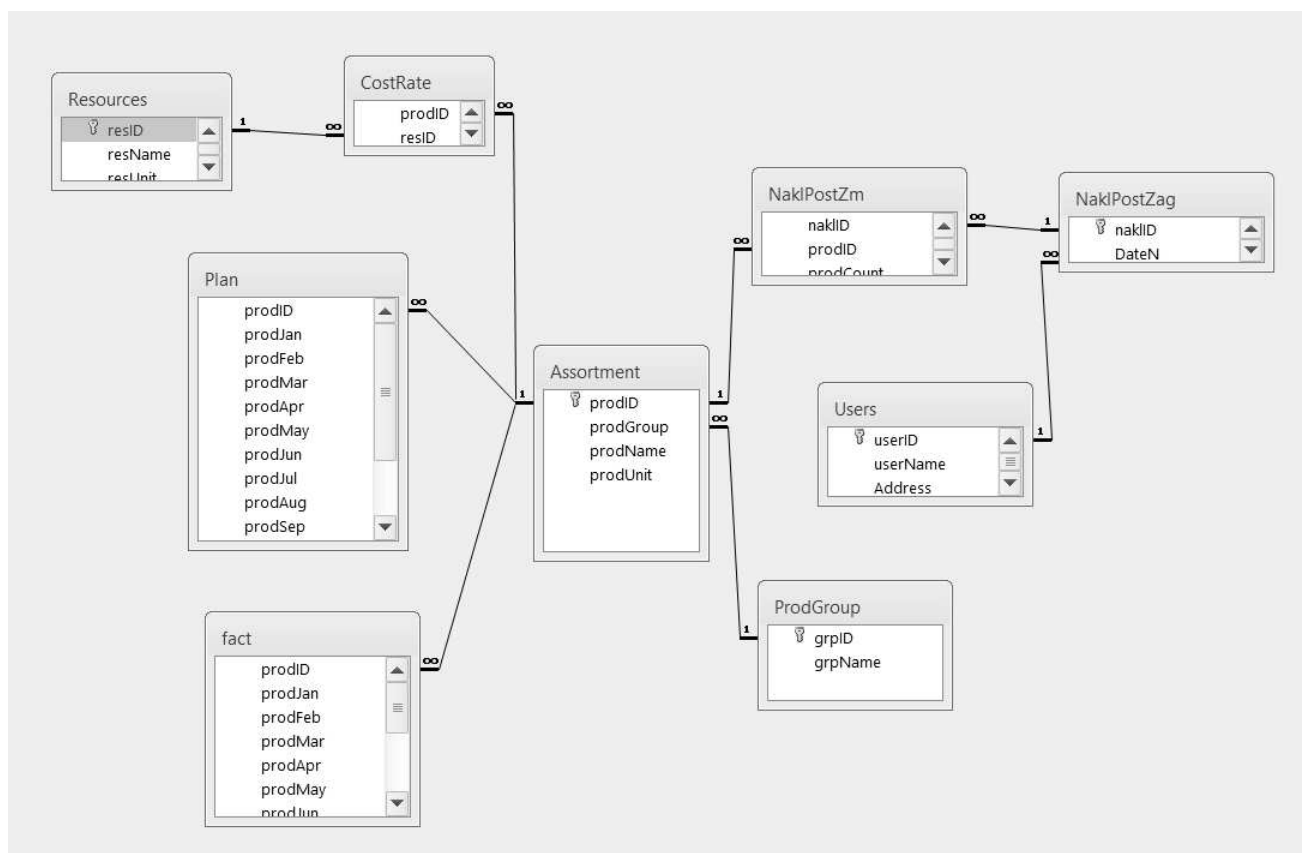


Рисунок 2.9 Схема бази даних

2.2.2 Розробка архітектури програми

Архітектура ПЗ являє собою набір структур або уявлень, що мають різні рівні абстракції і показують різні аспекти (структуру класів ПО, структуру розгортання, тобто прив'язки компонентів ПО до фізичних машин, можливі

сценарії взаємодій компонентів та ін.), що об'єднуються прив'язкою всіх представлених даних до структурних елементів ПЗ

Архітектура програмного забезпечення подається у вигляді окремих програмних модулів. Перелік модулів визначаємо на основі створеної раніше Use-Case діаграми. Функції, які виконують актори, становлять певні модулі програми. Роботою програми керує головний модуль, який дозволяє користувачу обрати режим роботи. Після вибору одного з режимів роботи активізується форма.

Перелік модулів включає:

- Модуль перевірки прав доступу;
- Модуль вибору режимів роботи;
- Модуль роботи з довідниками;
- Модуль роботи з звітами
- Модуль роботи з запитами;
- Модуль налаштувань;
- Модуль завершення роботи.

2.2.3 Специфікація модулів

В архітектурі програми передбачено модульну структуру. Кожен модуль реалізує певну операцію.

Модуль «mainUnit», призначений для отримання доступу до програми, після якого користувач може відразу обрати необхідний режим роботи з програмою.

Вхідні дані: логін та пароль користувача.

Дія: перевірка введеного логіну та паролю.

Вихідні дані: головна форма програми якщо дані були введені коректно або повідомлення про помилку при вводі даних авторизації.

Модуль “Асортимент” (assUnit) викликається головним модулем.

Призначення: перегляд/редагування/доповнення інформації про певний товар.

Вхідні дані: вибір позиції на видалення, коригування, або реквізити нового запису (нової позиції асортименту).

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Асортимент».

Модуль “Ресурси” (resUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації про певний ресурс.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Ресурси».

Даний модуль дозволяє нам переглянути/редагувати інформацію про ресурси.

Модуль “Тип продукції” (groupUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації про тип продукту.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Тип продукції».

Даний модуль дозволяє нам переглянути/редагувати інформацію про тип ресурсів.

Модуль “План” (planUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення плану.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «План».

Даний модуль дозволяє нам переглянути/редагувати/додати інформацію про план виготовлення продукції за певний період.

Модуль “Фактичний продаж” (factUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення фактичного продажу.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Фактичний продаж».

За допомогою даного модуля можна переглянути/редагувати/додати інформацію про фактичний продаж продукції за певний період.

Модуль “Норми-витрат” (costForm) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації щодо норми-витрат.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Норми-витрат».

Даний модуль дозволяє нам переглянути/редагувати інформацію про кількість використання ресурсів на виготовлення одиниці продукції.

Модуль “Замовники” (userForm) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації замовників продукції.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Замовники».

Даний модуль дозволяє нам переглянути/редагувати/додати інформацію про замовників.

Модуль “Накладна” (invoiceForm) викликається головним модулем.

Вхідні дані: перегляд інформації накладних за обраний період.

Дія: перегляд записів.

Вихідна інформація: формування плану.

Даний модуль дозволяє нам проаналізувати обсяги реалізації продукції за певний місяць та сформувати план виробництва.

Модуль “Звіти” (reportForm) викликається головним модулем.

Вхідні дані: інформація таблиць.

Дія: створення звіту.

Вихідна інформація: створений звіт.

Даний модуль дозволяє нам створити та переглянути звіти по певним таблицям, які також можна роздрукувати або зберегти в окремому файлі за бажанням користувача.

Модуль «settingsUnit» дозволяє користувачу налаштувати програмне забезпечення.

Вхідні дані: відсутні.

Дія: зміна інтерфейсу, мови програми або налаштування певних функцій програми.

Вихідні дані: відсутні.

2.2.4 Алгоритм роботи програми

Алгоритм — це послідовність точних команд виконавцеві алгоритму спрямованих на розв'язання певної задачі. Алгоритм поданий графічним способом у вигляді блок-схеми.

Блок-схема алгоритму — це графічне зображення алгоритму у вигляді спеціальних блоків із необхідними словесними поясненнями. Кожний етап алгоритму на блок-схемі подається у вигляді геометричної фігури, яка має певну форму залежно від характеру дії.

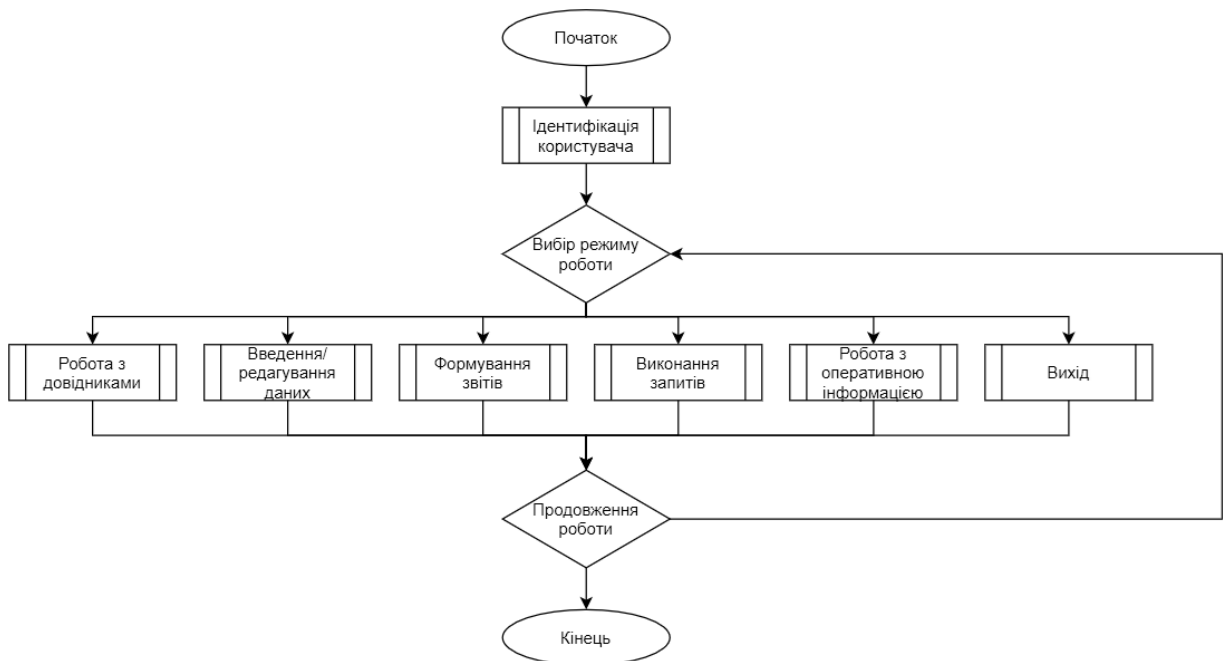


Рисунок 2.10 Алгоритм роботи програми

2.3 Робочий проект

2.3.1 Вибір засобів розробки програми

В якості середовища розробки обрано Delphi 7.0 від компанії Borland. Delphi 7.0 є одним з середовищ, яке дозволяє швидко створювати прикладні програми з графічним інтерфейсом для платформи Windows. До його складу входить біля 200 самих різних компонентів, крім того, Delphi 7 дозволяє створювати, встановлювати та використовувати в подальшому власні компоненти.

Замість окремого інструмента, що дозволяє працювати з візуальними компонентами, в Delphi 7.0 інтегрована так звана «Палітра компонентів», розділена вкладками на кілька функціональних груп.

Середовище містить бібліотеку з більше ніж 100 повторно використовуваних візуальних компонентів, які перетягуються мишею на форму і відразу стають елементами управління прототипу програми. Окрім знайомих елементів управління Windows (кнопки, полоси прокрутки, поля редагування, прості та комбіновані списки та ін.) бібліотека містить нові компоненти підтримки діалогів, обслуговування баз даних та багато інших.

Після розміщення компоненту на формі, можна редагувати їх властивості та встановлювати обробку подій об'єкту за допомогою кодування. Проект створюється поступово, на тлі змін у властивостях, які виконує розробник, подіях і функціях використовуваних елементів.

IDE Delphi 7.0 забезпечує високу швидкодію при компіляції і складанні застосувань для сучасних операційних систем платформи Windows. Система може відобразити кількість часу затраченого на створення ПЗ. В результаті програми добре оптимізовані за швидкістю виконання і витратам пам'яті. Хоча налагоджувальний режим низького рівня є в даному середовищі, до налагодження також доводиться звикати. Дизайнер форм, інспектор об'єктів і інші засоби залишаються доступними під час роботи програми, тому вносити зміни можна в процесі налагодження.

Borland Delphi 7.0 підтримує зв'язок з різними базами даних 3-х видів: dBASE и Paradox; Sybase, Oracle, InterBase та Informix; Excel, Access, FoxPro та Btrieve. Механізм BDE (Borland Database Engine) надає обслуговуванню зв'язків з базами даних простоту і прозорість. Database Explorer дозволяє зображувати зв'язки та об'єкти баз даних графічно.

Delphi 7 дозволяє проводити налагодження та тестування програми завдяки засобам керування проектами, двосторонній інтеграції додатку синхронізації між засобами візуального і текстового редагування, а також вбудованому налагоджувачу (з асемблерним вікном прокрутки, покроковим виконанням, точками зупинки, трасуванням тощо).

В якості СУБД було обрано MS Access. MS Access на даний час є однією з популярних СУБД. Серед причин такої популярності слід зазначити:

- універсальність і продуманість інтерфейсу, який дозволяє працювати з БД користувачам з різними навичками. Зокрема, реалізована система управління об'єктами бази даних, що дозволяє гнучко і оперативно переходити з режиму конструювання в режим експлуатації;
- глибоко розвинені можливості інтеграції з іншими програмними продуктами, що входять до складу MS Office (наприклад Excel), а також з будь-якими програмними продуктами, що підтримують технологію OLE.
- багатий набір візуальних засобів розробки.

Microsoft Access — система управління базами даних, програма, що входить до складу пакету офісних програм Microsoft Office. Має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних. Завдяки вбудованій мові VBA, в самому Access можна писати підпрограми, що працюють з старими версіями Microsoft Office Access. Дане середовище працює з сімейством ОС Windows та має доступний і зрозумілий інтерфейс для користувачів будь-якого рівня.

2.3.2 Фізична модель бази даних

Побудова фізичної моделі бази даних повністю заснована на принципах організації даних, реалізованих в логічній моделі бази даних.

Фізична модель визначає розміщення даних у зовнішній пам'яті. Вона ще називається внутрішньою моделлю системи і форма її представлення залежить від обраної СКБД. Так, ідентифікатори (імена) таблиць і полів мають задовольняти вимогам СКБД, типи даних, розміри полів, обмеження теж мають бути приведені у відповідність до прийнятих у даній СКБД. В даному випадку використовується MS Access.

Таблиця 2.14 Фізична модель таблиці «Assortment»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код продукту	prodID	Числовий	Ціле
Тип продукту	prodGroup	Числовий	Довге ціле
Назва продукту	prodName	Текстовий	50
Одиниця виміру	prodUnit	Текстовий	15

Таблиця 2.15 Фізична модель таблиці «ProdGroup»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код типу	grpID	Числовий	Ціле
Назва типу	grpName	Текстовий	20

Таблиця 2.16 Фізична модель таблиці «Resources»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код ресурсу	resID	Числовий	Ціле

Продовження таблиці 2.16

Назва ресурсу	resName	Текстовий	50
Одиниця виміру	resUnit	Текстовий	15
Ціна	resCost	Числовий	Дійсне з подвійною точністю

Таблиця 2.17 Фізична модель таблиці «CostRate»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код продукту	prodID	Числовий	Ціле
Код ресурсу	resID	Числовий	Ціле
Кількість	costRate	Числовий	Дійсне з подвійною точністю

Таблиця 2.18 Фізична модель таблиці «Users»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код замовника	userID	Текстовий	20
Назва/ім'я замовника	userName	Текстовий	50
Адреса	Address	Текстовий	100
Телефон	phoneNumber	Текстовий	20

Таблиця 2.19 Фізична модель таблиці «NaklPostZag»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код накладної	naklID	Текстовий	20
Дата відвантаження	dateN	Дата	Короткий формат дати

Продовження таблиці 2.19

Код замовника	userID	Текстовий	20
---------------	--------	-----------	----

Таблиця 2.20 Фізична модель таблиці «Plan»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код накладної	naklID	Текстовий	20
Код продукту	prodID	Числовий	Ціле
Кількість продукції	prodCount	Числовий	Ціле
Ціна відвантаження	unloadPrice	Числовий	Дійсне з подвійною точністю

Таблиця 2.21 Фізична модель таблиці «Plan»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код продукту	prodID	Числовий	Ціле
План січня	prodJan	Числовий	Ціле
План лютого	prodFeb	Числовий	Ціле
План березня	prodMar	Числовий	Ціле
План квітня	prodApr	Числовий	Ціле
План травня	prodMay	Числовий	Ціле
План червня	prodJun	Числовий	Ціле
План липня	prodJul	Числовий	Ціле
План серпня	prodAug	Числовий	Ціле
План вересня	prodSep	Числовий	Ціле
План жовтня	prodOct	Числовий	Ціле

Продовження таблиці 2.21

План листопада	prodNov	Числовий	Ціле
План грудня	prodDec	Числовий	Ціле
Річний план	prodYear	Числовий	Ціле

Таблиця 2.22 Фізична модель таблиці «fact»

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код продукту	prodID	Числовий	Ціле
Продаж за січень	prodJan	Числовий	Ціле
Продаж за лютий	prodFeb	Числовий	Ціле
Продаж за березень	prodMar	Числовий	Ціле
Продаж за квітень	prodApr	Числовий	Ціле
Продаж за травень	prodMay	Числовий	Ціле
Продаж за червень	prodJun	Числовий	Ціле
Продаж за липень	prodJul	Числовий	Ціле
Продаж за серпень	prodAug	Числовий	Ціле
Продаж за вересень	prodSep	Числовий	Ціле
Продаж за жовтень	prodOct	Числовий	Ціле
Продаж за листопад	prodNov	Числовий	Ціле
Продаж за грудень	prodDec	Числовий	Ціле
Рік	prodYear	Числовий	Ціле

2.3.3 Кодування

Кодування — процес написання коду обраною або обраними мовами програмування з метою реалізації певного алгоритму або алгоритмів. В більш точному сенсі — процес перетворення моделей проектування в моделі реалізації обраною мовою програмування. В Delphi програмний код використовується для обробки подій компонентів форми з якими взаємодіє користувач.

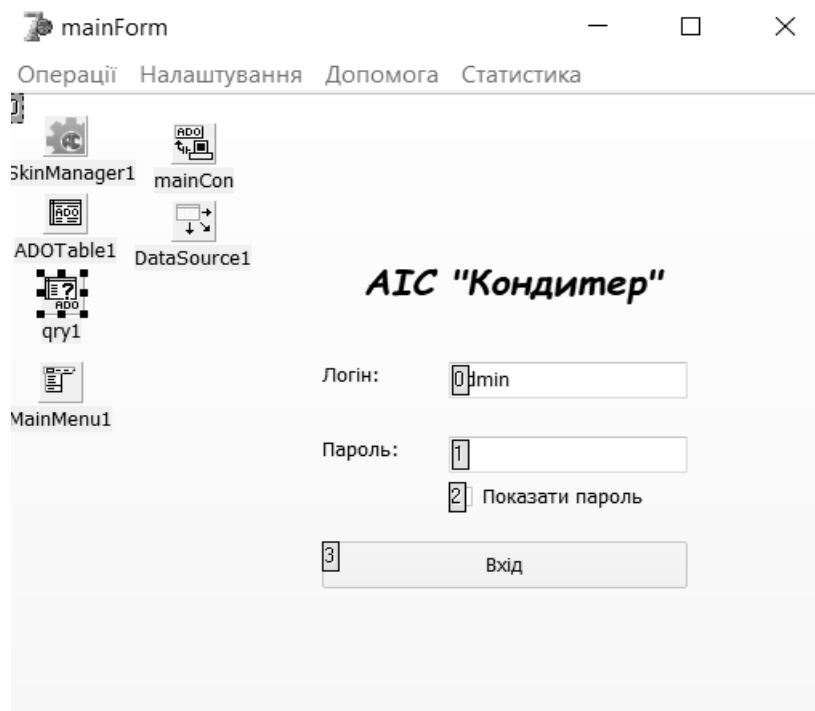


Рисунок 2.11 Головна форма програми в режимі проектування

Приклад обробки натиснення на кнопку «Вхід» на мові програмування Delphi.

```
procedure TmainForm.btnLogClick(Sender: TObject);
var Ini: TIniFile;
begin
  if (edtLog.Text <> 'admin') or (edtPass.Text <> '') then
    ShowMessage('Логін або пароль введений невірно!')
  else begin
    c1.Enabled:=true;
    spnl1.Visible:=false;
    spnl2.Visible:=true;
    Ini:= TIniFile.Create('\config.ini');
    try
      //Edit1.Enabled:= Ini.ReadBool('TMenuItem', 'EditEnabled', true);
      Calc1.Enabled:= Ini.ReadBool('TMenuItem', 'CalcEnabled', true);
      Set1.Enabled:= Ini.ReadBool('TMenuItem', 'SetEnabled', true);
```

```
//Account1.Enabled:= Ini.ReadBool('TMenuItem', 'AccEnabled', true);  
Help1.Enabled:= Ini.ReadBool('TMenuItem', 'HelpEnabled', true);  
finally  
  Ini.Free;  
end;  
end;  
end;
```

2.3.4 Тестування та налагодження програми

Тестування програмного забезпечення — це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки.

Тестування програмного забезпечення - процес перевірки відповідності заявлених до продукту вимог і реально реалізованої функціональності, здійснюваний шляхом спостереження за його роботою в штучно створених ситуаціях і на обмеженому наборі тестів, обраних певним чином.

Об'єктом випробувань є програмне забезпечення задачі „Розробка Автоматизованої інформаційної системи «Кондитер»”.

Тестування буде проводитись за такими критеріями:

- відповідність вимогам, які були поставлені даному програмному забезпеченню;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з програмним забезпеченням та операційними системами;
- відповідність задачам замовника.

Оскільки число можливих тестів навіть для нескладних програмних компонент практично нескінченне, то стратегія тестування полягає в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів. Як результат програмне забезпечення (ПЗ) тестується стандартним виконанням програми з метою виявлення багів (помилки або інших дефектів).

Під час тестування визначались вхідні дані, їх перетворення програмою та форма запису до БД вихідних даних. На основі визначалася працездатність модулів.

Проведемо тест модуля авторизації. В таблиці 2.3.10 представлені виділені класи еквівалентності.

Таблиця 2.23 Виділені класи еквівалентності

№ теста	Вхідні дані	Вірні класи	Невірні класи
1	Логін	Admin	root
		Admin	user
2	Пароль	Admin	1234567890
		Admin	password

Результати тестування на основі поданих вище класів еквівалентності можна побачити в таблиці 2.24

Таблиця 2.24 Тести вірних та невірних класів еквівалентності

№ теста	Класи	Вхідні дані	Вихідні дані
1	Вірний клас	Admin/Admin	Надання доступу до програми
2	Невірний клас	Admin/root	Неправильний логін/пароль
3		Admin/1234567890	Неправильний логін/пароль
4		Admin/password	Неправильний логін/пароль

Аналогічні класи еквівалентності використано для інших модулів програми.

В результаті виконання тестування та налагодження програми було отримано програмний продукт, який забезпечує виконання всіх запланованих функцій та локалізує помилки користувача.

3. РОЗРОБКА АРМ

В результаті розробки програмного забезпечення було реалізовано всі закладені в технічному завданні функції програми. Програмний продукт умовно можна розподілити на три напрямки:

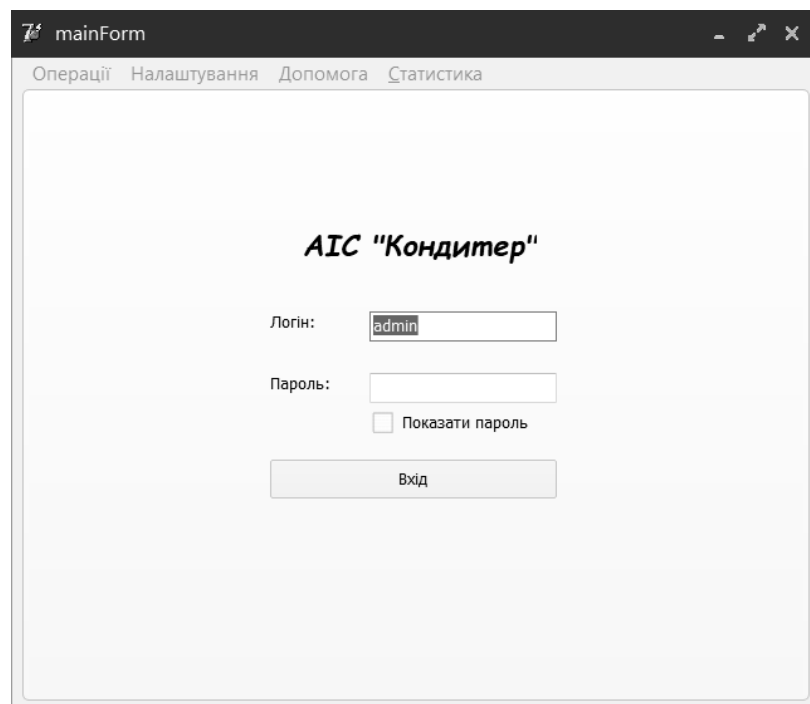
Формування та ведення довідників задачі;

Оброблення оперативних даних перевірки об'єктів;

Формування статистичних звітів.

Користувачу надано інструментарій автоматизації ручних операцій, наприклад, можливість швидкого пошуку даних за критеріями. При формуванні масивів даних для завантаження в БД проводиться контроль повноти даних контроль унікальності значень ключового поля, надано систему підказок при формуванні даних, що значно полегшує процес завантаження даних.

Інтерфейс програми на рисунках 3.1 – 3.8.



The screenshot shows a web application window titled 'mainForm'. It features a menu bar with 'Операції', 'Налаштування', 'Допомога', and 'Статистика'. The main content area displays the title 'АІС "Кондитер"' in a bold, italicized font. Below the title are two input fields: 'Логін:' with the value 'admin' and 'Пароль:'. A checkbox labeled 'Показати пароль' is positioned below the password field. A 'Вхід' button is located at the bottom of the form.

Рисунок 3.1 Форма ідентифікації користувача

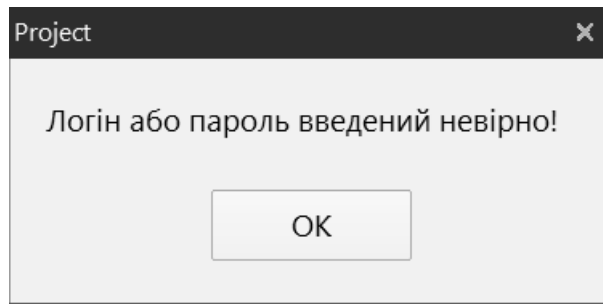


Рисунок 3.2 Діалогове вікно помилки доступу

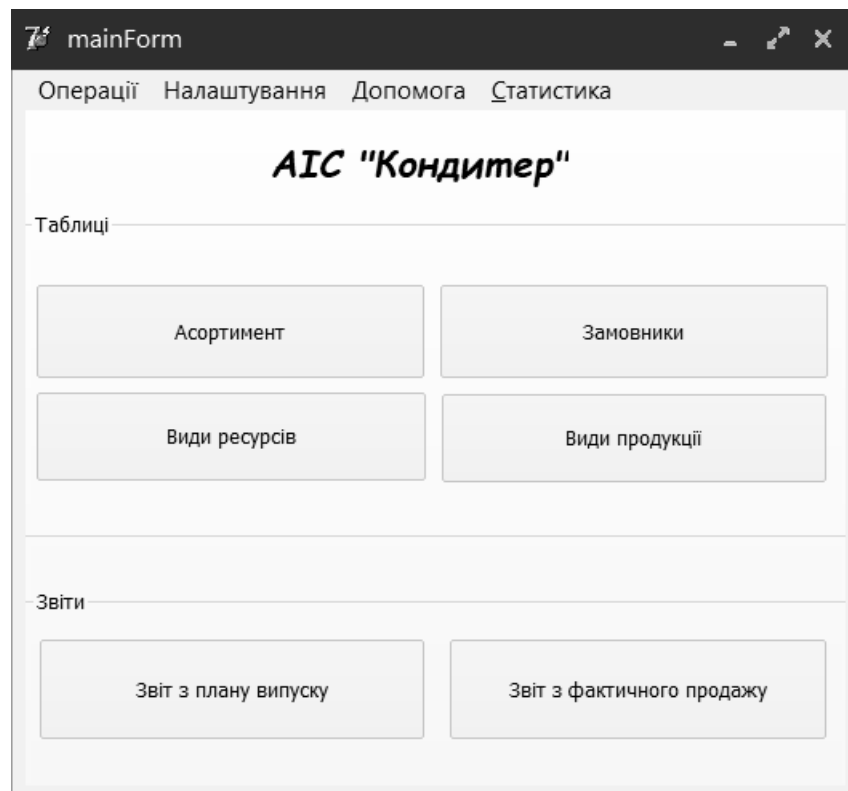


Рисунок 3.3 Головна форма в режимі роботи

assForm

Пошук

Знайти: Очистити ☒ По назві ☐ По типу

prodID	prodGr...	prodName	prodUnit
1	1	test	Kг
2	1	Жорж	Kг
3	1	Марія	Kг
4	12	Грім	Kг
6	3	Мрія	Kг
7	2	Північна ніч	Kг
8	6	ОС з лісовим горіхом	Kг
9	6	ОС гіркий	Kг
10	10	Grimasiki	Kг
11	1	Зоряний вечір	Kг
12	2	Ренклюд	Kг
13	11	Помадка вишнева	Kг
15	2	Забодайка молоко	Kг

Редагування

Код продукції: 7 Тип продукції: Цукерки Назва продукції: Північна ніч Оддиниця виміру: Кг

⏪ ⏴ ⏵ ⏩ + - ▲ ✓ ✕ ↺

Рисунок 3.4 Форма роботи з довідником «Асортимент»

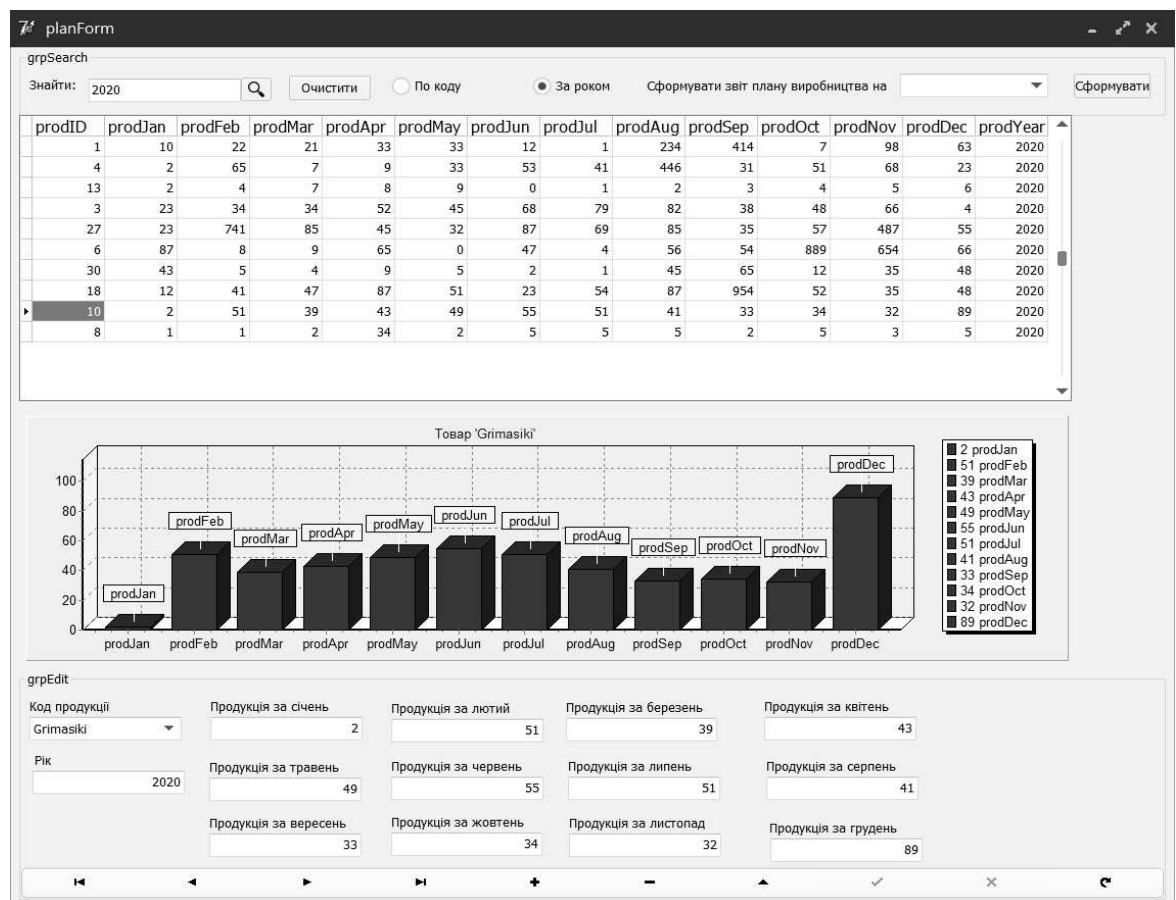


Рисунок 3.5 Результат пошуку продукції за роком

Print Preview

План випуску продукції 2020 року

Назва	Січень	Лютий	Березень	Квітень	Травень	Червень	Липень	Серпень	Вересень	Жовтень	Листопад	Грудень
test	10	22	21	33	33	12	1	234	414	7	98	63
Грім	2	65	7	9	33	53	41	446	31	51	68	23
Помадка вишнева	2	4	7	8	9	0	1	2	3	4	5	6
Марія	23	34	34	52	45	68	79	82	38	48	66	4
La prima nuts	23	741	85	45	32	87	69	85	35	57	487	55
Мрія	87	8	9	65	0	47	4	56	54	889	654	66
Nicole	43	5	4	9	5	2	1	45	65	12	35	48
Ластівка	12	41	47	87	51	23	54	87	954	52	35	48
Grimasiki	2	51	39	43	49	55	51	41	33	34	32	89
ОС з лісовим горіхом	1	1	2	34	2	5	5	5	2	5	3	5

Page 1 of 1

Рисунок 3.6 Звіт плану випуску продукції

Print Preview

Кількість випуску продукції за 2020 рік щоквартально

Назва	Перший квартал	Другий квартал	Третій квартал	Четвертий квартал	Річна сума
test	53	78	649	168	948
Грім	74	95	518	142	829
Помадка вишнева	13	17	6	15	51
Марія	91	165	199	118	573
La prima nuts	849	164	189	599	1801
Мрія	104	112	114	1609	1939
Nicole	52	16	111	95	274
Ластівка	100	161	1095	135	1491
Grimasiki	92	147	125	155	519
Грім	4	41	12	13	70

Рисунок 3.7 Квартальний та річний звіт з плану випуску продукції

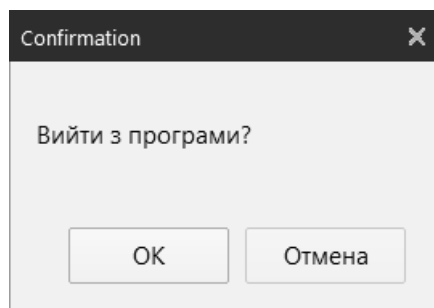


Рисунок 3.8 Діалогове вікно підтвердження виходу

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Згідно ст. 1 Закону України «Про охорону праці» під охороною праці розуміється система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів і засобів спрямованих на збереження здоров'я і працездатності людини в процесі праці. На основі вимог КЗОТ конкретні заходи по створенню здорових та безпечних умов праці, попередженню нещасних випадків та професійних захворювань регламентуються спеціальними правилами та нормами. Згідно з Законом України «Про охорону праці» нормативні акти можуть бути як єдиними для всіх галузей народного господарства, так і спеціальними для даної галузі. Державні міжгалузеві й галузеві нормативні акти про охорону праці - це правила, стандарти, норми, положення, інструкції та інші документи, яким надана сила правових норм, обов'язкових для виконання.

Даний розділ дипломного проекту присвячений розгляду питань охорони праці при роботі за комп'ютером.

4.2 Основні законодавчі і нормативні документи, які регламентують Охорону праці України

До основних документів, що регламентують охорону праці в Україні, відносяться:

- Конституція України;
- Кодекс законів про працю;
- закони «Про охорону праці», «Про охорону здоров'я», «Про пожежну безпеку», «Про використання ядерної енергії та радіоактивний захист», «Про забезпечення санітарного та епідемічного благополуччя населення».

Кодекс законів про працю України затверджено Законом Української РСР від 10 грудня 1971 р. і введено в дію з 1 червня 1972 р. До нього неодноразово вносилися зміни і доповнення. Правове регулювання охорони праці в ньому не обмежується главою XI “Охорона праці”. Норми щодо охорони праці містяться в

багатьох статтях інших глав: “Трудовий договір”, “Робочий час”, “Час відпочинку”, “Праця жінок”, “Праця молоді”, “Професійні спілки”, “Нагляд і контроль за дотриманням законодавства про працю”.

Закон України “Про забезпечення санітарного та епідемічного благополуччя населення” встановлює необхідність гігієнічної регламентації небезпечних і шкідливих факторів фізичної, хімічної та біологічної природи, присутніх в середовищі життєдіяльності людини, та їхньої державної реєстрації, вимоги до проектування, будівництва, розробки, виготовлення і використання нових засобів виробництва та технологій, гігієнічні вимоги до атмосферного повітря в населених пунктах, повітря у виробничих та інших приміщеннях, вимоги щодо забезпечення радіаційної безпеки тощо.

Закон України “Про охорону здоров’я” регулює суспільні відносини в цій галузі з метою забезпечення гармонічного розвитку фізичних і духовних сил, високої працездатності та довголітнього активного життя громадян, усунення чинників, які шкідливо впливають на їхнє здоров’я, попередження і зниження захворюваності, інвалідності та смертності, поліпшення спадкоємності; передбачає встановлення єдиних санітарно-гігієнічних вимог до організації виробничих та інших процесів, пов’язаних з діяльністю людей, а також до якості машин, устаткування, будинків та таких об’єктів, що можуть шкідливо впливати на здоров’я людей; вимагає проведення обов’язкових медичних оглядів осіб певних категорій, в тому числі працівників, зайнятих на роботах із шкідливими та небезпечними умовами праці; закладає правові основи медико-соціальної експертизи втрати працездатності.

Закон України “Про пожежну безпеку” визначає загальні правові, економічні та соціальні основи забезпечення пожежної безпеки на території України, регулює відносини державних органів, юридичних і фізичних осіб у цій галузі незалежно від виду їх діяльності та форм власності. Забезпечення пожежної безпеки є складовою частиною виробничої та іншої діяльності посадових осіб, працівників підприємств, установ, організацій та підприємців, що повинно бути відображено у трудових договорах (контрактах) та статутах підприємств, установ

та організацій. Забезпечення пожежної безпеки підприємств, установ та організацій покладається на їх керівників і уповноважених ними осіб, якщо інше не передбачено відповідним договором.

До спеціальних законодавчих актів в галузі охорони праці належать:

- Закон України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності» (23.09.1999 р. № 1105);
- Порядок розслідування та обліку нещасних випадків, професійних захворювань і аварій на виробництві (25.08.2004 р. №1112);
- Порядок видачі дозволів Державним комітетом з нагляду за охороною праці та його територіальними органами (15.10.2003 р. № 1631);
- Порядок проведення атестації робочих місць за умовами праці (01.08.1992 р. №442).

4.3 Вимоги до охорони праці користувачів ПК

Однією із характерних особливостей сучасного розвитку суспільства є зростання сфер діяльності людини, в яких використовуються інформаційні технології. Широке розповсюдження отримали персональні комп'ютери. Однак їх використання загострило проблеми збереження власного та суспільного здоров'я, вимагає удосконалення існуючих та розробки нових підходів до організації робочих місць, проведення профілактичних заходів для запобігання розвитку негативних наслідків впливу ПК на здоров'я користувачів.

Заходи з охорони праці користувачів ПК необхідно розглядати в трьох основних аспектах: соціальному, психологічному та медичному.

У соціальному плані розв'язання цих проблем пов'язане з оптимізацією умов життя, праці, відпочинку, харчування, побуту, розвитком культури, транспорту.

Значне місце у профілактиці розладів здоров'я належить психології праці. Тому заходи, пов'язані з формуванням раціональних виробничих колективів, у яких відсутня психологічна несумісність, сприяють зменшенню нервово-психічного перенапруження, підвищенню працездатності та ефективності праці.

Особливої значущості у користувачів відео дисплейних терміналів набуває психоемоційний стрес, який більшою або меншою мірою проявляється у кожного з них.

Значна роль у профілактиці захворювань користувачів ПК відводиться медицині. Існує перелік профілактичних заходів для користувачів ПК, що включає як складові первинної профілактики здоров'я (професійний відбір), так і вторинної, яка направлена на зниження ймовірності розвитку перевтоми та перенапруження. Ці комплексні заходи спрямовані на відновлення функціонального стану зорового та опорно-рухового апарату.

У Правилах викладені гігієнічні й ергономічні вимоги до організації робочих приміщень та робочих місць, параметрів робочого середовища, дотримання яких дає змогу запобігти порушенням стану здоров'я користувачів ПК. Правилами забороняється затверджувати нормативну і технічну документацію на нові ВДТ, постачання їх у виробництво, продаж і використання у виробничих умовах, а також закупівля їх і ввезення в Україну без:

- гігієнічної оцінки їх безпеки для здоров'я людини;
- узгодження нормативної і технічної документації на ці види продукції з органами Держсанепідемслужби і Держкомохоронпраці України.

Відповідальність за виконання цих Правил покладається на посадових осіб, фізичних осіб, які займаються підприємницькою діяльністю і здійснюють розробку, виробництво, закупівлю, реалізацію й застосування ЕОМ та ПЕОМ в адміністративних і промислових приміщеннях.

Державний санітарний нагляд за дотриманням цих Правил державними органами, підприємствами, установами, організаціями незалежно від форми власності, а також фізичними особами, які займаються підприємницькою діяльністю, покладається на органи і установи санітарно-епідеміологічного профілю Міністерства охорони здоров'я.

України, відповідні установи, організації, частини й підрозділи Міністерства оборони України, Міністерства внутрішніх справ України, Державного комітету у справах охорони державного кордону України, Національної гвардії України,

Служби безпеки України (ст. 31 Закону України "Про забезпечення санітарного та епідеміологічного благополуччя населення").

4.4 Загальні вимоги до виробничих приміщень ЕОМ

Облаштованість робочих місць, обладнаних відеотерміналами, повинне забезпечувати:

- належні умови освітлення приміщення і місця робітника, відсутність відблисків;
- оптимальні параметри мікроклімату (температура, відносна вологість повітря, швидкість руху, рівень іонізації повітря);
- належні ергономічні характеристики основних елементів робочого місця.

Також враховувати наступні небезпечні і шкідливі фактори:

- наявність шуму і вібрації;
- м'яке рентгенівське випромінювання;
- електромагнітне випромінювання;
- ультрафіолетове й інфрачервоне випромінювання;
- електростатичне поле між екраном і оператором;
- наявність пилу, озону, окислів азоту й іонізації.

4.5 Санітарно-гігієнічні вимоги виробничого середовища ЕОМ

Санітарно-гігієнічні вимоги до виробничого середовища ЕОМ визначають:

- мікроклімат;
- степінь освітленості;
- рівень шуму і вібрації;
- іонізуюче та неіонізуюче випромінювання.

Мікроклімат

Мікроклімат на робочих місцях з ПК нормується ДЕСТ 12.1.005-88 і повинен мати:

- температуру оточуючого повітря – 21–24°C;
- відносну вологість – 40–60 %;
- швидкість руху повітря – 0,1–0,2 м/с.

Освітлення

- робочі місця з ПК повинні мати природне і штучне освітлення;
- вікна приміщень з ПК повинні мати орієнтацію на північ або на північний схід, з КПО 1,5%. Дозволяється експлуатація ЕОМ у приміщеннях без природного освітлення за узгодженням з органами Держнаглядохоронпраці та санітарно-епідеміологічними установами;
- рівень освітленості на робочому місці в зоні розміщення документів має бути 300–500 лк;
- штучне освітлення виконується, як правило, люмінесцентними лампами. Допускається для місцевого освітлення використовувати лампи розжарювання;
- місцеве освітлення не повинно створювати блисків на екрані ПК, а його освітленість повинна не перевищувати 300 лк.

Рівні шуму

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях і на робочих місцях застосовуються шумопоглинальні засоби, вибір яких влаштовується спеціальними інженерно-акустичними розрахунками.

Як засоби звукопоглинання повинні використовуватися непальні чи важко запалювані спеціальні перфоровані плити, панелі, мінеральна вата з максимальним коефіцієнтом звукопоглинання в межах частот 31,5 – 8000 Гц, чи інші матеріали аналогічного призначення, дозволені органами державного нагляду для оформлення приміщень. Крім того, необхідно застосовувати підвісні стелі з аналогічними властивостями.

Рівні вібрації

Рівні вібрації не повинні перевищувати допустимих норм, визначених ДсанПіН 3.3.2-007-98.

Рівні іонізуючого випромінювання

Рівні іонізуючого випромінювання повинні відповідати НРБУ–97. Потужність експозиційної дози рентгенівського випромінювання на відстані 5 см від екрану ПК повинна бути не більше 0,1 мбер/год (100 мкР/год).

Рівні неіонізуючого випромінювання

Рівні неіонізуючого випромінювання повинні відповідати вимогам ДЕСТ 12.1.006, СН №3206-85 «Гранично допустимі рівні магнітного поля частотою 50 Гц, СН №4557-88 «Санітарні норми ультрафіолетового випромінювання у виробничих приміщеннях».

4.6 Перелік шкідливих та небезпечних факторів, які супроводжують роботу програміста

Тонізуюче (рентгенівське) випромінювання. Дослідження показують, що монітор не є небезпечним для користувача ПЕОМ" оскільки інтенсивність такого випромінювання нижча за гранично допустимі норми (ГДН). Відповідно до "Норм радіаційної безпеки України" (НРБУ-97), гранично допустима потужність експозиційної дози рентгенівського випромінювання на відстані 5 см від екрану відео-термінала становить $7,74 \times 10^{-12}$ Кл/кг, що відповідає еквівалентній дозі 0,1 мбер/год. (100 мкР/год.).

4.7 Ергономічні вимоги до робочого місця програміста

Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого столу має регулюватися в межах 680-800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600-1400мм, глибина – 800-1000мм).

Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600мм, завширшки не менше ніж 500мм, завглибшки (на рівні колін) не менше ніж 450мм,

на рівні простягнутої ноги не менше ніж 650мм. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися. Шаг регулювання елементів стільця має становити: для лінійних розмірів – 15-20мм, для кутових – 2-5 градусів. Зусилля регулювання має не перевищувати 20Н. Висота поверхні сидіння має регулюватися в межах 400-500мм, а ширина і глибина становити не менше ніж 400мм. Кут нахилу сидіння – до 15 градусів вперед і до 5 градусів назад. Висота спинки стільця має становити (300+/-20) мм, ширина – не менше ніж 380 мм, радіус кривизни горизонтальної площини – 400мм.

Кут нахилу спинки має регулюватися в межах 1-30 градусів від вертикального положення. Відстань від спинки до переднього краю сидіння має регулюватися в межах 260-400мм. Для зниження статичного напруження м'язів верхніх кінцівок слід використовувати стаціонарні або змінні підлокітники завдовжки не менше ніж 250мм, завширшки 50-70мм, що регулюються за висотою над сидінням у межах 230-260мм і відстанню між підлокітниками в межах 350-500мм. Поверхня сидіння і спинки стільця має бути напівм'якою з нековзним, повітронепроникним покриттям, що легко чиститься і не електризується. Робоче місце має бути обладнане підставкою для ніг завширшки не менше ніж 300мм, завглибшки не менше ніж 400мм, що регулюється за висотою в межах до 150мм і за кутом нахилу опорної поверхні підставки до 20 градусів. Підставка повинна мати рифлену поверхню і бортик по передньому краю заввишки 10мм.

Робочі місця слід розташовувати відносно світових прорізів так, щоб природне світло падало переважно з лівого боку. Монітор має розташовуватися на оптимальній відстані від очей користувача, що становить 600-700мм, але не ближче ніж за 600мм з урахуванням розміру літерно-цифрових знаків і символів. Розташування екрана монітору має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30 градусів до нормальної лінії погляду працівника. Клавіатуру слід розташовувати на поверхні столу на відстані

100-300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовільному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5-15 градусів. Висота середнього рядка клавіш має не перевищувати 30мм. Поверхня клавіатури має бути матовою з коефіцієнтом відбиття 0,4. Розташування пристрою введення – виведення інформації має забезпечувати добру видимість монітору, зручність ручного керування в зоні досяжності моторного поля і за висотою – 900-1300мм, за шириною 400-500мм. Під матричні принтери потрібно підкладати вібраційні килимки для гасіння вібрації та шуму.

Робоче місце з персональним комп'ютером слід обладнати пюпітром для документів, що легко переміщуються.

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідно застосування при екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат.

4.8 Висновки

В даному розділі було проаналізовано основні проблеми охорони праці, що можуть виникнути підчас роботи працівника. Було виділено основні вимоги до приміщення, мікроклімату в приміщенні, освітлення та основних ергономічних характеристик.

ВИСНОВОК

У процесі створення АРМ були вивчені матеріали з предметної області, розглянуті принципи роботи інтерфейсу ADO для роботи з СУБД. У результаті було створено програму, БД, інструкцію користувача. Програмний продукт було протестовано на контрольному прикладу.

Основна задача по створенню МІС в якій забезпечено автоматизацію операцій по обробці статистичних даних на кондитерській фабриці виконана в повному обсязі. Використання візуальних засобів розробки ПЗ дозволило створити програмний продукт, який відповідає вимогам технічного завдання. Середовище програмування має інструментальні засоби візуального програмування та надає можливості використання готових компонентів і створення складних програм, що можуть працювати в багато користувальницькому режимі. Застосування методів тестування дозволило забезпечити якість програмного продукту.

Для створення діаграм та схем був використаний CASE-засіб Rational Rose.

Інтерфейс програмного забезпечення є зрозумілим для користувачів, які не мають досконалих навичок володіння персональним комп'ютером.

Використання діаграм, при проектуванні ПЗ, дозволило наочно подати, як структуру БД так і архітектуру програми. Розробка дозволяє: автоматизувати операції обробки статистичних даних результатів продажу товару кондитерської фабрики; отримувати довідкову інформацію про асортимент, ресурси та замовників продукції; отримувати аналітичну інформацію в вигляді звітів за певний період часу.

Результатом розробки стало АРМ керівника, що містить в собі базу даних кондитерської фабрики, створену в MS Access за допомогою SQL-засобів, та програмний продукт, що був створений в IDE Delphi.

В дипломному проекті було вирішено задачу з створення АРМ, що допоможе оброблювати статистичні дані поставок продукції кондитерської фабрики та формувати план виробництва на наступний рік.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika/>
2. http://vtei.com.ua/images/VN/25_10.pdf
3. http://4ua.co.ua/management/sa2ad69a4d53a88521206c36_0.html
4. <https://studfile.net/preview/5561956/page:2/>
5. <http://www.student-works.com.ua/kursovi/bjd/493.html>
6. <http://www.fa.ru/org/chair/gmu/Documents/News/2013/07/Bern,31%20of%20May.pdf>
7. https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення
8. http://opcb.kpi.ua/wp-content/uploads/2014/05/Лекц_ООП_ІТС.pdf
9. <https://docplayer.net/66793764-Osnovi-ohoroni-praci-k-n-tkachuk-m-o-halimovskiy-v-v-zaparniy-d-v-zsrkalov-r-v-sabarno-o-i-polukarov-v-s-koz-yakov-l-o.html>
10. <http://opcb.kpi.ua/wp-content/uploads/2014/09/Конспект-ООП-2013-5УЕУКУМУС.pdf>
11. <https://studfile.net/preview/9924583/page:8/>
12. <https://library.if.ua/book/9/973.html>
13. “Економіка підприємства” О.С. Іванілов. Київ «Центр учбової літератури» 2009
14. <https://epdf.tips/-9eaa324fbc25d1713a1a9659177da66685732.html>
15. https://studopedia.su/16_177901_logichna-model-bazi-danih.html
16. http://eprints.library.odeku.edu.ua/id/eprint/7509/1/Tereschenko_Modeli_tehn_proekt_IS_KL_2017.pdf

Додаток А – Інструкція користувача

Користувач отримує два файли: файл програми (konditProg.exe) та файл бази даних (kondDB.mdb). Для правильної роботи програми вище згадані файли слід розмістити в одному каталозі, в іншому випадку, програма не буде функціонувати.

При завантаженні програми активізується форма доступу. Меню вгорі заблоковано до тих пір, поки не буде введено правильний логін та пароль.

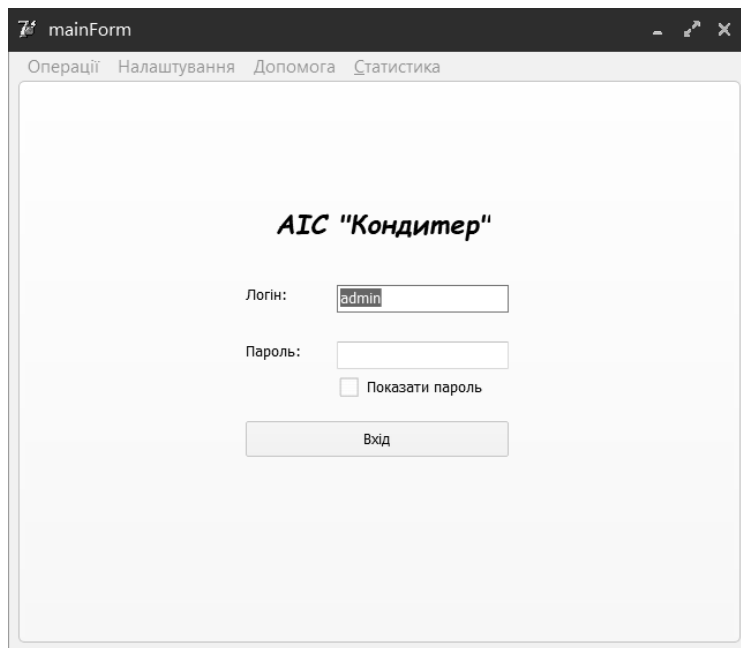


Рисунок А.1 Вікно доступу до системи

Якщо пароль або логін був введений неправильно, видається наступне повідомлення.

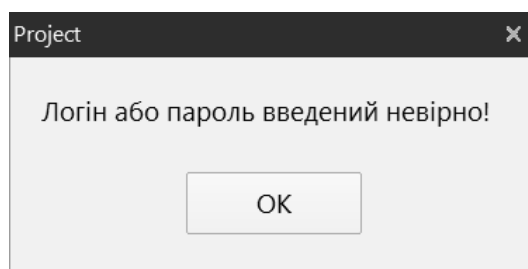


Рисунок А.2 Діалогове вікно помилки доступу

В іншому випадку доступ до системи отримано, активізується форма з режимами роботи.

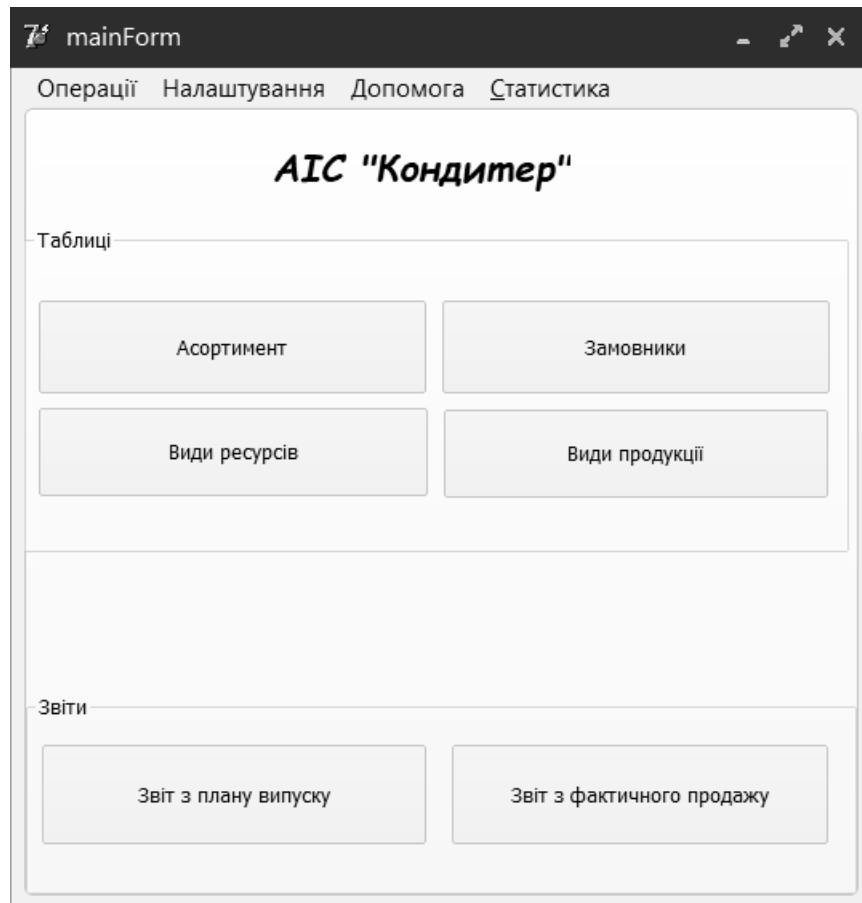
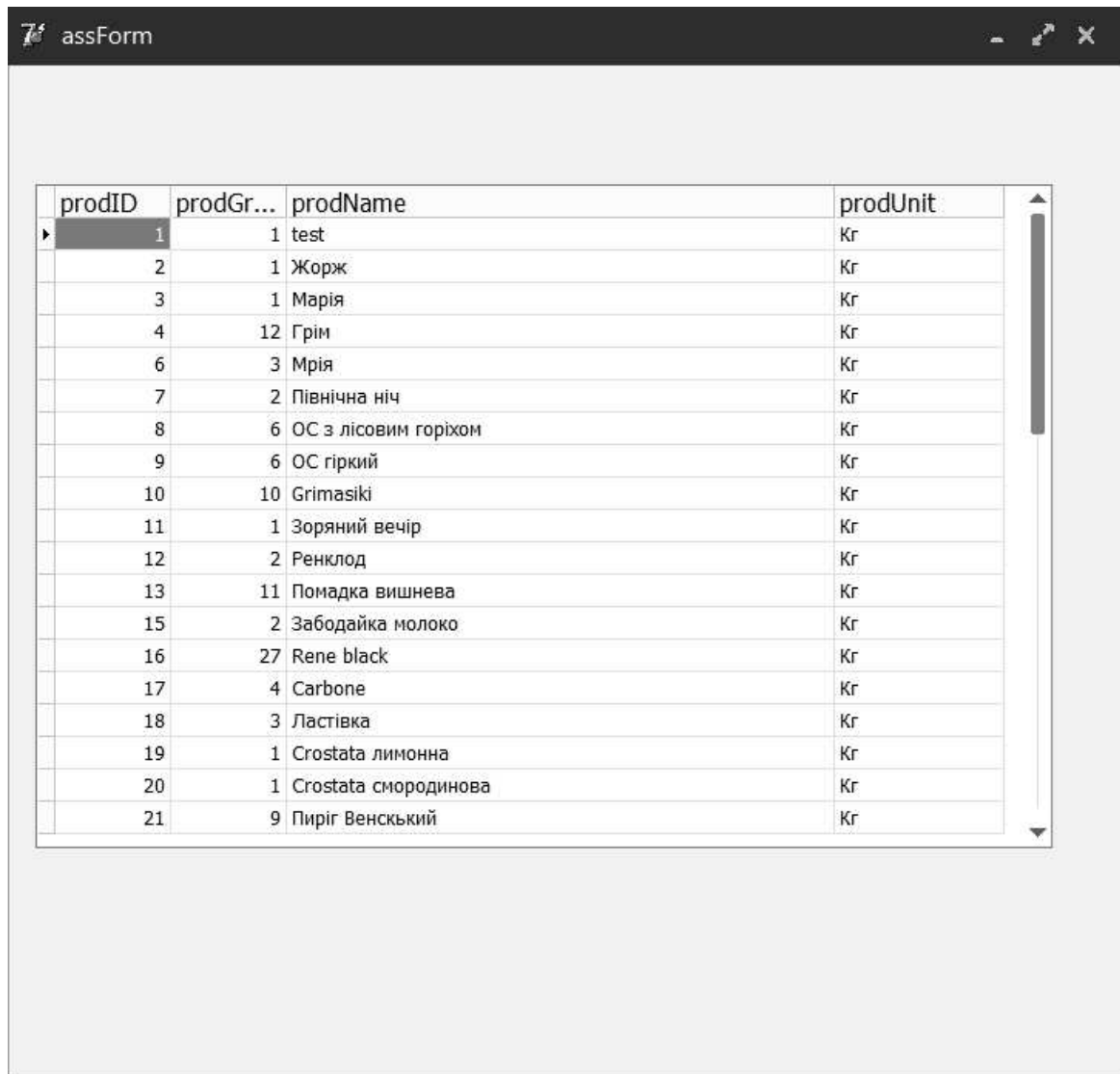


Рисунок А.3 Головна форма в режимі роботи

Вибір режиму відбувається шляхом вибору пункту меню вгорі або натисненням на кнопки на формі.



prodID	prodGr...	prodName	prodUnit
1	1	test	Кг
2	1	Жорж	Кг
3	1	Марія	Кг
4	12	Грім	Кг
6	3	Мрія	Кг
7	2	Північна ніч	Кг
8	6	ОС з лісовим горіхом	Кг
9	6	ОС гіркий	Кг
10	10	Grimasiki	Кг
11	1	Зоряний вечір	Кг
12	2	Ренклюд	Кг
13	11	Помадка вишнева	Кг
15	2	Забодайка молоко	Кг
16	27	Rene black	Кг
17	4	Carbone	Кг
18	3	Ластівка	Кг
19	1	Crostata лимонна	Кг
20	1	Crostata смородинова	Кг
21	9	Пиріг Венський	Кг

Рисунок А.4 Форма «Асортимент»

Після вибору таблиці можна бачити її зміст, при натисненні правою кнопкою миші в області таблиці буде відкрито меню, в якому можна обрати, дії над таблицею – це редагування та пошук. Також можна використовувати комбінації клавіш.

Режим редагування	Ctrl+E
Режим пошуку	Ctrl+F

Рисунок А.5 Режим обрання дій

При виборі режиму пошуку, зверху з'явиться панель пошуку, в ній можна обрати вид пошуку та ввести в поле «Знайти» параметр пошуку. Пошук є динамічним, тобто при введенні даних результати будуть відразу показані на

формі. Кнопка «Очистити» очищає поле пошуку, для виходу з панелі слід натиснути комбінацію клавіш «Ctrl + F» або праву кнопку мишки (ПКМ) – Режим пошуку, після чого панель буде приховано.

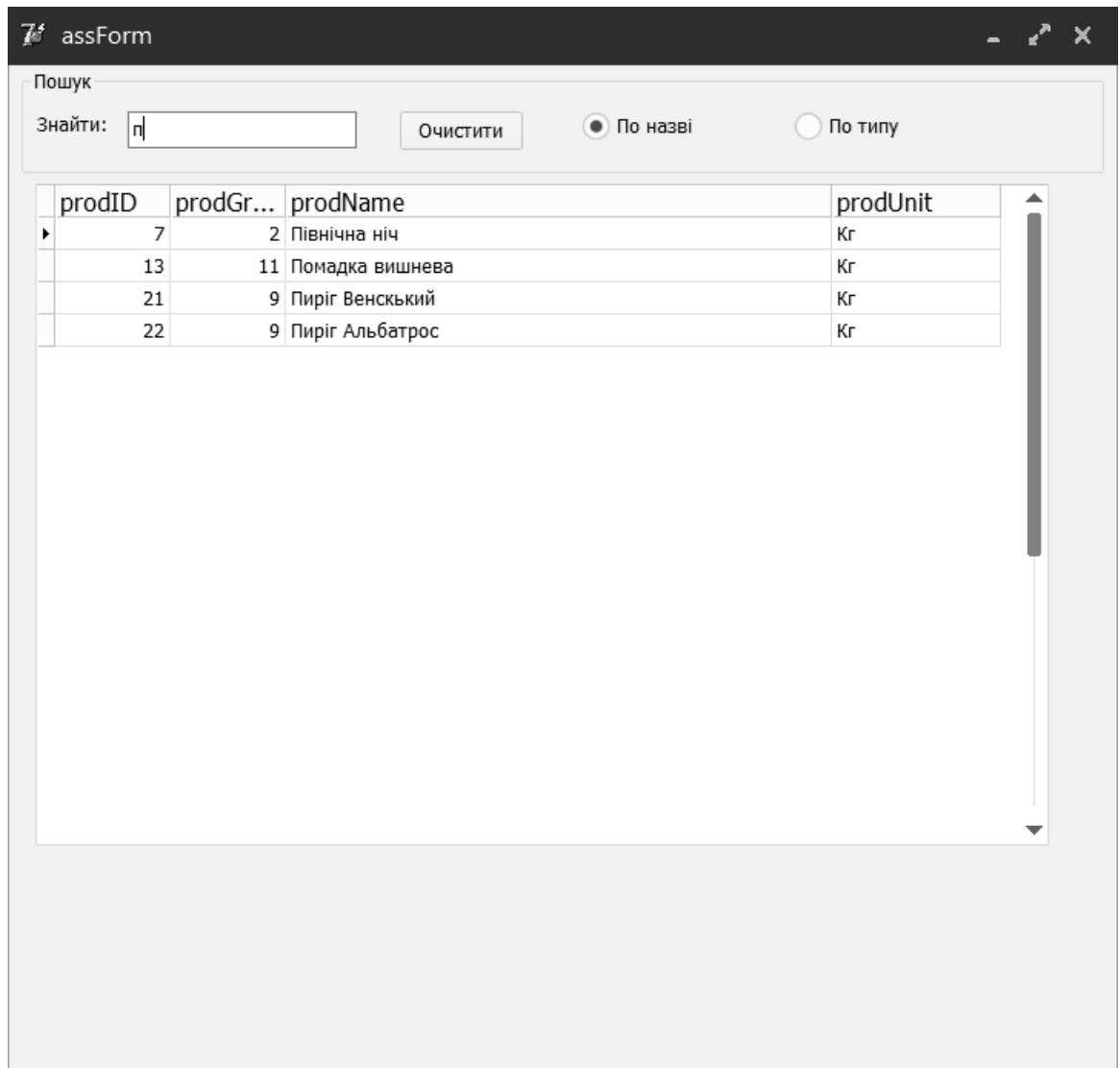


Рисунок А.6 Приклад пошуку по назві

Режим редагування дозволяє редагувати/додати/видалити записи в таблиці. Для активізації слід натиснути ПКМ – Режим редагування або натиснути комбінацію клавіш «Ctrl + E», після чого з'явиться панель редагування.

assForm

Пошук

Знайти: Очистити ☒ По назві ☐ По типу

prodID	prodGr...	prodName	prodUnit
8	6	ОС з лісовим горіхом	Кг
9	6	ОС гіркий	Кг
10	10	Grimasiki	Кг
11	1	Зоряний вечір	Кг
12	2	Ренклюд	Кг
13	11	Помадка вишнева	Кг
15	2	Забодайка молоко	Кг
16	27	Rene black	Кг
17	4	Carbone	Кг
18	3	Ластівка	Кг
19	1	Crostata лимонна	Кг
20	1	Crostata смородинова	Кг
21	9	Пиріг Венський	Кг
22	9	Пиріг Альбатрос	Кг
23	1	Georges premium	Кг
24	1	Курочка ряба	Кг
25	8	Cripsy	Кг
26	5	Duet	Кг
27	16	La prima nuts	Кг

Редагування

Код продукції: Тип продукції: Назва продукції: Одиниця виміру:

⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿

Рисунок А.7 Форма в режимі редагування

Для додавання нового запису необхідно натиснути кнопку  (нижня панель форми), після чого можна буде вводити дані в поля (панель «Редагування»). Після введення необхідно записати дану інформацію, для цього слід натиснути кнопку , після цього інформацію буде додано в таблицю.

Для видалення потрібно натиснути кнопку , після цього в діалоговому вікні натиснути «ОК», якщо рішення про видалення підтверджується або «Отмена» в протилежному випадку. Видалені дані відновленню не підлягають.

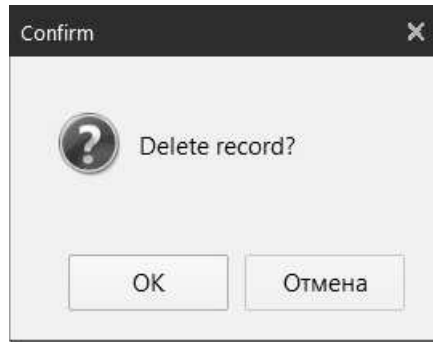


Рисунок А.8 Діалогове вікно видалення запису

Для редагування потрібно обрати необхідний запис, після чого дані будуть відображені в полях панелі «Редагування», де їх можна змінювати, після завершення редагування потрібно натиснути кнопку .

Для виходу з режиму редагування таблиці потрібно натиснути комбінацію клавіш «Ctrl + E» або ПКМ – Режим редагування.

Для створення звіту з плану виробництва або звіту з фактичного продажу потрібно на формі плану/попиту обрати рік за який треба створити план та натиснути кнопку «Сформувати».

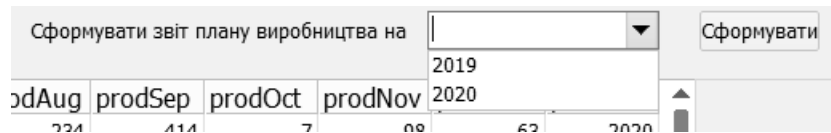


Рисунок А.9 Вибір року для створення звіту

Після чого буде сформований звіт та показано діалогове вікно переходу на створений звіт.

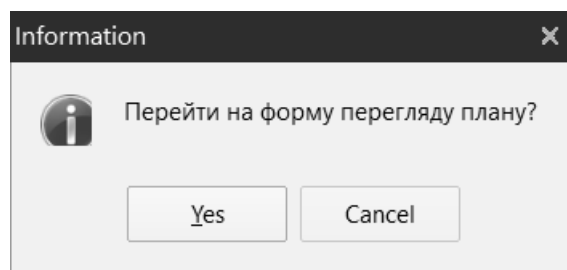


Рисунок А.10 Діалогове вікно переходу

Після переходу буде відкрито створений звіт.

План випуску продукції 2020 року

Назва	Січень	Лютий	Березень	Квітень	Травень	Червень	Липень	Серпень	Вересень	Жовтень	Листопад	Грудень
test	10	22	21	33	33	12	1	234	414	7	98	63
Грім	2	65	7	9	33	53	41	446	31	51	68	23
Помадка вишнева	2	4	7	8	9	0	1	2	3	4	5	6
Марія	23	34	34	52	45	68	79	82	38	48	66	4
La prima nuts	23	741	85	45	32	87	69	85	35	57	487	55
Мрія	87	8	9	65	0	47	4	56	54	889	654	66
Nicole	43	5	4	9	5	2	1	45	65	12	35	48
Ластівка	12	41	47	87	51	23	54	87	954	52	35	48
Grimasiki	2	51	39	43	49	55	51	41	33	34	32	89
Грім	1	1	2	34	2	5	5	5	2	5	3	5

Рисунок А.11 Сформований звіт з плану виробництва

При необхідності звіт можна роздрукувати, для цього треба натиснути на кнопку , на панелі зверху, після чого почнеться друкування звіту.

При виході з програми з'явиться діалогове вікно виходу. Для виходу потрібно натиснути кнопку «ОК».

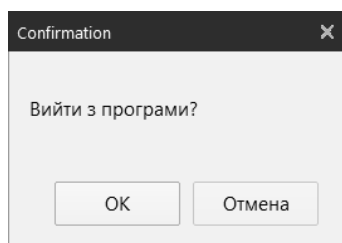


Рисунок А.12 Діалогове вікно виходу

Додаток Б – Код розробки

Лістинг 1. Модуль перевірки прав доступу.

```
unit mainUnit;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, sListBox, sSkinManager, Menus, DB, ADODB,
  sSkinProvider, ComCtrls, Grids, DBGrids, acDBGrid, sGroupBox, ExtCtrls,
  sSplitter, TeeProcs, TeEngine, Chart, Series, Mask, sMaskEdit,
  sCustomComboEdit, sCurrEdit, DbChart, sEdit, sLabel, sCheckBox, sButton,
  sPanel, sListView, MPlayer;
type
  TmainForm = class(TForm)
    MainMenu1: TMainMenu;
    Help1: TMenuItem;
    sSkinManager1: TsSkinManager;
    mainCon: TADOConnection;
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    Set1: TMenuItem;
    Calc1: TMenuItem;
    mniRes1: TMenuItem;
    qry1: TADOQuery;
    Createinvoice1: TMenuItem;
    C1: TMenuItem;
    N1: TMenuItem;
    N2: TMenuItem;
    spnl1: TsPanel;
    lbl1: TsLabel;
    lbl2: TsLabel;
    lbl3: TsLabel;
    btnLog: TsButton;
    chkVis: TsCheckBox;
    edtPass: TsMaskEdit;
    edtLog: TsEdit;
    spnl2: TsPanel;
    lbl4: TsLabel;
    grp1: TsGroupBox;
    btn6: TsButton;
    ..
    btn9: TsButton;
    grp2: TsGroupBox;
    btn5: TsButton;
    btn1: TsButton;
    N3: TMenuItem;
    mp1: TMediaPlayer;
    procedure mniRes1Click(Sender: TObject);
    procedure Set1Click(Sender: TObject);
    procedure chkVisClick(Sender: TObject);
    procedure btnLogClick(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure FormCreate(Sender: TObject);
    procedure Createinvoice1Click(Sender: TObject);
    procedure N1Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure btn1Click(Sender: TObject);
    procedure btn2Click(Sender: TObject);
    procedure btn3Click(Sender: TObject);
    procedure btn4Click(Sender: TObject);
    procedure btn5Click(Sender: TObject);
```

```

    procedure N3Click(Sender: TObject);
    public procedure numberInput(Sender: TObject; var Key: Char); //custom
    public procedure charInput(Sender: TObject; var Key: Char); //custom
private
    { Private declarations }
public
    { Public declarations }
end;
var
    MainForm: TmainForm;
implementation
uses assUnit, resUnit, costUnit, planUnit, groupUnit, calcUnit, settingsUnit,
IniFiles,
    reportUnit, userUnit, invoiceUnit, factUnit, reportForm2;
{$R *.dfm}
procedure TmainForm.mniRes1Click(Sender: TObject);
begin
    calcForm.Show;
end;
procedure TmainForm.Set1Click(Sender: TObject);
begin
    SettingsForm.Show;
end;
procedure TmainForm.chkVisClick(Sender: TObject);
begin
    if (chkVis.Checked = true) then edtPass.PasswordChar:= #0
    else edtPass.PasswordChar:= '*';
end;
procedure TmainForm.btnLogClick(Sender: TObject);
var Ini: TIniFile;
begin
    if (edtLog.Text = 'sans') then
        mp1.Play;
    if (edtLog.Text <> 'admin') or (edtPass.Text <> '') then
        ShowMessage('Логін або пароль введені невірно!')
    else begin
        c1.Enabled:=true;
        n3.enabled:=true;
        spn11.Visible:=false;
        spn12.Visible:=true;
        Ini:= TIniFile.Create('..\config.ini');
        try
            //Edit1.Enabled:= Ini.ReadBool('TMenuItem', 'EditEnabled', true);
            Calc1.Enabled:= Ini.ReadBool('TMenuItem', 'CalcEnabled', true);
            Set1.Enabled:= Ini.ReadBool('TMenuItem', 'SetEnabled', true);
            //Account1.Enabled:= Ini.ReadBool('TMenuItem', 'AccEnabled', true);
            Help1.Enabled:= Ini.ReadBool('TMenuItem', 'HelpEnabled', true);
        finally
            Ini.Free;
        end;
    end;
end;
procedure TmainForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    CanClose:= MessageBox(Handle, 'Вийти з програми?', 'Confirmation', MB_OKCANCEL)
= mrOk;
end;
procedure TmainForm.FormCreate(Sender: TObject);
var DevMode: Boolean;
begin
    DevMode:= False;
    if DevMode = True then
        begin
            MainForm.Caption:=mainForm.Caption + ' (work in Dev Mode)';

```

```

        MainForm.sSkinManager1.SkinName:= 'Material Dark';
    end;
    spnl1.Show;
    spnl2.Visible:=false;
end;
procedure TMainForm.numberInput(Sender: TObject; var Key: Char);
begin
    if not(Key in['0'..'9', '.', #8]) then Key:= #0;
end;
procedure TMainForm.charInput(Sender: TObject; var Key: Char);
begin
    if not(Key in['a'..'z', 'A'..'Z']) then Key:= #0;
end;
procedure TmainForm.Createinvoice1Click(Sender: TObject);
begin
    invoiceForm.Show;
end;
procedure TmainForm.N1Click(Sender: TObject);
begin
    factForm.Show;
end;
procedure TmainForm.N2Click(Sender: TObject);
begin
    planform.Show;
end;
procedure TmainForm.btn1Click(Sender: TObject);
begin
    assForm.Show;
end;
procedure TmainForm.btn2Click(Sender: TObject);
begin
    userForm.Show;
end;
procedure TmainForm.btn3Click(Sender: TObject);
begin
    groupform.Show;
end;
procedure TmainForm.btn4Click(Sender: TObject);
begin
    resform.Show;
end;
procedure TmainForm.btn5Click(Sender: TObject);
begin
    reportform.qckrp1.Preview;
end;
procedure TmainForm.N3Click(Sender: TObject);
begin
    reportUnit2.qckrp2.Preview;
end;
end.

```

Лістинг 2. Модуль роботи з товаром (assUnit).

```

unit assUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, Grids, DBGrids, acDBGrid, StdCtrls, sGroupBox, Menus,
    Mask, DBCtrls, sDBEdit, ExtCtrls, sPanel, sDBNavigator, sLabel,
    sDBComboBox, sDBLookupComboBox, sButton, sCheckBox, sEdit, sRadioButton;
type
    TassForm = class(TForm)
        grpTable: TsGroupBox;
        grpEdit: TsGroupBox;
        sDBGrid1: TsDBGrid;

```

```

    tbl: TADOTable;
    DataSource1: TDataSource;
    PopupMenu1: TPopupMenu;
    nEdit: TMenuItem;
    sDBNavigator1: TsDBNavigator;
    sDBEdit1: TsDBEdit;
...
    sLabel1: TsLabel;
...
    sDBLookupComboBox1: TsDBLookupComboBox;
    grpSearch: TsGroupBox;
    sLabel5: TsLabel;
    edtSearch: TsEdit;
    btnClear: TsButton;
    nSearch: TMenuItem;
    dsSearch: TDataSource;
    qSearch: TADOQuery;
    chkGroup: TsRadioButton;
    chkName: TsRadioButton;
    procedure nEditClick(Sender: TObject);
    procedure sDBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
    procedure nSearchClick(Sender: TObject);
    procedure edtSearchChange(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure chkNameValueChanged(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
End;
var
    assForm: TassForm;
implementation
{$R *.dfm}
uses mainUnit, groupUnit;
procedure TassForm.nEditClick(Sender: TObject);
begin
    if grpEdit.Visible = true then
        grpEdit.Visible:= false
    else grpEdit.Visible:= true;
end;
procedure TassForm.sDBNavigator1Click(Sender: TObject;
    Button: TNavigateBtn);
begin
    if Button = nbInsert then begin
        sDBEdit1.SetFocus;
    end;
end;
procedure TassForm.nSearchClick(Sender: TObject);
begin
    if grpSearch.Visible = true then
        grpSearch.Visible:= false
    else grpSearch.Visible:= true;
end;
procedure TassForm.edtSearchChange(Sender: TObject);
begin
    if edtSearch.Text <> '' then
        sDBGrid1.DataSource:= dsSearch
    else sDBGrid1.DataSource:= DataSource1;
    if chkGroup.Checked = true then begin
        qSearch.Close;
        qSearch.SQL.Clear;

```

```

        qSearch.SQL.Add('SELECT * FROM Assortment WHERE prodGroup LIKE ''' +
edtSearch.Text + '%''');
        qSearch.Open;
    end;

    if chkName.Checked = true then begin
        qSearch.Close;
        qSearch.SQL.Clear;
        qSearch.SQL.Add('SELECT * FROM Assortment WHERE prodName LIKE ''' +
edtSearch.Text + '%''');
        qSearch.Open;
    end;
end;
procedure TassForm.btnCancelClick(Sender: TObject);
begin
    edtSearch.Text:= '';
end;
procedure TassForm.FormCreate(Sender: TObject);
begin
    sDBEdit1.OnKeyPress:= mainForm.numberInput;
end;
procedure TassForm.chkNameValueChanged(Sender: TObject);
begin
    edtSearch.SetFocus;
end;
end.

```

Лістинг 3. Модуль розрахунку ресурсів (clacUnit)

```

unit calcUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, DBCtrls, sDBComboBox, StdCtrls, sComboBox,
    sDBLookupComboBox, Grids, DBGrids, acDBGrid, sLabel, sGroupBox;
type
    TcalcForm = class(TForm)
        ds1: TDataSource;
        qry1: TADOQuery;
        sDBGrid1: TsDBGrid;
        grpStats: TsGroupBox;
        sLabel1: TsLabel;
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    calcForm: TcalcForm;
implementation
uses mainUnit;
{$R *.dfm}
procedure TcalcForm.FormShow(Sender: TObject);
begin
    qry1.Close;
    qry1.ExecSQL;
    qry1.Open;
end;
end.

```

Лістинг 4. Модуль нормо-витрат (costUnit)

```
unit costUnit;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, DB, ADODB, StdCtrls, Mask, DBCtrls, sDBEdit, ExtCtrls,
  sPanel, sDBNavigator, sLabel, Grids, DBGrids, acDBGrid, sGroupBox,
  sButton, sCheckBox, sEdit, sDBLookupComboBox, sRadioButton;
type
  TcostForm = class(TForm)
    grpTable: TsGroupBox;
    sDBGrid1: TsDBGrid;
    grpEdit: TsGroupBox;
    sLabel1: TsLabel;
    ...
    sDBNavigator1: TsDBNavigator;
    sDBEdit3: TsDBEdit;
    tbl: TADOTable;
    DataSource1: TDataSource;
    PopupMenu1: TPopupMenu;
    nEdit: TMenuItem;
    nSearch: TMenuItem;
    grpSearch: TsGroupBox;
    sLabel6: TsLabel;
    edtSearch: TsEdit;
    sDBLookupComboBox1: TsDBLookupComboBox;
    sDBLookupComboBox2: TsDBLookupComboBox;
    btnSearch: TsButton;
    rbVisName: TsRadioButton;
    rbVisID: TsRadioButton;
    qrySearch: TADOQuery;
    dsSearch: TDataSource;
    procedure nEditClick(Sender: TObject);
    procedure nSearchClick(Sender: TObject);
    procedure sDBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
    procedure btnSearchClick(Sender: TObject);
    procedure edtSearchChange(Sender: TObject);
    procedure rbVisNameValueChanged(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  costForm: TcostForm;
implementation
{$R *.dfm}
uses mainUnit, assUnit, resUnit;
procedure TcostForm.nEditClick(Sender: TObject);
begin
  if grpEdit.Visible = true then
    grpEdit.Visible:= false
  else grpEdit.Visible:= true;
end;
procedure TcostForm.nSearchClick(Sender: TObject);
begin
  if grpSearch.Visible = true then
    grpSearch.Visible:= false
  else grpSearch.Visible:= true;
end;
procedure TcostForm.sDBNavigator1Click(Sender: TObject;
  Button: TNavigateBtn);
begin
```



```

        if Button = nbInsert then sDBLookupComboBox1.SetFocus;
    end;
    procedure TcostForm.btnSearchClick(Sender: TObject);
    begin
        edtSearch.text:='';
    end;
    procedure TcostForm.edtSearchChange(Sender: TObject);
    begin
        if edtSearch.Text <> '' then
            sDBGrid1.DataSource:= dsSearch
        else sDBGrid1.DataSource:= DataSource1;
        if rbVisName.Checked = true then begin
            qrySearch.Close;
            qrySearch.SQL.Clear;
            qrySearch.SQL.Add('SELECT * FROM CostRate WHERE prodID LIKE ''%' +
            edtSearch.Text + '%''');
            qrySearch.Open;
        end;
        if rbVisID.Checked = true then begin
            qrySearch.Close;
            qrySearch.SQL.Clear;
            qrySearch.SQL.Add('SELECT * FROM CostRate WHERE resID LIKE ''%' +
            edtSearch.Text + '%''');
            qrySearch.Open;
        end;
    end;
    procedure TcostForm.rbVisNameValueChanged(Sender: TObject);
    begin
        edtSearch.SetFocus;
    end;
end.

```

Лістинг 5. Модуль типів продукції (groupUnit)

```

unit groupUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Menus, DB, ADODB, StdCtrls, Mask, DBCtrls, sDBEdit, ExtCtrls,
    sPanel, sDBNavigator, sLabel, Grids, DBGrids, acDBGrid, sGroupBox,
    sButton, sCheckBox, sEdit, sRadioButton;
type
    TgroupForm = class(TForm)
        grpTable: TsGroupBox;
        sDBGrid1: TsDBGrid;
        grpEdit: TsGroupBox;
        sLabel1: TsLabel;
        sLabel2: TsLabel;
        sDBNavigator1: TsDBNavigator;
        sDBEdit1: TsDBEdit;
        sDBEdit2: TsDBEdit;
        ADOTable1: TADOTable;
        DataSource1: TDataSource;
        PopupMenu1: TPopupMenu;
        nEdit: TMenuItem;
        nSearch: TMenuItem;
        grpSearch: TsGroupBox;
        sLabel6: TsLabel;
        btnClear: TsButton;
        rbVisName: TsRadioButton;
        rbVisID: TsRadioButton;
        edtSearch: TsEdit;
        qrySearch: TADOQuery;
    end;

```

```

    dsSearch: TDataSource;
    procedure nEditClick(Sender: TObject);
    procedure nSearchClick(Sender: TObject);
    procedure edtSearchChange(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure rbVisNameValueChanged(Sender: TObject);
    { Private declarations }
public
    { Public declarations }
end;
var
    groupForm: TgroupForm;
implementation
{$R *.dfm}
uses mainUnit, assUnit;
procedure TgroupForm.nEditClick(Sender: TObject);
begin
    if grpEdit.Visible = true then
        grpEdit.Visible:= false
    else grpEdit.Visible:= true;
end;
procedure TgroupForm.nSearchClick(Sender: TObject);
begin
    if grpSearch.Visible = true then
        grpSearch.Visible:= false
    else grpSearch.Visible:= true;
end;
procedure TgroupForm.edtSearchChange(Sender: TObject);
begin
    if edtSearch.Text <> '' then
        sDBGGrid1.DataSource:= dsSearch
    else sDBGGrid1.DataSource:= DataSource1;
    if rbVisName.Checked = true then begin
        qrySearch.Close;
        qrySearch.SQL.Clear;
        qrySearch.SQL.Add('SELECT * FROM ProdGroup WHERE grpName LIKE ''%' +
edtSearch.Text + '%''');
        qrySearch.Open;
    end;
    if rbVisID.Checked = true then begin
        qrySearch.Close;
        qrySearch.SQL.Clear;
        qrySearch.SQL.Add('SELECT * FROM ProdGroup WHERE grpID LIKE ''%' +
edtSearch.Text + '%''');
        qrySearch.Open;
    end;
end;
procedure TgroupForm.btnClearClick(Sender: TObject);
begin
    edtSearch.Text:='';
end;
procedure TgroupForm.rbVisNameValueChanged(Sender: TObject);
begin
    edtSearch.SetFocus;
end;
end.

```

Лістинг 6. Модуль роботи з накладними (invoiceUnit)

```

unit invoiceUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

```

    Dialogs, StdCtrls, sComboBox, DB, ADODB, sLabel, Grids, DBGrids,
    acDBGrid, sGroupBox, ExtCtrls, sPanel, sMonthCalendar, DateUtils;
type
    TinvoiceForm = class(TForm)
        grpTable: TsGroupBox;
        sDBGrid1: TsDBGrid;
        grpSelect: TsGroupBox;
        sLabel5: TsLabel;
        dsSearch: TDataSource;
        qSearch: TADOQuery;
        sCal: TsMonthCalendar;
        procedure grpSelectMouseMove(Sender: TObject; Shift: TShiftState; X,
            Y: Integer);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    invoiceForm: TinvoiceForm;
implementation
{$R *.dfm}
procedure TinvoiceForm.grpSelectMouseMove(Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
var day, month, year: word;
begin
    DecodeDate(sCal.CalendarDate, year, month, day);
    qSearch.Close;
    qSearch.Sql.Clear;
    qSearch.SQL.Add('SELECT NaklPostZag.naklID, NaklPostZag.DateN,
    NaklPostZag.userID, NaklPostZm.prodID, NaklPostZm.prodCount FROM NaklPostZag,
    NaklPostZm');
    qSearch.SQL.Add('WHERE (Year([DateN]) = ''' + IntToStr(year) + ''' and
    (Month([DateN]) = ''' + inttostr(month) + '''))');
    qSearch.SQL.Add('AND NaklPostZag.naklID = NaklPostZm.naklID');
    qSearch.Open;
end;
end.

```

Лістинг 7. Модуль роботи з планом (planUnit)

```

unit planUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Menus, DB, ADODB, StdCtrls, Mask, DBCtrls, sDBEdit, ExtCtrls,
    sPanel, sDBNavigator, sLabel, Grids, DBGrids, acDBGrid, sGroupBox,
    sButton, sCheckBox, sEdit, sRadioButton, sDBLookupComboBox, Buttons,
    sSpeedButton, TeEngine, Series, TeeProcs, Chart, DbChart, sComboBox,
    sListBox, sMemo;
type
    TplanForm = class(TForm)
        tbl: TADOTable;
        DataSource1: TDataSource;
        PopupMenu1: TPopupMenu;
        nEdit: TMenuItem;
        nSearch: TMenuItem;
        qry1: TADOQuery;
        ds1: TDataSource;
        grpSearch: TsGroupBox;
        lbl1: TsLabel;
        btn1: TsSpeedButton;
        edtSearch: TsEdit;
    end;

```

```

    btnSearch: TsButton;
    rbID: TsRadioButton;
    rbYear: TsRadioButton;
    sdbgrd1: TsDBGrid;
    grpEdit: TsGroupBox;
    sLabel1: TsLabel;
    ..
    sLabel15: TsLabel;
    sDBNavigator1: TsDBNavigator;
    sDBEdit2: TsDBEdit;
    ..
    sDBEdit14: TsDBEdit;
    sDBLookupComboBox1: TsDBLookupComboBox;
    dbcht1: TDBChart;
    brsrsSeries1: TBarSeries;
    lbl2: TsLabel;
    cbb1: TsComboBox;
    btn2: TsButton;
    procedure nEditClick(Sender: TObject);
    procedure nSearchClick(Sender: TObject);
    procedure btnSearchClick(Sender: TObject);
    procedure edtSearchChange(Sender: TObject);
    procedure sdbgrd1GetCellText(Sender: TObject; Field: TField;
        var Text: String);
    procedure btn1Click(Sender: TObject);
    procedure edtSearchKeyPress(Sender: TObject; var Key: Char);
    procedure FormCreate(Sender: TObject);
    procedure btn2Click(Sender: TObject);
    procedure btn3Click(Sender: TObject);
    procedure sDBNavigator1BeforeAction(Sender: TObject;
        Button: TNavigateBtn);
    procedure sDBNavigator1MouseMove(Sender: TObject; Shift: TShiftState;
        X, Y: Integer);
    procedure sDBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    planForm: TplanForm;
implementation
{$R *.dfm}
uses mainUnit, assUnit, factUnit, settingsUnit, reportUnit, reportForm2;
procedure TplanForm.Nsearchclick(Sender: TObject);
begin
    if grpSearch.Visible = true then
        grpSearch.Visible:= false
    else grpSearch.Visible:= true;
end;
procedure TplanForm.Neditclick(Sender: TObject);
begin
    if grpEdit.Visible = true then
        grpEdit.Visible:= false
    else grpEdit.Visible:= true;
end;
procedure TplanForm.edtSearchChange(Sender: TObject);
begin
    {if settingsform.chkfastset.Checked = false then
    begin
        if edtSearch.Text = '' then
            sDBGrd1.DataSource:= ds1;
        end;}
    //if ((btn1.Enabled = false) and (edtsearch.Text = '')) then

```

```

//tbl1.filtered:=false;
if settingsform.chkfastset.Checked = true then
begin
btn1.Enabled:= false;
if edtSearch.Text <> '' then
sDBGrd1.DataSource:= ds1
else sDBGrd1.DataSource:= datasourcel;
if rbID.Checked = true then begin
qry1.Close;
qry1.SQL.Clear;
qry1.SQL.Add('SELECT * FROM Plan WHERE prodID LIKE ''%' + edtSearch.Text +
'%''');
qry1.Open;
end;
if rbYear.Checked = true then begin
qry1.Close;
qry1.SQL.Clear;
qry1.SQL.Add('SELECT * FROM Plan WHERE prodYear LIKE ''%' + edtSearch.Text +
'%''');
qry1.Open;
end;
end;
end;
procedure TplanForm.btnSearchClick(Sender: TObject);
begin
tbl.Filtered:=false;
edtsearch.Text:='';
end;
procedure TplanForm.sdbgrd1GetCellText(Sender: TObject; Field: TField;
var Text: String);
begin
dbcht1.Title.text.Clear;
dbcht1.title.text.Add('Товар ' + chr(39) + sdblookupcombobox1.Text + chr(39));
end;
procedure TplanForm.btn1Click(Sender: TObject);
begin
if edtSearch.Text = '' then
MessageBox(0, 'Помилка пошуку, поле для вводу не може бути пустим', 'Помилка',
MB_ICONERROR + MB_OK);
if settingsform.chkfastset.Checked = false then
begin
if rbID.Checked = true then begin
tbl.Filter:='prodID=' + edtsearch.Text;
tbl.Filtered := True;
end;
if rbYear.Checked = true then begin
tbl.Filter:='prodYear='+ edtsearch.Text;
tbl.Filtered := True;
end;
end;
end;
end;
procedure TplanForm.edtSearchKeyPress(Sender: TObject; var Key: Char);
begin
if key = #13 then begin
planform.btn1.Click;
end;
if key = #27 then begin
planform.btnSearch.click;
end;
end;
end;
procedure TplanForm.FormCreate(Sender: TObject);
var i:integer;
begin
for i:=0 to sdbgrd1.Columns.Count - 1 do

```

```

    sdbgrd1.columns[i].Width:=70;
end;
procedure TplanForm.btn2Click(Sender: TObject);
begin
    {reportform.qry1.Close;
    reportform.qry1.SQL.Clear;
    reportform.qry1.SQL.Add('SELECT * FROM asstest WHERE prodYear LIKE ''%' +
cbb1.Text + '%''');
    reportform.qry1.Open;}
    reportform.tbl1.Filter:='prodYear=' + cbb1.Text;
    reportUnit2.tbl1.Filter:='prodYear=' + cbb1.Text;
    reportform.tbl1.Filtered := True;
    reportUnit2.tbl1.Filtered := True;
    reportform.qrlbl15.Caption:='План випуску продукції ' + cbb1.Text + ' року';
    reportUnit2.qrlbl14.Caption:='Кількість випуску продукції за ' + cbb1.Text +
' рік щоквартально';
    if MessageDlg('Перейти на форму перегляду
плану?',mtInformation,[mbYes,mbCancel], 0) = mrYes
    then reportform.qckrpl.Preview;
end;
procedure TplanForm.btn3Click(Sender: TObject);
begin
    if ((sdbgrd1.Columns[0].Field = sdblookupcombobox1.Field) and
(sdbgrd1.Columns[13].Field = sdbedit14.field))then
    begin
        showmessage('asd');
        sdbgrd1.SetFocus;
    end;
end;
procedure TplanForm.sDBNavigator1BeforeAction(Sender: TObject;
    Button: TNavigateBtn);
begin
    {if button = nbpost then
        if ((sdbgrd1.Columns[0].Field = sdblookupcombobox1.Field) and
(sdbgrd1.Columns[13].Field = sdbedit14.field))then
            //if ((sdblookupcombobox1.field = tbl.FieldByName('prodID').AsString) and
(sdbedit14.Text = sdbgrd1.Columns[13].Field))then
            begin
                MessageBox(0, 'Неможливо додати новий запис. Товар має запис на введений
рік. Перевірте правильність введених даних.', 'title', MB_ICONWARNING + MB_OK);
                tbl.Cancel;
                //abort;
                //tbl.CancelUpdates();
                //tbl.Edit;
                //sdbgrd1.DataSource.dataset.Cancel;
                //sdbgrd1.DataSource.dataset.ca
                //sdbgrd1.DataSource.DataSet.afterinsert(tbl);
                //MessageBox(0, 'text', 'title', MB_ICONWARNING + MB_OK);
                //sdbgrd1.SetFocus;
                //abort;
            end; }
end;
procedure TplanForm.sDBNavigator1MouseMove(Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
begin
    //sdblookupcombobox1.SetFocus;
end;
procedure TplanForm.sDBNavigator1Click(Sender: TObject;
    Button: TNavigateBtn);
begin
    { if button = nbpost then
        if ((sdbgrd1.Columns[0].Field = sdblookupcombobox1.Field) and
(sdbgrd1.Columns[13].Field = sdbedit14.field))then
        begin

```

```

        //tbl.Delete();
        MessageBox(0, 'text', 'title', MB_ICONWARNING + MB_OK);
        tbl.Cancel;
        abort;
        tbl.Post;
    end;
    end;
end.

```

Лістинг 8. Модуль створення звітів (reportUnit)

```

unit reportUnit;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, QuickRpt, QRCtrls, DB, ADODB, ExtCtrls, StdCtrls, sButton;

type
    TreportForm = class(TForm)
        tbl1: TADOTable;
        qry1: TADOQuery;
        qckrpl: TQuickRep;
        qrbndPageHeaderBand1: TQRBand;
        qrbndPageFooterBand1: TQRBand;
        qrbndTitleBand1: TQRBand;
        qrlbl1: TQRLabel;
        qrbnd1: TQRBand;
        qrdbtxtprodJan1: TQRDBText;
        qrlbl2: TQRLabel;
        qrlbl7: TQRLabel;
        qrlbl8: TQRLabel;
        qrlbl9: TQRLabel;
        qrlbl10: TQRLabel;
        qrlbl11: TQRLabel;
        qrlbl12: TQRLabel;
        qrlbl13: TQRLabel;
        qrdbtxtprodJan: TQRDBText;
        qrdbtxtprodFeb: TQRDBText;
        qrdbtxtprodMar: TQRDBText;
        qrdbtxtprodApr: TQRDBText;
        qrdbtxtprodMay: TQRDBText;
        qrdbtxtprodJun: TQRDBText;
        qrdbtxtprodJul: TQRDBText;
        qrdbtxtprodAug: TQRDBText;
        qrdbtxtprodSep: TQRDBText;
        qrdbtxtprodOct: TQRDBText;
        qrdbtxtprodNov: TQRDBText;
        qrdbtxtprodDec: TQRDBText;
        qrlbl3: TQRLabel;
        qrlbl4: TQRLabel;
        qrlbl5: TQRLabel;
        qrlbl6: TQRLabel;
        qrlbl15: TQRLabel;
        procedure FormCreate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var

```

```

    reportForm: TreportForm;

implementation

uses mainUnit, planUnit, assUnit;

{$R *.dfm}

procedure TreportForm.FormCreate(Sender: TObject);
begin
    {SELECT TOP 5 Assortment.prodName, prodJan, prodFeb, prodMar, prodApr, prodMay,
    prodJun, prodJul, prodAug, prodSep, prodNov, prodOct, prodDec
    FROM Plan, Assortment
    WHERE (prodYear = 2019 and Assortment.prodID = Plan.prodID)}

    {qry1.Close;
    qry1.SQL.Clear;
    qry1.SQL.Add('SELECT * FROM fact ');
    qry1.Open;}
    {reportform.qry1.Close;
    reportform.qry1.SQL.Clear;
    reportform.qry1.SQL.Add('SELECT * FROM asstest WHERE prodYear = 2019');
    reportform.qry1.Open; }

end;

end.

```

Лістинг 9. Модуль роботи з ресурсами (resUnit)

```

unit resUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Menus, DB, ADODB, StdCtrls, Mask, DBCtrls, sDBEdit, ExtCtrls,
    sPanel, sDBNavigator, sLabel, Grids, DBGrids, acDBGrid, sGroupBox,
    sButton, sCheckBox, sEdit, sRadioButton;
type
    TresForm = class(TForm)
        grpTable: TsGroupBox;
        sDBGrid1: TsDBGrid;
        grpEdit: TsGroupBox;
        sLabel1: TsLabel;
        ...
        sDBNavigator1: TsDBNavigator;
        sDBEdit1: TsDBEdit;
        ...
        t11: TADOTable;
        DataSource1: TDataSource;
        PopupMenu1: TPopupMenu;
        nEdit: TMenuItem;
        nSearch: TMenuItem;
        grpSearch: TsGroupBox;
        sLabel6: TsLabel;
        edtSearch: TsEdit;
        btnSearch: TsButton;
        rbVisName: TsRadioButton;
        rbVisID: TsRadioButton;
        qrySearch: TADOQuery;
    end;

```



```

    dsSearch: TDataSource;
    procedure nEditClick(Sender: TObject);
    procedure nSearchClick(Sender: TObject);
    procedure sDBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
    procedure edtSearchChange(Sender: TObject);
    procedure btnSearchClick(Sender: TObject);
    procedure rbVisNameValueChanged(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    resForm: TresForm;
implementation
{$R *.dfm}
uses mainUnit;
procedure TresForm.nEditClick(Sender: TObject);
begin
    if grpEdit.Visible = true then
        grpEdit.Visible:= false
    else grpEdit.Visible:= true;
end;
procedure TresForm.nSearchClick(Sender: TObject);
begin
    if grpSearch.Visible = true then
        grpSearch.Visible:= false
    else grpSearch.Visible:= true;
end;
procedure TresForm.sDBNavigator1Click(Sender: TObject;
    Button: TNavigateBtn);
begin
    if Button = nbInsert then sBEdit1.SetFocus;
end;
procedure TresForm.edtSearchChange(Sender: TObject);
begin
    if edtSearch.Text <> '' then
        sDBGrid1.DataSource:= dsSearch
    else sDBGrid1.DataSource:= DataSource1;
    if rbVisName.Checked = true then begin
        qrySearch.Close;
        qrySearch.SQL.Clear;
        qrySearch.SQL.Add('SELECT * FROM Resources WHERE resName LIKE ''%' +
edtSearch.Text + '%''');
        qrySearch.Open;
    end;
    if rbVisID.Checked = true then begin
        qrySearch.Close;
        qrySearch.SQL.Clear;
        qrySearch.SQL.Add('SELECT * FROM Resources WHERE resID LIKE ''%' +
edtSearch.Text + '%''');
        qrySearch.Open;
    end;
end;
procedure TresForm.btnSearchClick(Sender: TObject);
begin
    edtSearch.Text:='';
end;
procedure TresForm.rbVisNameValueChanged(Sender: TObject);
begin
    edtSearch.SetFocus;
end;
end.

```

Лістинг 10. Модуль налаштувань (settingsUnit)

```
unit settingsUnit;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, sComboBox, sLabel, ComCtrls, sComboBoxes, sButton,
  ExtCtrls, sPanel, acSlider, sGroupBox, sCheckBox;
type
  TSettingsForm = class(TForm)
    sGroupBox1: TsGroupBox;
    lblLoginLabel: TsLabel;
    lblLoginLabel1: TsLabel;
    lblLogin1: TsLabel;
    sknslctrl: TsSkinSelector;
    cbbLangCBB: TsComboBox;
    btnResetButton: TsButton;
    sldr1: TsSlider;
    lbl1: TsLabel;
    chkfastset: TsCheckBox;
    procedure sldr1Changing(Sender: TObject; var CanChange: Boolean);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btnResetButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  SettingsForm: TSettingsForm;
implementation
uses mainUnit, IniFiles, factUnit;
{$R *.dfm}
procedure TSettingsForm.sldr1Changing(Sender: TObject;
  var CanChange: Boolean);
begin
  //mainForm.sSkinManager1.SkinName:='Windows 10';
  if mainForm.sSkinManager1.SkinName= 'GPlus' then begin
    mainForm.sSkinManager1.SkinName:= 'Material Dark';
  end
  else mainForm.sSkinManager1.SkinName:= 'GPlus';
end;
procedure TSettingsForm.FormClose(Sender: TObject;
  var Action: TCloseAction);
var Ini: TIniFile; Lang:TIniFile;
begin
  Ini:= TIniFile.Create('.\config.ini');
  Ini.WriteInteger('TsComboBox', 'lanItemIndex', cbbLangCBB.ItemIndex);
  Ini.WriteInteger('TsComboBox', 'styleItemIndex', sknslctrl.ItemIndex);
  Ini.WriteBool('TsComboBox', 'SliderOn', sldr1.SliderOn);
  Lang:= TIniFile.Create('.\lang.ini');
  if cbbLangCBB.Text='Українська' then
    SettingsForm.Caption:= Lang.ReadString('SettingsForm', 'Caption',
SettingsForm.Caption);
  Lang.Free;
  Ini.Free;
end;
procedure TSettingsForm.btnResetButtonClick(Sender: TObject);
begin
  CopyFile('..\Kursach.mdb', '../backup/Kursach_back.mdb', false);
end;
end.
```

Лістинг 11. Модуль для роботи з замовниками (userUnit)

```
unit userUnit;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ADODB, Menus, DB, StdCtrls, sRadioButton, sButton, sEdit, Mask,
  DBCtrls, sDBEdit, ExtCtrls, sPanel, sDBNavigator, sLabel, Grids, DBGrids,
  acDBGrid, sGroupBox;
type
  TuserForm = class(TForm)
    grpTable: TsGroupBox;
    sDBGrid1: TsDBGrid;
    grpEdit: TsGroupBox;
    sLabel1: TsLabel;
    ...
    sDBNavigator1: TsDBNavigator;
    sDBEdit1: TsDBEdit;
    ...
    grpSearch: TsGroupBox;
    sLabel5: TsLabel;
    edtSearch: TsEdit;
    btnClear: TsButton;
    chkID: TsRadioButton;
    chkName: TsRadioButton;
    tbl1: TADOTable;
    DataSource1: TDataSource;
    PopupMenu1: TPopupMenu;
    nEdit: TMenuItem;
    nSearch: TMenuItem;
    dsSearch: TDataSource;
    qSearch: TADOQuery;
    sDBEdit2: TsDBEdit;
    procedure edtSearchChange(Sender: TObject);
    procedure nEditClick(Sender: TObject);
    procedure nSearchClick(Sender: TObject);
    procedure chkIDValueChanged(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  userForm: TuserForm;
implementation
{$R *.dfm}
procedure TuserForm.edtSearchChange(Sender: TObject);
begin
  if edtSearch.Text <> '' then
    sDBGrid1.DataSource:= dsSearch
  else sDBGrid1.DataSource:= DataSource1;
  if chkName.Checked = true then begin
    qSearch.Close;
    qSearch.SQL.Clear;
    qSearch.SQL.Add('SELECT * FROM Users WHERE userName LIKE ''%' +
edtSearch.Text + '%''');
    qSearch.Open;
  end;
  if chkID.Checked = true then begin
    qSearch.Close;
    qSearch.SQL.Clear;
    qSearch.SQL.Add('SELECT * FROM Users WHERE userID LIKE ''%' + edtSearch.Text
+ '%''');
```

```

        qSearch.Open;
    end;
end;
procedure TuserForm.nEditClick(Sender: TObject);
begin
    if grpEdit.Visible = true then
        grpEdit.Visible:= false
    else grpEdit.Visible:= true;
end;
procedure TuserForm.nSearchClick(Sender: TObject);
begin
    if grpSearch.Visible = true then
        grpSearch.Visible:= false
    else grpSearch.Visible:= true;
end;
procedure TuserForm.chkIDValueChanged(Sender: TObject);
begin
    edtSearch.setfocus;
end;
end.

```

Лістинг 12. Модуль фактичного виготовлення продукції (factUnit)

```

unit factUnit;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, Grids, DBGrids, acDBGrid, DBCtrls, sDBLookupComboBox,
    Mask, sDBEdit, ExtCtrls, sPanel, sDBNavigator, StdCtrls, sRadioButton,
    sButton, sEdit, sLabel, sGroupBox, sMaskEdit, sCustomComboEdit,
    sComboEdit, acPopupCtrls, Menus, TeEngine, Series, TeeProcs, Chart,
    DbChart, Buttons, sSpeedButton;
type
    TfactForm = class(TForm)
        grpSearch: TsGroupBox;
        lbl1: TsLabel;
        edtSearch: TsEdit;
        btnSearch: TsButton;
        rbID: TsRadioButton;
        rbYear: TsRadioButton;
        grpEdit: TsGroupBox;
        lbl2: TsLabel;
        ..
        lbl15: TsLabel;
        sdbnvgtrl: TsDBNavigator;
        edt1: TsDBEdit;
        ..
        edt13: TsDBEdit;
        cbb1: TsDBLookupComboBox;
        sdbgrdl: TsDBGrid;
        tbl1: TADOTable;
        qry1: TADOQuery;
        ds1: TDataSource;
        ds2: TDataSource;
        pml: TPopupMenu;
        N1: TMenuItem;
        N2: TMenuItem;
        dbcht1: TDBChart;
        brsrsSeries1: TBarSeries;
        btn1: TsSpeedButton;
        procedure N1Click(Sender: TObject);
        procedure N2Click(Sender: TObject);
    end;

```

```

    procedure sdbgrd1GetCellText(Sender: TObject; Field: TField;
        var Text: String);
    procedure edtSearchChange(Sender: TObject);
    procedure btn1Click(Sender: TObject);
    procedure btnSearchClick(Sender: TObject);
    procedure edtSearchKeyPress(Sender: TObject; var Key: Char);
    procedure FormCreate(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    factForm: TfactForm;
implementation
uses mainUnit, settingsUnit, planUnit;
{$R *.dfm}
procedure TfactForm.N1Click(Sender: TObject);
begin
    if grpSearch.Visible = true then
        grpSearch.Visible:= false
    else grpSearch.Visible:= true;
end;
procedure TfactForm.N2Click(Sender: TObject);
begin
    if grpEdit.Visible = true then
        grpEdit.Visible:= false
    else grpEdit.Visible:= true;
end;
procedure TfactForm.sdbgrd1GetCellText(Sender: TObject; Field: TField;
    var Text: String);
begin
    dbcht1.Title.text.Clear;
    dbcht1.title.text.Add('Òîâàð ' + chr(39) + cbb1.Text + chr(39));
end;
procedure TfactForm.edtSearchChange(Sender: TObject);
begin
    {if settingsform.chkfastset.Checked = false then
    begin
        if edtSearch.Text = '' then
            sDBGrd1.DataSource:= ds1;
        end;}
    //if ((btn1.Enabled = false) and (edtsearch.Text = '')) then
    //tbl1.filtered:=false;
    if settingsform.chkfastset.Checked = true then
        begin
            btn1.Enabled:= false;
            if edtSearch.Text <> '' then
                sDBGrd1.DataSource:= ds2
            else sDBGrd1.DataSource:= ds1;
            if rbID.Checked = true then begin
                qry1.Close;
                qry1.SQL.Clear;
                qry1.SQL.Add('SELECT * FROM fact WHERE prodID LIKE ''%' + edtSearch.Text +
                '%''');
                qry1.Open;
            end;
            if rbYear.Checked = true then begin
                qry1.Close;
                qry1.SQL.Clear;
                qry1.SQL.Add('SELECT * FROM fact WHERE prodYear LIKE ''%' + edtSearch.Text +
                '%''');
                qry1.Open;
            end;
        end;
    end;
end;

```

```

    end;
end;
procedure TfactForm.btn1Click(Sender: TObject);
begin
    if settingsform.chkfastset.Checked = false then
    begin
        if rbID.Checked = true then begin
            tbl1.Filter:='prodID=' + edtsearch.Text;
            tbl1.Filtered := True;
        end;
        if rbYear.Checked = true then begin
            tbl1.Filter:='prodYear=' + edtsearch.Text;
            tbl1.Filtered := True;
        end;
    end;
end;
end;
procedure TfactForm.btnSearchClick(Sender: TObject);
begin
    tbl1.Filtered:=false;
    edtsearch.Text:='';
end;
procedure TfactForm.edtSearchKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then begin
        factform.btn1.Click;
    end;
    if key = #27 then begin
        factform.btnSearch.click;
    end;
end;
procedure TfactForm.FormCreate(Sender: TObject);
var i:integer;
begin
    for i:=0 to sdbgrd1.Columns.Count - 1 do
        sdbgrd1.columns[i].Width:=72;
    end;
end.

```

Тексти SQL-запитів

SQL – запит для створення звіту потреби в ресурсах на план випуску продукції

```

SELECT Resources.resName, Sum(Plan!prodJan*CostRate!costRate) AS sumJan,
Sum(Plan!prodFeb*CostRate!costRate) AS sumFeb, Sum(Plan!prodMar*CostRate!costRate) AS sumMar,
Sum(Plan!prodApr*CostRate!costRate) AS sumApr, Sum(Plan!prodApr*CostRate!costRate) AS sumMay,
Sum(Plan!prodMay*CostRate!costRate) AS sumJun, Sum(Plan!prodJun*CostRate!costRate) AS sumJul,
Sum(Plan!prodJul*CostRate!costRate) AS sumAug, Sum(Plan!prodAug*CostRate!costRate) AS sumSep,
Sum(Plan!prodSep*CostRate!costRate) AS sumOct, Sum(Plan!prodOct*CostRate!costRate) AS sumNov,
Sum([Plan!][prodNov]*[CostRate!][costRate]) AS sumDec,
[sumJan]+[sumFeb]+[sumMar]+[sumApr]+[sumMay]+[sumJun]+[sumJul]+[sumAug]+[sumSep]+[sumOct]+[sumNov]+[sumDec] AS sumYear
FROM Resources INNER JOIN ((Assortment INNER JOIN CostRate ON Assortment.prodID = CostRate.prodID) INNER
JOIN Plan ON Assortment.prodID = Plan.prodID) ON Resources.resID = CostRate.resID
GROUP BY Resources.resName;

```