

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,  
обліку та фінансів

Кафедра комп'ютерних технологій  
і моделювання систем

Кваліфікаційна робота  
на правах рукопису

Климець Юрій Андрійович

УДК 004:615.2

## **КВАЛІФІКАЦІЙНА РОБОТА**

Інформаційна підсистема обліку медичних препаратів

Спеціальності 122 “Комп'ютерні науки”

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

---

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Маєвський Олександр Володимирович

Доцент, к.т.н

Житомир – 2023

**Висновок кафедри** \_\_\_\_\_

за результатами попереднього захисту: \_\_\_\_\_

Протокол засідання кафедри \_\_\_\_\_

№ \_\_\_\_\_ від «\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

Завідувач кафедри комп'ютерних технологій і моделювання систем

\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

### **Результати захисту кваліфікаційної роботи**

Здобувач вищої освіти \_\_\_\_\_ захистив (ла)

(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою \_\_\_\_\_

за шкалою ECTS \_\_\_\_\_

за національною шкалою \_\_\_\_\_

Секретар ЕК

\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

## АНОТАЦІЯ

Випускна робота бакалавра представляє систему, яка дозволяє зберігати інформацію про інформаційну підсистему обліку медичних препаратів. Пояснювальна записка до випускної роботи складається з 55 сторінок, 6 ілюстрацій та 10 таблиць.

Інформаційна підсистема обліку медичних препаратів – дипломна робота, яка присвячена створенню програми, яка полегшує роботу для медичних закладів. Основною метою роботи є розробка системи, яка дозволяє деталізувати інформацію про медичні препарати і реалізувати його продаж. Система була реалізована на програмі C# за допомогою Framework .net4.58. В якості управління системами було обрано MySQL. Буде розглянуто основні принципи необхідності впровадження ІТ технологій, пов'язаних з використанням інформаційних технологій в фармацевтичній діяльності. На основі цього аналізу буде розроблено програмне забезпечення, яке полегшує роботу медичним закладам.

Отже, інформаційна підсистема обліку медичних препаратів – являється важливою роботою, яка має певний потенціал для його застосування.

**КЛЮЧОВІ СЛОВА:** C#, ІНФОРМАЦІЯ, ДАНІ, ІНФОРМАЦІЙНА ПІДСИСТЕМА, FRAMEWORK .NET.4.58.

## SUMMARY

Final work of bachelor presents the system that allows to keep information about the informative subsystem of account of medical preparations the Explanatory message to final work consists of 55 pages,6 illustrations and 10 tables

An informative subsystem of account of medical preparations is diploma work, that is sanctified to creation of the program that facilitates work for medical establishments. The primary purpose of work is development of the system, that allows to go into detail information about medical preparations and realize his sale.

The system was realized on the program C# with help Framework .net.4.58

. As a management the systems it was select MySQL.

Basic principles of necessity of introduction of IT of the technologies constrained with the use of information technologies in pharmaceutical activity will be considered. On the basis of this analysis software that facilitates work to medical establishments will be worked out.

Thus, informative subsystem of account of medical preparations is important work that has certain potential for his

**KEYWORDS:** C#, INFORMATION, DATA, INFORMATIVE SUBSYSTEM, FRAMEWORK .NET.4.58

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CRUD – create, read, update, delete (з *англ.* створення, читання, оновлення, видалення)

MVC – Model–View–Controller (з *англ.* Модель–Представлення–Контролер)

SQL – structured query language (з *англ.* мова структурних запитів)

БД – база даних

ОС – операційна система

СУБД – система управління базами даних

## ЗМІСТ

АНОТАЦІЯ.....	3
SUMMARY.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Коротка характеристика об’єкту дослідження.....	11
1.2 Обґрунтування необхідності впровадження ІТ технологій та комп’ютерних засобів.....	13
1.3 Постановка задачі.....	16
2 ОБ’ЄКТНА МОДЕЛЬ СИСТЕМИ.....	20
2.1 Діаграма варіантів використання.....	20
2.2 Діаграма класів.....	24
2.3 Діаграма прецедентів.....	26
2.4 Діаграма діяльності.....	27
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	29
3.1 Інформаційна модель ІС.....	29
3.2 Окремі аспекти програмної реалізації.....	32
4 ТЕСТОВИЙ ЗАПУСК ПРОГРАМИ, ІНСТРУКЦІЯ КОРИСТУВАЧА.....	37
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	44
Додаток 1 Приклад коду взаємодії АІС з БД.....	46
Додаток 2 Лістинг фрагмента коду, який відповідає за додавання замовлення на поставку.....	47
Додаток 3 Головний клас програми ( class F_Main).....	49
Додаток 4 Загальний алгоритм обробки даних.....	55

## ВСТУП

Своєчасне прийняття рішень, що базуються на правильному і продуктивному аналізі інформації, є першочерговим завданням будь-якого керівника. Тому створення і експлуатація інформаційних систем обробки даних в процесах ведення бізнесу будь-якого рівня стає, на даний момент, дуже актуальною задачею.

Інформаційні системи грають важливу роль у фармацевтичній діяльності. В аптечному секторі, де велика кількість даних пов'язана з ліками, замовленнями, інвентарем, клієнтами та іншими аспектами бізнесу, ефективне використання інформаційних систем стає необхідністю. Ось деякі ролі, які вони виконують:

1. **Управління запасами:** Інформаційні системи допомагають вести облік запасів ліків і медичних продуктів. Вони автоматично оновлюють запаси, виконують замовлення і контролюють терміни придатності. Це допомагає уникнути дефіциту або перевищення запасів, забезпечує точний облік та зменшує витрати.

2. **Обробка рецептів:** Інформаційні системи дозволяють фармацевтам обробляти рецепти швидко і ефективно. Вони зберігають інформацію про рецепти, дозування, взаємодії між ліками та інші релевантні дані. Це допомагає уникнути помилок при видачі ліків і підвищує безпеку пацієнтів.

3. **Керування клієнтською базою:** Інформаційні системи дозволяють зберігати дані про клієнтів, їхній медичний історії, рецепти та іншу інформацію. Це допомагає фармацевтам забезпечувати індивідуальний підхід до кожного клієнта, надавати рекомендації та підтримувати довгострокові відносини.

4. **Взаємодія з постачальниками:** Інформаційні системи дозволяють спілкуватися з постачальниками, оформлювати замовлення, контролювати оббіг.

Тематиці застосування ІТ в фармацевтичній діяльності присвячено рід статей та наукових публікацій, як зарубіжних авторів так і вітчизняних. Коротко оглянемо деякі з них.

Так стаття [1] "Digital transformation and disruption of the pharmaceutical and healthcare industries" (2015) авторів Palvia, P., Aeron, P., Gupta, P., Mahapatra, D., & Parida, R., досліджує тему цифрової трансформації і порушень у фармацевтичній сфері та охороні здоров'я. У статті розглядаються наслідки технологічного прогресу і впровадження інформаційних технологій у фармацевтичну та медичну галузі. Автори досліджують вплив цифрової трансформації на бізнес-моделі, процеси, способи роботи і взаємодію між різними зацікавленими сторонами в цих індустріях.

Статті [2,3] зосереджуються на таких аспектах, як впровадження електронних медичних записів, використання аналітики даних, інтернету речей (IoT), штучного інтелекту (AI) та інших інформаційних технологій у фармацевтичній сфері та сфері охорони здоров'я.

Стаття [6] досліджує роль інформаційних систем у підтримці управління знаннями на прикладі фармацевтичної компанії у Йорданії. Стаття [7] розглядає роль електронного бізнесу у покращенні інтеграції ланцюга постачання в фармацевтичній галузі. В статті[8] розглядаються етичні та правові аспекти фармацевтичного маркетингу. Українські видання та публікації за останні роки досить скоротились в силу відомих причин та військових подій в країні. Хоча деякі з публікацій на ведено в переліку джерел [9-12] і були проаналізовані в ході виконання бакалаврської роботи.

Дослідження та література підтверджують, що ІТ мають значний вплив на фармацевтичну індустрію і можуть поліпшити якість надання медичних послуг, ефективність процесів та безпеку пацієнтів. Ось деякі основні області використання ІТ в фармацевтиці:

1. Електронні медичні записи (ЕМР): Впровадження ЕМР дозволяє фармацевтам електронно зберігати та обробляти медичну інформацію пацієнтів, включаючи рецепти, алергії, історію лікування та результати



аналізів. Це поліпшує доступність та обмін інформацією між медичними працівниками та допомагає уникнути помилок.

2. Аптечні інформаційні системи: Спеціалізовані аптечні інформаційні системи дозволяють фармацевтам ефективно керувати запасами ліків, виконувати замовлення, вести облік рецептів та контролювати взаємодії між ліками. Вони автоматизують багато рутинних процесів, поліпшуючи ефективність та точність.

3. Електронний обмін даними: Використання електронного обміну даними дозволяє фармацевтам спілкуватися з медичними системами, постачальниками та іншими сторонами шляхом передачі стандартизованих даних

Система, що досліджується в роботі, відноситься до систему обліку ТМЦ (товарно-матеріальних цінностей). Одна з основних цілей систем обробки даних полягає у підвищенні ефективності роботи компанії, установи або організації. Система обробки даних повинна:

- забезпечувати отримання загальних або деталізованих даних за підсумками роботи,
- визначати тенденції зміни найважливіших показників,
- забезпечувати отримання інформації, виконувати точний і повний аналіз даних,
- забезпечувати найбільш ефективну форму надання кінцевих результатів,
- використовувати оптимальні методи архівування та каталогізації даних.

**Об'єктом** дослідження роботи є склад фармацевтичної дистриб'юторської компанії «Farmaco+», що спеціалізується на оптовому продажу ліків, та постачанню ліків в роздрібні аптеки. А також бізнес процеси, що виникають під час здійснення підприємницької діяльності.

**Предметом** дослідження є інформаційні технології та засоби автоматизації процесів обліку товарів і продажів, інформаційна підтримка реалізації бізнесу.

**Актуальність** роботи - облік медичних препаратів з метою своєчасної закупівлі та усунення ризику їх дефіцитів.

Звертаючи увагу на кількість товарних позицій і обсяги товарообороту складу, виникає проблема зручного обліку та систематизації продукції. На даний момент досліджувана організація не має спеціалізованого програмного комплексу, який здатний вирішити і автоматизувати більшість організаційних питань по зберіганню та обліку продукції. Актуальність визначається практичною спрямованістю дослідження і програмної реалізації в ході роботи

**Метою** роботи даного проекту є створення програмного комплексу, який автоматизує облік товару складу.

**Задачами** роботи є:

1. Аналіз бізнес-процесів компанії, виділення процесів для автоматизації;
2. Розробка об'єктної моделі;
3. Розробка інформаційної моделі;
4. Опис вхідних, нормативно-довідкової і результативної інформації;
5. Фізична реалізація БД в обраному форматі;
6. Реалізація програмної автоматизованої системи для управління базою даних: ведення довідників, фільтрація даних, облік надходжень і витрат товару, формування звітів;
7. Опис програмних компонентів системи.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Коротка характеристика об'єкту дослідження

Областю функціонування розроблюваної системи є автоматизація управління товарооборотом складу дистриб'ютора лікарських засобів та медичних препаратів «Farmaco+». Компанія умовна, і фіксується для конкретизації компанії-замовника. Загалом досліджуванні бізнес-процеси не залежать від конкретної компанії, і, в більшості, відрізняються лише об'ємами товарообігу та рівнем глобальності покриття.

Організація працює за простою схемою: закупка товарів великим гуртом (оптом) безпосередньо у виробників або великих компаній-дилерів, що імпортують лікарські засоби закордонного виробництва. З метою подальшого продажу товарів мілко гуртовими партіями для подальшого роздрібного продажу з моржою в середньому 25-30%

Організаційна структура торгового підприємства складається з директора, та підпорядкованих йому співробітників і завідувачів відділами.

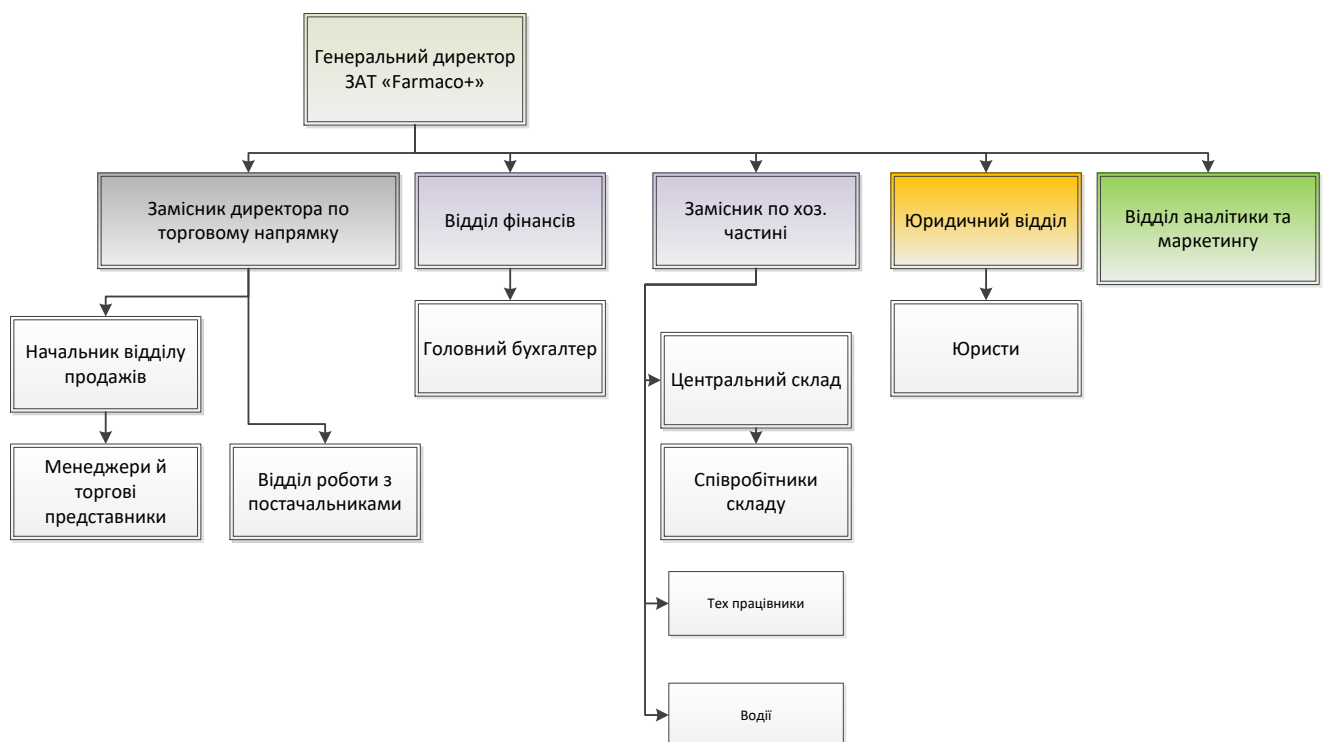


Рисунок 1.1 - Організаційна структура складу «Farmaco+»

Генеральний директор, керує всіма відділами і видами діяльності компанії. Спрямовує діяльність філій та інших підрозділів підприємства для досягнення високих темпів розвитку і зростання продуктивності праці. Він також видає накази і розпорядження, несе відповідальність в межах своїх повноважень за діяльність підприємства, забезпечення збереження товарно-матеріальних цінностей, грошових коштів та іншого майна підприємства.

Заступник директора з торговельного напрямку курує продавців на всіх торгових точках, в його підпорядкуванні також знаходиться відділ роботи з постачальниками.

Юридичний відділ займається оформленням договорів з компаніями.

Бухгалтерія виконує роботу з бухгалтерського обліку (облік матеріальних цінностей, витрат на виробництво, розрахунки з постачальниками і замовниками, розрахунок заробітної плати, оподаткування, контроль за процесом реалізацією продукції).

Заступник директора по госп. частини відповідає за технічне забезпечення і стан матеріальних активів організації. У зону його управління входять: технічні працівники (слюсар, прибиральниці), водії (доставка та інші потреби організації), склад, його кадрове та технічне забезпечення.

Склад здійснює прийом і відвантаження товару клієнтам компанії. Також склад виконує наступні функції:

1. забезпечення виконання зобов'язань по поставках продукції і товарів відповідно до укладених договорів;
2. ревізія залишків товарів;
3. підготовка звіту, що є підставою для замовлення товару у постачальників.

## 1.2 Обґрунтування необхідності впровадження ІТ технологій та комп'ютерних засобів

Аналіз дає можливість обрати декілька основних напрямків, пов'язаних з використанням інформаційних технологій в фармацевтичній діяльності. А саме це:

1. Електронні медичні записи (ЕМЗ): Дослідження можуть висвітлювати вплив інформаційних технологій на впровадження ЕМЗ у фармацевтичних закладах. Аналіз переваг, викликів і результатів використання ЕМЗ, таких як покращення точності і доступності медичної інформації, зменшення помилок та підвищення безпеки пацієнтів.

2. Автоматизовані системи управління аптекою: застосування інформаційних технологій для автоматизації різних аспектів аптечної діяльності, включаючи управління запасами, видачу рецептів, контроль за придатністю ліків та інше. Аналіз може виявити ефективність таких систем, їх вплив на продуктивність та економічні показники аптек.

3. Електронна комерція в фармацевтичній галузі: роль інформаційних технологій у розвитку електронної комерції в аптечному секторі. Використання онлайн-платформ для замовлення ліків та медичних продуктів, впровадження електронних систем платежів та досліджувати переваги та виклики, пов'язані з цим процесом.

4. Захист персональних даних та кібербезпека: важливість захисту персональних даних пацієнтів та інших конфіденційних інформаційних ресурсів у фармацевтичній галузі. Він може аналізувати заходи кібербезпеки, політики та стандарти, необхідні для запобігання порушенням та зловживанням.

Тематика роботи з одного боку конкретизує напрямок саме в області обліку медикаментів для аптечних закладів, а з іншого, навпаки – узагальнює предмет дослідження до загалом систем обліку. Бо, хоча облік медикаментів і має свої певні особливості, загалом можна розглядати медикаменти як товар,

а систему, що досліджується, як представника систем для обліку товарів на складі. Тому для узагальнення надалі будемо більш абстрактно розглядати задачу, а медикаменти в загальному називати «товаром».

Даний напрямок (облік товарів) є одним з ключових в роботі компанії. З точки зору економічного аспекту, організація ефективної роботи персоналу потрібна не тільки для поліпшення структури роботи всередині компанії, а й для більш швидкої і надійної роботи з клієнтами.

В даний час автоматизація управління торгівельною мережею та товарообігом на даному підприємстві зведена до мінімальних, основних задач. Сортування новоприбулого товару відбувається ручним способом, дані при цьому записуються в робочих зошитах, файлах Excel. В результаті такої постановки питання будь-яка спроба аналізу або коригування місць зберігання товарів відбирає багато часу і людино-годин. Так само варто відзначити, що при такому способі зберігання інформації про наявну на складі продукції, досить складно здійснювати за нею контроль, при відправці різним клієнтам. При складанні документації на замовлення нового товару так само виникають деякі складнощі, так як потрібно вручну перевірити записи існуючих одиниць на складі, а так же підготувати бланки, що при великих обсягах замовляються одиницях дуже трудомісткий процес.

Підприємства, що стоять перед вибором, яку інформаційну систему придбати, повинні добре розуміти функціональні відмінності. Загалом, системи управління пропонують великі функціональні можливості для управління товарооборотом, логістикою, системи управління і взаємодії з клієнтами підприємства. Але використання повнофункціональних систем має і свої недоліки: суттєві вимоги до апаратної частини, висока ціна програмних продуктів, необхідність стажування та навчання персоналу.

З огляду на останнє варто вибрати найбільш вигідний баланс, правильно обрати функції та напрямки, що підлягають автоматизації.

Після впровадження системи основний процес змінюється кардинальним чином. Всі вантажні одиниці, що надійшли на склад, заносяться

в програму і маркуються індивідуальним кодом, який містить інформацію про характер вантажу, коли і від кого він прийшов. Якщо вантажні одиниці, що надійшли, вже мали індивідуальне маркування, програма автоматично реєструє їх і додає кількість прибулого товару до існуючого. Так само спрощується ведення економічного обліку всередині магазину. Всі розрахунки, пов'язані з касової звітністю, а так само витратами на придбання нового товару і іншими категоріями складаються автоматично, після чого в готовому вигляді надаються користувачеві.

Практика показує, що загальна продуктивність зростає мінімум на 20 - 30%, точність обліку - більш ніж на 99%, трудовитрати зменшуються на 25% і більше.

Програмний комплекс повинен приймати інформацію, пов'язану з отриманим товаром на склад (кількість, постачальник, найменування і тд), визначати кількість місць в комірках на складі, для можливості їх заповнення. Так само даний комплекс повинен приймати всі замовлення на товар покупцями, здійснювати економічні розрахунки і виводити певні звіти по деяким з категорій.

Від того, наскільки раціонально буде спроектована реалізація завдання в розробляється програмному комплексі, настільки гарантовано буде зниження вартісних, трудових витрат.

Технологічний процес, як правило, складається з декількох етапів. Метою першого етапу є збір, реєстрація, передача даних для подальшої обробки. Результатом зазвичай є складання документа. Мета другого етапу - перенесення даних на машинні носії і початкове формування інформаційної бази. Третій етап включає операції накопичення, сортування, коригування та обробки даних.

При виборі варіанту технологічного процесу потрібно враховувати наступні вимоги:

1. забезпечення достовірності оброблюваної інформації;
2. рішення задач у встановлений термін;

3. забезпечення мінімальних трудових і вартісних витрат на обробку даних;
4. наявність можливості обробки даних на ЕОМ;
5. можливість вирішення завдання в різних режимах.
6. У процесі технічної підготовки вирішуються такі основні завдання:
  7. розробка технологічних процесів;
  8. технологічне оснащення;
  9. технічний супровід.

**Вихідними даними є:**

1. максимальний річний обсяг продажів;
2. планований режим роботи (кількість змін, тривалість робочого тижня);
3. плановані поставки матеріалів/продукції і підприємства-постачальники;
4. інвентаризація наявних товарів на складі та їх місце розташування.

### **1.3 Постановка задачі**

Завданням роботи є дослідження та аналіз процесів, що відбуваються на підприємстві ЗАТ «Farmaco+» в області обліку товарів та моніторингу залишків та розробка програмного комплексу для автоматизації даного напрямку.

В ході роботи необхідно розробити універсальний програмний комплекс для автоматизації роботи складу та обліку медичних препаратів на складі компанії. Програмний комплекс повинен окремо і зручно відображати наявність на складах актуального кількості товару та/або матеріалів. Програмний комплекс повинен приймати замовлення на отримання товару в формі бланка і визначати - чи достатньо на складах необхідного кількості товару для виконання замовлення. Програма повинна мати можливість окремо



відображати замовлення по кожному окремому клієнту. Також є необхідність формувати статистичні дані по товарах магазину. Програмний комплекс повинен мати зручний інтерфейс.

Узагальнивши всі вимоги можна сказати що програма повинна відповідати наступним вимогам:

- Працювати під управлінням ОС Windows;
- Мати дружній, простий у використанні інтерфейс;
- Окремо відображати дані по товару;
- Відображати замовлення щодо кожного замовника;
- Приймати бланки замовлень на поставку продукції на склад, зберігати і видавати інформацію по цих замовленнях;
- Автоматично формувати звіти;
- Мати базу даних для зберігання записів з даними щодо кількості товару на складі, бракований товар, замовленнями і постачальниками.

Після впровадження розробленої системи методика роботи складу зміниться в кращу сторону. Зміни стосуватимуться методики зберігання інформації пов'язаної з інформацією стосується товарообігу магазину, а також фінансової інформації. Система автоматизує багато факторів обліку складського приміщення, обліку додаткової інформації по кожному товару. Власне система зможе грати роль інформаційного довідника по кожному з одиниць товару наявного на складі.

Після впровадження співробітник/адміністратор системи складу звільниться від необхідності ручної каталогізації та обліку товару, що зберігається, або ж новоприбулого на склад. У програму потрібно лише вчасно вносити необхідні дані, а весь процес зберігання, розподілу інформації, створення і обробку звітів за різними характеристиками, вона візьме на себе.

Автоматизація даної діяльності має на увазі рішення наступних завдань:

- Автоматичне сортування даних
- Безпека зберігання даних

- Неможливість редагування і перемішування інформації
- Формування та виведення звітності

На рисунку 1.2 представлена схема, що характеризує даний процес «Як буде (TO-BE)» з використанням розробленої АІС.

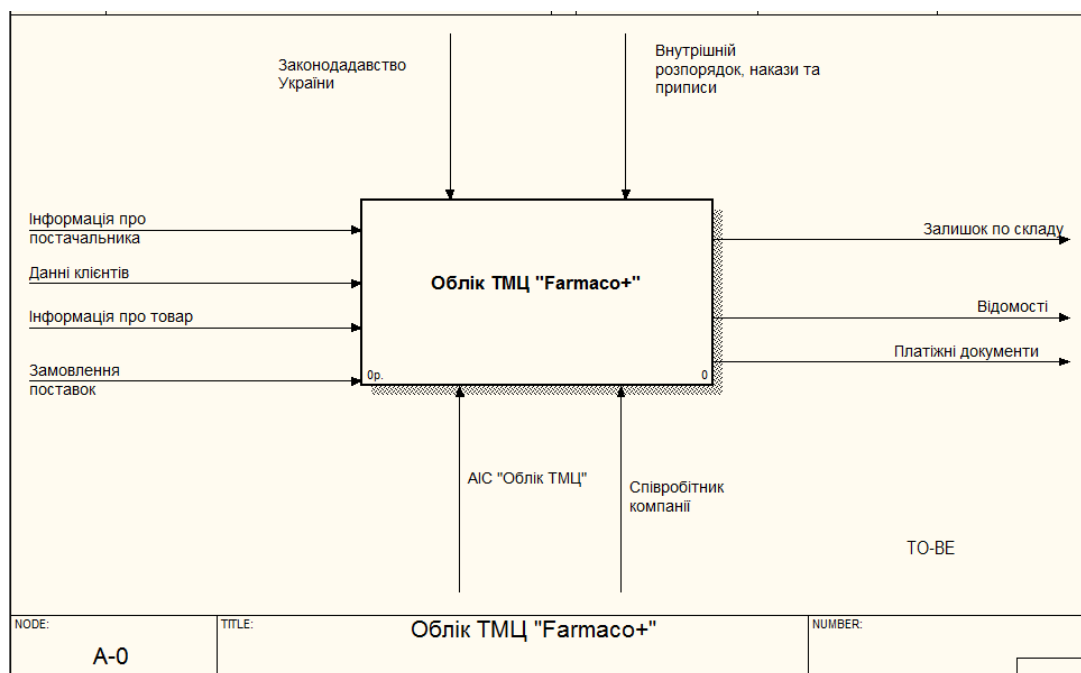


Рисунок 1.2 - Контекстна діаграма характеристики діяльності після впровадження розроблюваної системи

Як видно з діаграми процес обліку істотно спроститься, у всіх функціях співробітник бере участь тільки як оператор, а обробкою інформації займається ІС. Внесення даних по постачальникам і товарів виконується одноразово, з урахуванням того, що набір постачальників досить статичний - менеджеру (продавцю) не доведеться щоразу вносити інформацію заново, досить вибрати потрібного постачальника з БД. Оновлювати або вносити нові дані потрібно тільки в разі появи нового постачальника або нової групи товарів. Найголовнішою перевагою впровадження системи є централізований спосіб зберігання даних, що дозволяє володіти оперативною інформацією про наявність товару відразу всім користувачам. Іншою істотною перевагою є автоматизація підготовки звітностей і швидкого пошуку інформації.

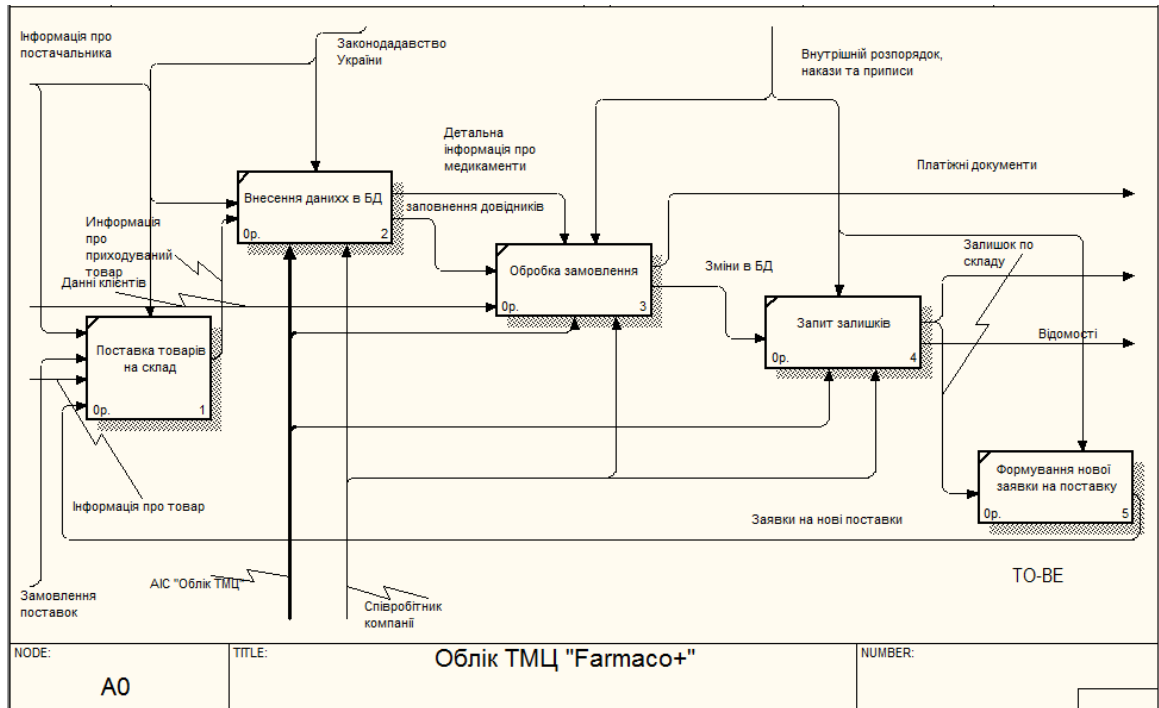


Рисунок 1.3 - Діаграма першого рівня декомпозиції характеристики діяльності після впровадження розроблюваної системи, декомпозиція

## 2 ОБ'ЄКТНА МОДЕЛЬ СИСТЕМИ

Зазвичай розробка великих проектів розпочинається з моделювання. Типи моделей можуть бути абсолютно різними, один з популярний типів моделей – об'єктне моделювання UML

### 2.1 Діаграма варіантів використання

Робота над моделлю UML в середовищі IBM Rational Rose починається з загального аналізу проблеми і побудови діаграми варіантів використання, яка відображає функціональне призначення проєктованої програмної системи. Зазвичай ця діаграма **НЕ деталізує** систему, а тільки описує основний можливий функціонал - показуючи доступні способи дій в системі. Варіант використання описує, з точки зору діючої особи, групу дій в системі, які призводять до конкретного результату.

Варіанти використання є описами типових взаємодій між користувачами системи і самою системою. Вони відображають зовнішній інтерфейс системи і вказують форму того, що система повинна зробити (саме що, а не як).

Побудова нашої моделі ми теж почнемо з діаграми варіантів використання.

Основним актором системи є оператор АІС, людина яка займається безпосередньою роботою з системою. Варіанти використання в даному випадку ілюструють функціонал системи (рис. 2.1).

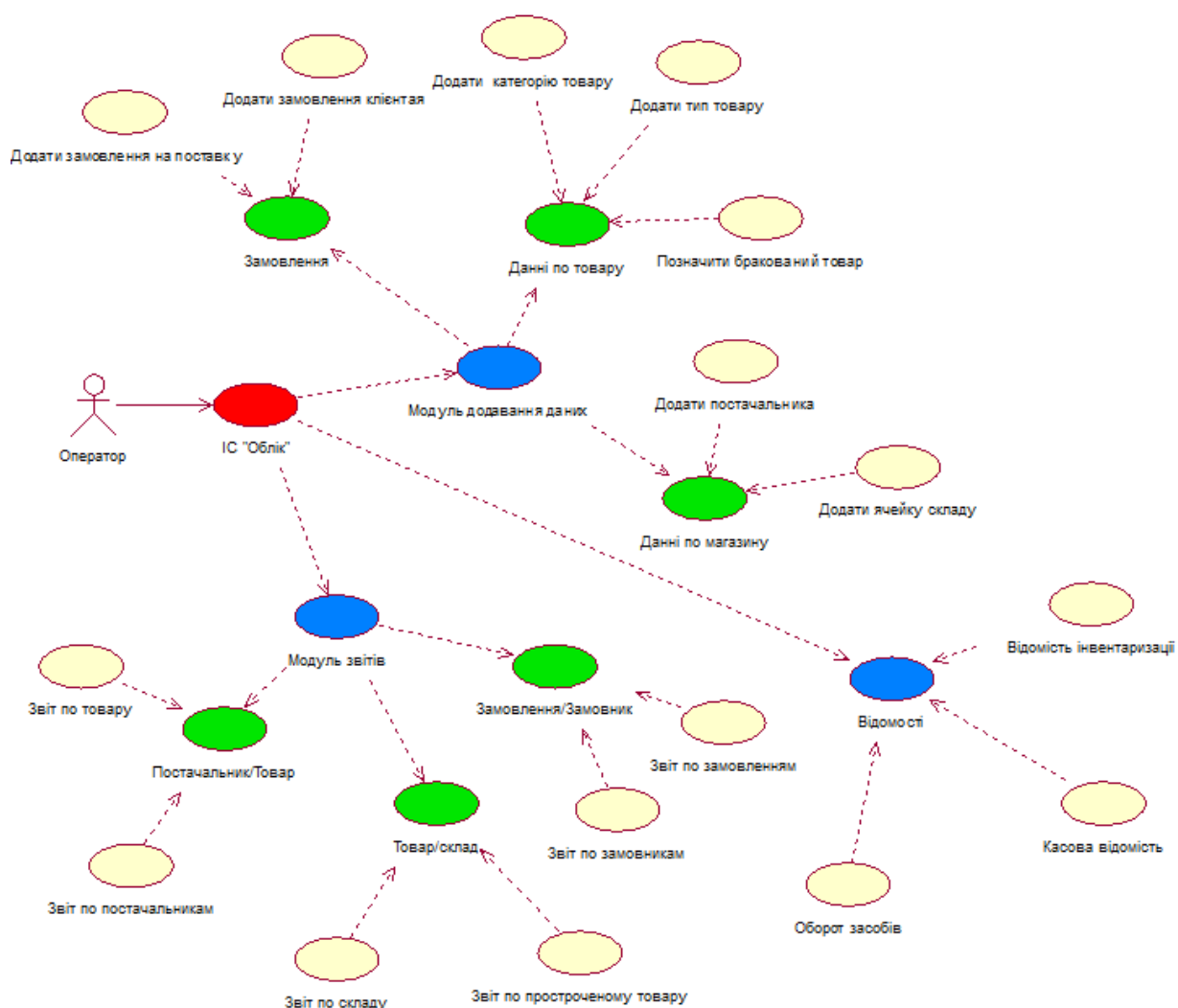


Рисунок 2.1 – Діаграма варіантів використання

Опосередковано з системою може працювати клієнт і директор, але їх взаємодії здійснюються через оператора, що і показано на діаграмі. З іншого боку, торгова компанія не може обійтися без постачальників - безпосередніх виробників або імпортерів товарів, які є предметом торгівлі. Постачальник виконує поставку товару на склад, цей процес протоколюється в системі - товар вноситься в базу даних після надходження на склад. Окремим прецедентом є розрахунки між торговою фірмою і постачальниками, хоча на схемі вони не показані. Таке спрощення вважаємо допустимим - так як основними функціями системи є облік товарів, з іншого боку зовсім не згадати цей прецедент теж неправильно - так як «розрахунок» є невід'ємною ланкою

торговельних відносин. Тим більше в системі має фіксуватись ціна закупівлі товару.

Опис прецедентів представимо в таблиці нижче, вона деталізує дії на рівні роздрібного продажу.

Весь периферійний рівень деталізує функції системи, всі дії мають тип зв'язку «розширення», вони не вказані на схемі для мінімізації кількості елементів з метою підвищення ступеня сприйняття та меншого нагромадження.

Таблиця 1 - Характеристики прецедентів системи

Назва	Передумови	Головний потік	Альтернативний потік (виключення)	Підпотоки
<b>Прецеденти Actor Оператор АІС</b>				
<b>Робота з АІС «Облік ТМЦ»</b>	ПК повинен знаходитися у включеному стані, Є зв'язок з сервером	Прецедент починає виконуватися, коли оператор входить в систему використовуючи комбінацію логін/пароль	Якщо ПК оператора не справний чи ні з'єднання з сервером - прецедент не виконується	Підключення до БД
<b>Додати дані</b>	Існує новий товар/постачальник/замовлення Виконано умови попереднього прецеденту	За допомогою інтерфейсу АІС оператор звертається до модуля роботи з БД. Заповнивши екранні форми інформація додається в БД	Оператор очікує появи нових даних, може проводити операції з існуючими	Виконання транзакцій з БД
<b>Побудувати відомість</b>	Умови першого прецеденту + введена коректна інформація для відомості	На екрані відображаються відомість обраного типу за введений період часу. Система дає можливість відправити відомість на друк або зберегти у зовнішній файл	В разі некоректних даних система дає можливість змінити вхідні дані	
<b>Прецеденти Actor Клієнт</b>				
<b>Зробити замовлення</b>	У клієнта є бажання і можливості купити товар. Склад працює, система справна. Товар бажаний клієнтом існує на складі в потрібному обсязі	За допомогою вимог озвучених Клієнтом з використанням системи за участю Оператора формується замовлення	Варіант 1. Не задоволені передумови прецеденту: магазин працює, система справна - Клієнт вибирає інший магазин Варіант 2 Немає бажаного товару - зміна в складі замовлення або варіант 1	Отримання накладної, отримання чека, стягування плати, видача здачі
<b>Прецеденти Actor Директор / Адміністратор</b>				
<b>Отримати звіт</b>	Умови першого прецеденту актора Оператор, необхідність в звіті для адміністрації, коректні дані	У списку буде відображено звіт - вибірка з БД (результат запиту) обраного типу по введених даних. Система дає можливість відправити звіт на друк або зберегти у зовнішній файл	В разі некоректних даних система дає можливість змінити вхідні дані. Звіт може бути порожнім	

## 2.2 Діаграма класів

Діаграма класів визначає типи класів системи і різного роду статичні зв'язки, які існують між ними. Діаграма класів складається з множини елементів, які в сукупності відображають декларативні знання про предметну область. На діаграмах класів зображуються також атрибути класів, операції класів та обмеження, які накладаються на зв'язку між класами. З точки зору розробки це одна з найважливіших діаграм моделі.

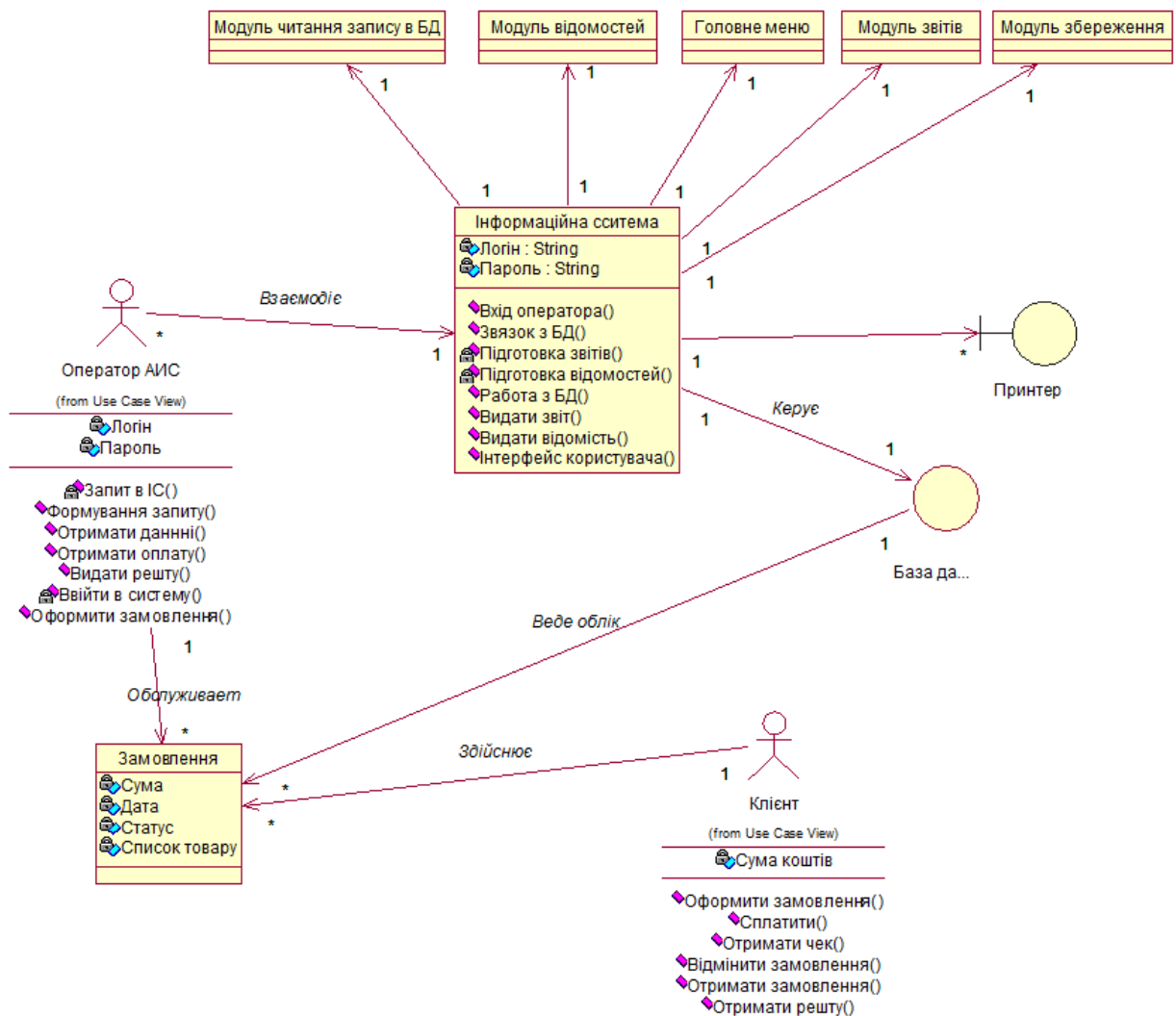


Рисунок 2.2 – Діаграма варіантів використання

Діаграма більш орієнтована на класи, що будуть реалізовані в програмі, в той час як деякі класи будуть представлятись як інформаційні сутності. Деякі



класи не представлені на схемі, але мають бути згадані, наприклад - «Накладна».

Клас «Накладна» описує об'єкт «Накладна», має атрибути:

- Список товарів - перелік товарів замовлення;
- Сума до оплати - загальна вартість замовлення (сума цін товарів замовлення);
- Тип «видаткова» для клієнта, «прибуткова» для постачальника;
- Номер - для фіксації операції в системі.

«Накладна» призначена для документування операцій в ході торгових відносин.

Клас «Товар» описує об'єкт «Товар», має атрибути:

- Назва товару;
- Ціна за одиницю (порцію);
- Термін придатності;

Призначення полів однозначно трактується по їх назвам.

Клас «Замовлення» описує об'єкт «Замовлення», має атрибути:

- Список товарів - перелік товарів замовлення;
- Сума до оплати - загальна вартість замовлення (сума цін товарів замовлення);
- Дата - визначає дату на яку потрібно виконати поставку замовлення для клієнта;
- Статус - визначає стан замовлення, може приймати значення:

«Потенційний» бажання заказчика придбати товар;

«Оформлений» замовлення зареєстрований в системі;

«Скасований» клієнт скасував замовлення;

«Оплачений» клієнт оплатив замовлення за накладною;

«Виданий» товари видані зі складу та отримані клієнтом.

Класи можна представити у вигляді CRC-card/ Класи «Накладна», «Товар», «Замовлення» взагалі не мають методів, а лише властивості (поля). Тому для ілюстрації виберемо класи: «Оператор», «Постачальник», «Клієнт»

<b>Оператор</b>	<i>Взаємодіє</i>
Оформити накладну	Накладна
Оприходувати/росходувати товар	Товар
Сформувати замовлення	Замовлення
Сформувати відомість	Відомість

<b>Постачальник</b>	<i>Взаємодіє</i>
Отримати замовлення	Товар
Виконати поставку	Накладна
Отримати оплату	

<b>Клієнт</b>	<i>Взаємодіє</i>
Оформити замовлення	Накладна
Отримати товар	Товар
Оплатити замовлення	
Отримати чек	

### 2.3 Діаграма прецедентів

Загалом модель може мати 9 типів основних діаграм, при цьому окремі дії(прецеденти) можуть мати по декілька різноманітних діаграм деталізації. Наприклад - послідовність виконання може бути проілюстрована діаграмами послідовності виконання або діаграмами кооперації. Об'єм роботи не дозволяє представляти та описувати модель детально, тому зупинимось в цьому моменті на діаграмі кооперації.

Ця діаграма в деякому сенсі дублює діаграму послідовності виконання. Але її основною метою є показати задіяні компоненти в ході виконання тієї чи іншої операції в системі. Для ілюстрації цього моменту виконаємо діаграму

«оформлення замовлення», але вже у вигляді коопераційної діаграми (collaboration).

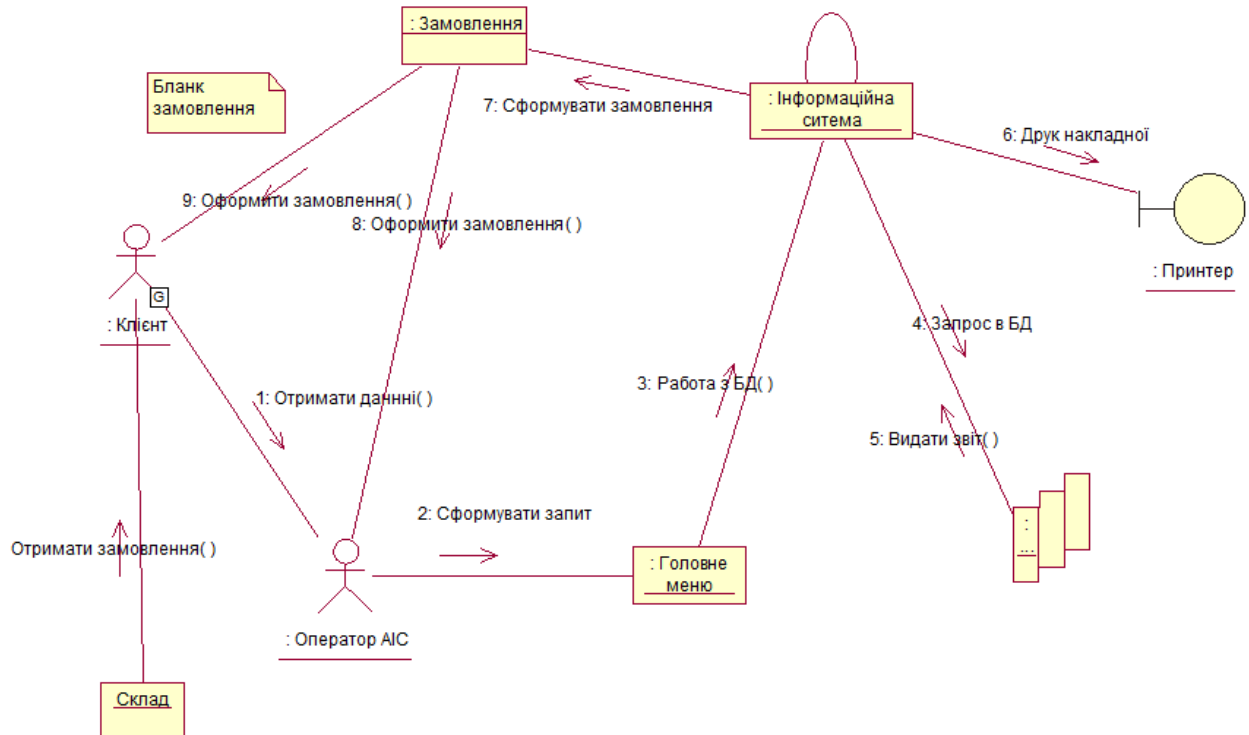


Рисунок 2.3 – Діаграма кооперації «Оформлення замовлення»

## 2.4 Діаграма діяльності

Діаграма діяльності найбільше походить на класичну блок-схему алгоритму. Основним елементом є діяльність. Інтерпретація цього терміна залежить від тієї точки зору, з якої будується дана діаграма. На концептуальній діаграмі діяльність - це деяка задача, яку необхідно виконати вручну або автоматизованим способом.

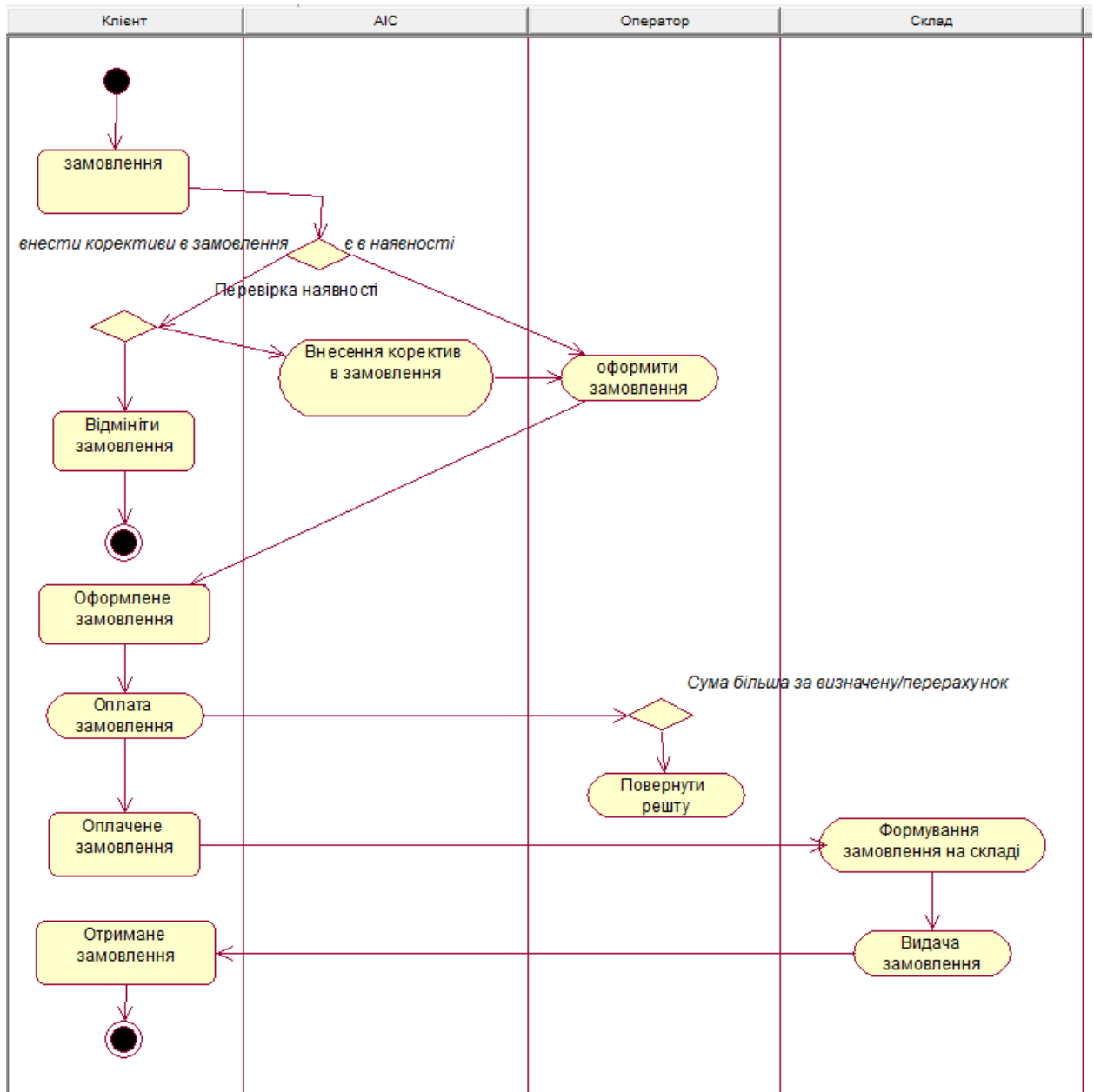


Рисунок 2.3 – Діаграма діяльності «Оформлення замовлення/стани замовлення»

Наявність на діаграмі «доріжок» дозволяє відразу зрозуміти - яка частина системи відповідає за виконання дії. Діаграма може мати багато альтернативних закінчень. У нашому випадку їх два: «скасоване замовлення», «отримане замовлення».

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

В попередньому розділі розроблена об'єктна модель системи, яка, зокрема включає й діаграму класів. Але по своїй суті інформаційна система являє собою СУБД. Використання саме бази даних дає можливість ефективної реалізації функцій та компактного збереження даних. Тому реалізація системи в своїй основі буде містити базу даних.

Окремі класи для представлення сутностей в програмній реалізації не мають особливого сенсу. Об'єкти будуть реалізовані опосередковано: поля класів реалізуються в сутностях бази даних (в таблицях), а методи реалізуються як запити.

### 3.1 Інформаційна модель ІС

Інформаційне забезпечення являє собою найважливіший елемент автоматизованих інформаційних систем.

Діаграма «сутність-зв'язок» являє собою модель даних верхнього рівня. Вона включає суті і взаємозв'язку, що відображають основні бізнес-правила предметної області.

Така діаграма не дуже деталізована, в неї включаються основні сутності і зв'язку між ними, які задовольняють основним вимогам, що пред'являються до інформаційної системи.

Для створення діаграми сутність зв'язок слід описати суті, атрибути, основні первинні ключі та встановити зв'язки між сутностями.

Для розроблюваної інформаційної системи сутностями логічної моделі даних є: «Бланк відправки», «Брак», «Бланк замовлення», «Категорія товару», «Категорія», «Тип товару», «Постачальник», «Склад», «Товар на складі ».

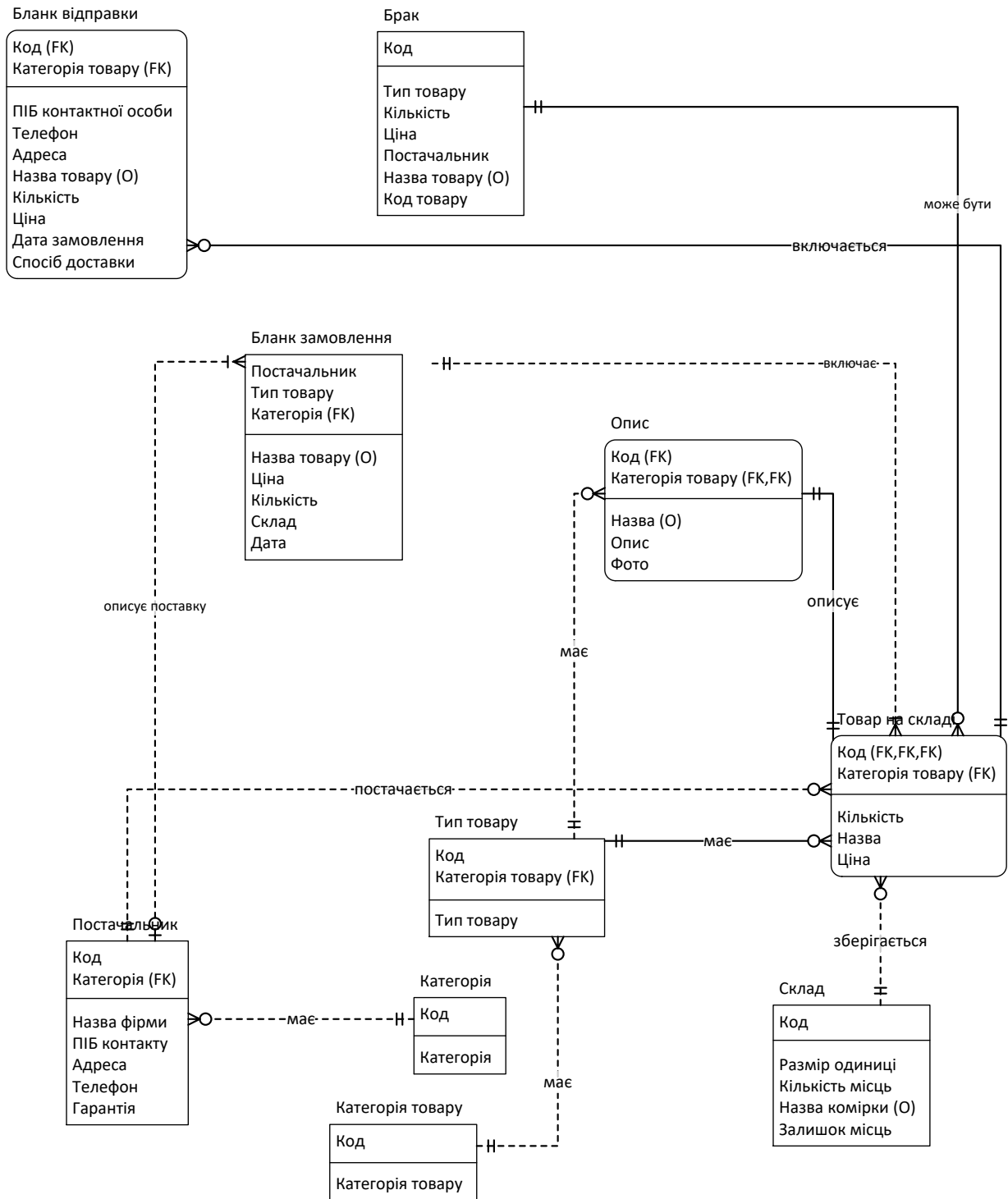


Рисунок 3.1 – ER модель бази даних

В БД переважають зв'язки типу «1 до багатьох», найяскравішим і зрозумілим прикладом тут буде зв'язок між сутностями «Постачальник» «Товар»: одному постачальнику відповідає багато товарів на складі (весь асортимент, що постачається даними постачальником). Присутні і зв'язки «1 до 1», наприклад: сутності «Товар» - «Опис».

База даних як описано вище складається з 10 таблиць (назви таблиць та їх полів дають однозначне представлення про призначення таблиці в системі), і забезпечує інформаційну частину системи, тобто, всі дані якими система оперує, зберігається безпосередньо там. БД спроектована відповідно до моделі даних розглянутої раніше.

Представимо структуру деяких сутностей, що в фізичній реалізації БД будуть перетворені на таблиці.

Таблиця 3.1 – Сутність «Товар»

Фізичне ім'я	Тип даних	Обов'язкове	PK
Код	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Категорія товару	BYTE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Кількість	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Назва	CHAR(50)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ціна	DECIMAL(10;2)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶		<input type="checkbox"/>	<input type="checkbox"/>

Таблиця 3.2 – Сутність «Бланк відправки»

Фізичне ім'я	Тип даних	Обов'язкове	PK
Код	LONG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ПІБ контактної особи	CHAR(60)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Телефон	CHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Адреса	CHAR(120)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Категорія товару	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Назва товару	CHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>
Кількість	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ціна	DECIMAL(10;2)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Дата замовлення	DATETIME	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Спосіб доставки	BYTE	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Таблиця 3.3 – Сутність «Постачальник»

Фізичне ім'я	Тип даних	Обов'язкове	PK
Код	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Назва фірми	CHAR(40)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ПІБ контакту	CHAR(50)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Адреса	CHAR(120)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Телефон	CHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Категорія	BYTE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Гарантія	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Далі розглянемо окремі аспекти програмної реалізації автоматизованої програмної системи обліку медичних препаратів (облік ТМЦ)

### **3.2 Окремі аспекти програмної реалізації**

#### **Дерево функцій і сценарій діалогу**

Для автоматизації розроблюваного програмного продукту існують функції управління і обробки даних. При цьому можна виділити і деталізувати дві підмножини функцій:

- службові функції (наприклад, перевірка коректності вводу даних, архівації баз даних та ін.)
- основні функції (введення первинної інформації, обробка, запити до БД).

Для розробки структури сценарію діалогу крім виявлення складу функцій і їх ієрархії необхідно вибрати мову спілкування. Меню як тип діалогу дуже зручний для кінцевого користувача. Реалізація діалогу типу «меню» можлива через виведення на екран відеотермінала певних функцій системи.

Вибір конкретної функції користувачем може здійснюватися:

1. Набором на клавіатурі необхідної директиви або її скороченого позначення
2. Набором на клавіатурі номера необхідної функції
3. Наведенням курсору на рядок екрана з потрібною користувачеві функцією
4. Натисканням функціональних клавіш, запрограмованих на реалізацію даної функції

При наявності різних варіантів відповіді на введення функції користувачем в наступних кроках відбувається деталізація, або уточнення дій, наприклад, яка інформація повинна вводитися, в якому вигляді або на який пристрій бажано здійснити виведення і т.д.



У зв'язку з тим, що при вирішенні завдання використовується технологія обробки інформації в режимі діалогу, взаємодія користувача з програмою можна представити у вигляді схеми діалогу (рис. 3.2).

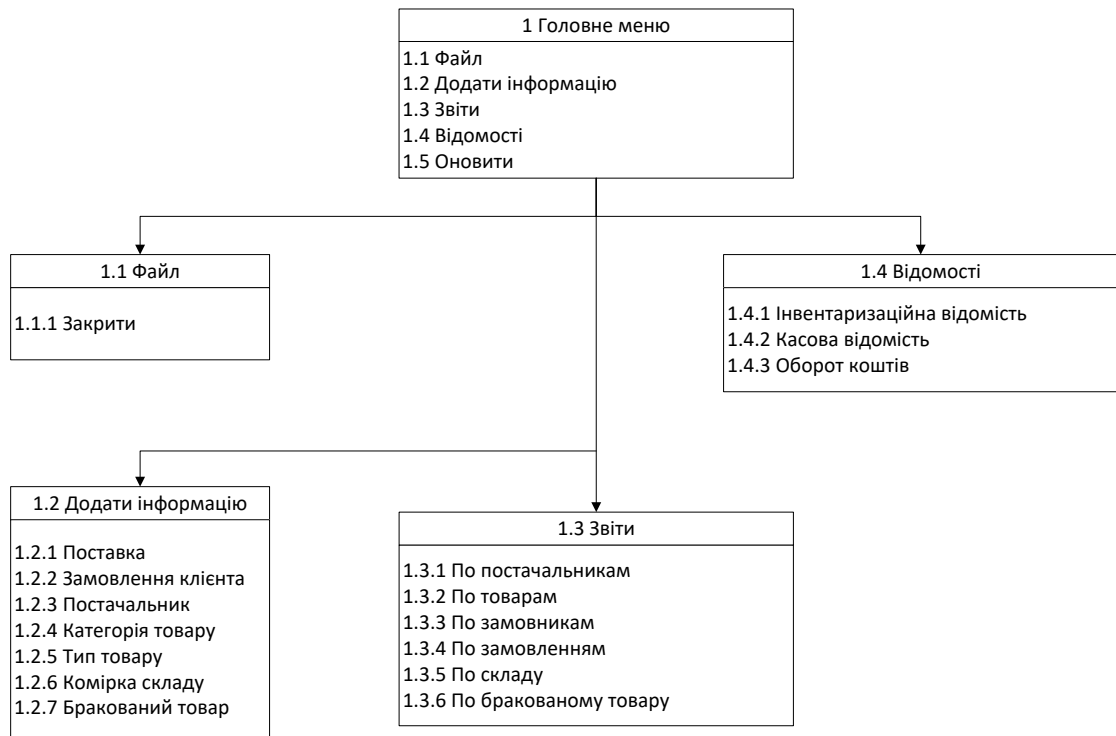


Рисунок 3.2 – Древо діалогу головного меню

### Опис програмних модулів

Реалізована програма розділена на 3 основних модуля і 8 незалежних підмодулів. Такий спосіб істотно спрощує реалізацію і подальшу підтримку програми, так як її частини розділені один від одного, що дає можливість без шкоди іншим частинам вносити корективи і видозмінювати модулі.

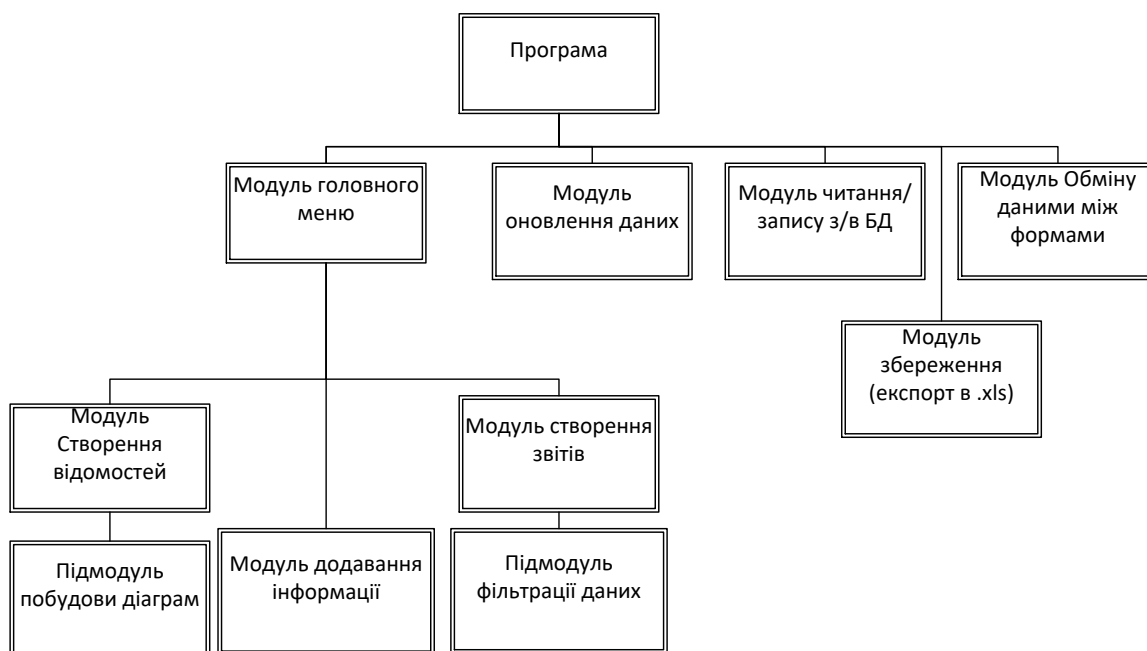


Рисунок 3.4 – Структура програмних модулів

Таблиця 3.4 – Опис модулів програми

№	Найменування модуля	Функції модуля
1	Модуль головного меню	Початковий і найголовніший модуль, що надає доступ до решти функціоналу та модулів програми. Модуль спілкування з формами - надає можливість обміну даними між різними формами / модулями програми.
2	Модуль читання/запису з/в БД	Надає функціонал для різних дій з даними між програмою і базою даних, здійснює підключення БД до системи
3	Підмодуль редагування	Даний підмодуль дає можливість редагування інформації внесеної в систему
4	Модуль формування відомостей	Надає набір засобів для складання відомостей по деяким категоріям
5	Модуль додавання інформації	Надає методи/бланки для внесення інформації в програму
6	Модуль складання звітів	Модуль дає можливість скласти і фільтрувати звіти за деякими внесеним до програми критеріями відбору даних
7	Модуль оновлення даних	Даний модуль надає засоби для оновлення інформації в програмі (в випадках якщо вона не оновилася автоматично)

Нижче наведемо приклади реалізації основних механізмів, показавши ділянки коду, який реалізує той чи інший механізм.

Приклади:

Один з основних механізмів даної системи, є взаємодія програмної частини проекту, і бази даних Access. Клас, який це здійснює: static class GetTable

```

static OleDbCommand OleDbComm = new OleDbCommand();
static OleDbConnection OleDbConn = new OleDbConnection();
static OleDbDataAdapter OleDbAdapter = new OleDbDataAdapter();
static public DataTable Get(string query)
{
    OleDbConn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=
BgamesShop.accdb;Persist Security Info=True";
    OleDbConn.ConnectionString = FormDialogMethods.ConnectionString;
    OleDbComm.Connection = OleDbConn;
    DataTable dt = new DataTable();
    OleDbAdapter.SelectCommand = new OleDbCommand(query, OleDbConn);
    OleDbAdapter.Fill(dt);
}

```

Лістинг класу наведено в додатку 1.

Даний клас містить один метод, який власне і здійснює з'єднання з БД, а також виконує запити, для роботи з таблицями які містить база даних, що передаються в метод класу як параметр.

Реалізація запитів в БД і програмний код класу наведено в додатку 2.

Даний метод надає можливість форматування виведення інформації про замовника по критерії відбору «Дата замовлення» і «Кількість» товару. Метод дозволяє вивести на екран інформацію про поставили вказаний товар постачальників в обсязі, не менше заданого за певний період. В якості вхідних параметрів служить дата початку відбору (З якої дати здійснювати вибірку) і дата кінця відбору (До якої дати здійснювати вибірку), а також кількість товару на складі.

Більшість функцій пов'язані з обробкою даних в БД, і реалізуються на основі параметричних запитів. Розглядати алгоритмічні схеми для таких процесів досить складно, в програмної реалізації практично немає фрагментів з наявністю складних вкладених циклів залежних від перевірки ряду умов, що властиво обчислювальним завданням. Тому варто говорити про певний

сценарій, так як послідовність роботи базується на подійному програмуванні та окремих діях користувача (оператора системи).

Загальна сценарій протікання процесу (роботи додатку) представлено в додатку 4.

## 4 ТЕСТОВИЙ ЗАПУСК ПРОГРАМИ, ІНСТРУКЦІЯ КОРИСТУВАЧА

В програмній реалізації передбачено досить багато функцій, особливо враховуючи, що дана розробка є практичною частиною бакалаврської роботи (прототипом), а не завершеним комерційним продуктом. Тим не менше програмний додаток є фактично повнофункціональною системою складського обліку товарів.

В даному пункті проілюструємо лише основні функції оператора системи.



Рисунок 4.1 – Заставка програмної реалізації

Після запуску програми відбувається підключення до бази даних і завантаження інформації для подальшої коректної роботи програми. При необхідності доступ до бази даних можна обмежити за допомогою стандартних методів Microsoft Access або парольного входу. Після успішного підключення до бази даних завантажується основне меню програми.

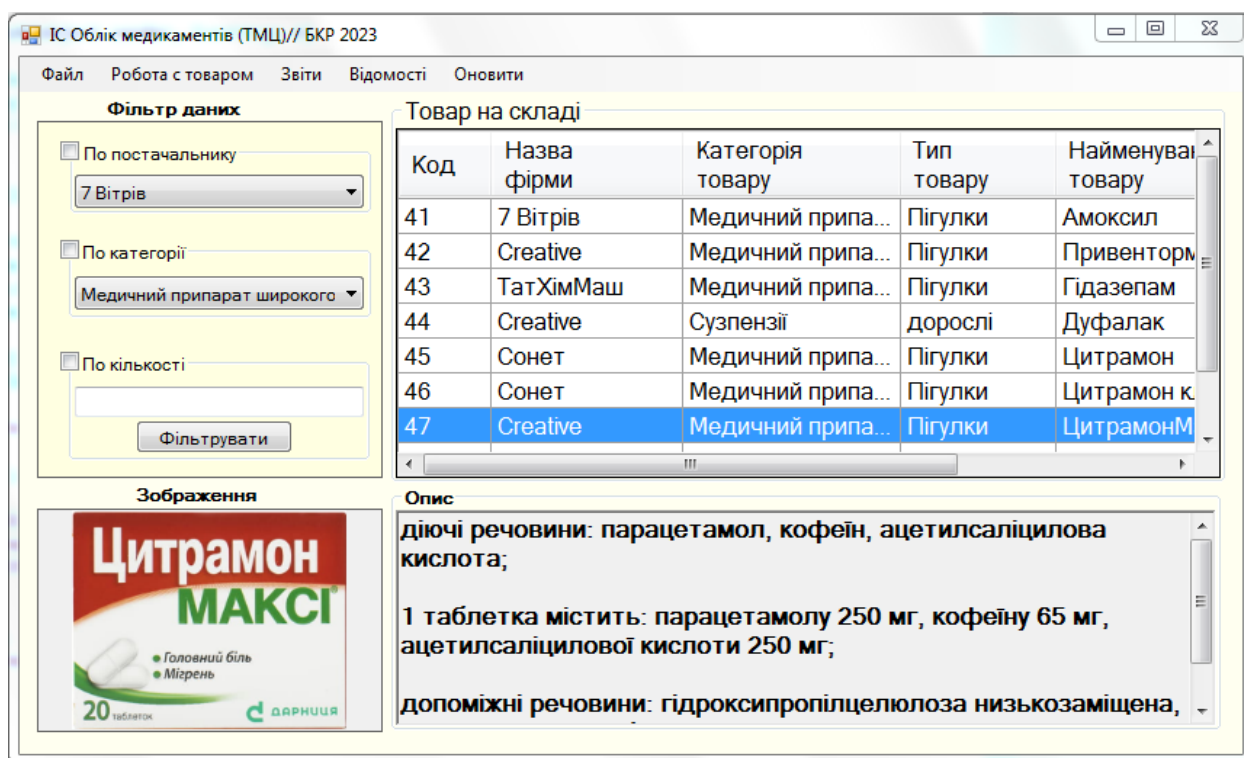


Рисунок 4.2 - Головна форма програми

У головному меню відображається інформація про товари, що знаходяться на складі. У лівому кутку знаходиться фільтр даних, який дозволяє відфільтрувати наведену інформацію за трьома категоріями: по постачальнику, по категорії товару і за кількістю товару на складі. Також меню головної форми надає доступ до решти підсистем розробленої програми.

Додати товар

Постачальник  
7 Вітрів

Категорія: Виробник

Категорія товару  
Витратні матеріали

Тип (підкатегорія) товару  
шприц

Назва товару  
Шприц 5мм3

Ціна: 560      Кількість: 1200

Склад  
№ №11 / 20х3л      Залишилось місць: 2600

Відміна      Додати

Рисунок 4.3 - Меню приходування товару на склад

Дане меню надає можливість користувачеві системи приходувати поставку товару на склад. Якщо такого товару раніше не було на складі, то система автоматично внесе його до переліку, якщо ж якась його кількість присутня, то до нього додається кількість новоприбулого товару.

До кожного товару можна додати деталізацію, що дає можливість продавцеві надати інформацію для покупця, проконсультувати, підібрати кращий варіант

Рисунок 4.4 – Додавання опису до товарної позиції

Рисунок 4.3 - Меню внесення нового замовлення від клієнта (розход товару)



Дане меню надає можливість внести замовлення від покупця на покупку товару наявного на складі. Ціна для клієнта нараховується автоматично враховуючи кількість одиниць замовлення.

**Касова відомість**

Сохранить

Закупівля

Назва фірми	Категорія товару	Тип товару	Найменування товару	Кількість	Ціна	Дата
7 Вітрів	Медичний припара...	Пігулки	ЦитрамонМакс	10	3100	07.05.2022
Creative	Витратні матеріали	шприц	п1	20	6500	07.05.2023
ТатХімМаш	Медичний припара...	Карточки, кубики, к...	п2	20	5240	07.05.2023
Creative	Медичний припара...	Карточки, кубики, к...	п3	10	2450	07.05.2023
Сонет	Суспензії	дитячі	п4	3	2550	07.05.2023
Сонет	Суспензії	комбіновані	п5	3	3750	07.05.2023
Creative	Мед. засоби	перев'язочні матер...	п6	50	710	07.05.2023

Продажі

ФІО контактного лица	Категорія товару	Тип товару	Найменування товару	Кількість	Ціна	Дата заказа
ФОП Буряченко	Суспензії	дорослі	Дуфалак	5	1375	07.05.2023
ЛікиПлюс	Медичний припара...	Пігулки	Цитрамон	1	1062	14.05.2023
ЛікиПлюс	Медичний припара...	Пігулки	Цитрамон класичний	1	1562	23.12.2022
Аптека ФармПлюс	Витратні матеріали	шприц	Шприц 5мм3	300	176	24.05.2023
Аптека ФармПлюс	Обладнання	вимірювальні прис...	Термометр цифро...	10	1125	09.06.2023

Загальний прибуток від продажів: 5300

Загальна вартість закупок: 31060

Рисунок 4.4 - Касова відомість

Касова відомість відображає списки закупівель складу і його продажів. У ній відображається вся необхідна інформація: ціна, дата, кількість і тд.

В практичній частині повністю реалізовано цикл обробки даних CRUD та основні типи звітів і відомостей. З певним доопрацюванням та розширенням функціоналу програмний комплекс можна впроваджувати в експлуатацію.

## ВИСНОВКИ

Розроблена система являє собою інформаційну систему автоматизації роботи складу ЗАТ «Farmaco+» для обліку медичних препаратів та медикаментів. Програмний комплекс працює на базі ОС сімейства Windows.

Реалізований проект являє собою зручний програмний продукт, використання якого сприяє підвищенню якості та ефективності і продуктивності праці. ІС дозволяє істотно підвищити швидкість виконання ряду бізнес процесів порівняно з первинним варіантом (без використання ІС). Особливо це стосується отримання різного роду звітностей і пошуку інформації по ключам (виробник, постачальник, дата операції та ін.). Програмна обробка даних істотно знижує ризик помилки за рахунок людського фактора - більшість даних вибирається з довідників і не вимагає постійного введення інформації. Зберігання БД в електронному вигляді дозволяє швидко зробити копію БД для підвищення збереження даних.

Навігація за доступними функціональними можливостями здійснюється за допомогою меню і системи підміню.

Розроблена система має такі переваги:

1. Виключається надмірність представлення інформації, це передбачає структура і принципи побудови самої БД
2. Зменшується вимоги до фізичному обсягу носія для зберігання інформації, що особливо актуально при великій кількості звітів.
3. Дозволяє прискорити і спростити обробку і оцінку інформації за обраними критеріями за рахунок автоматизації цих процесів.

В якості мови програмування вибрана мова "C#" і середовище розробки VisualStudio, а для зберігання даних обрана настільна СУБД MS Access.

В процесі реалізації проектних рішень з програмного забезпечення побудовані: дерево діалогу (сценарій роботи з системою), структурна схема пакета і ряд інших компонент проекту, що докладно розкривають сутність програмної реалізації задачі.

Впровадження комплексу дозволить прискорити і здешевити протікання бізнес-процесів підприємства.

Пояснювальна записка роботи складається з 4 розділів основної частини та включає вступ й висновки.

У даній БКР виконаний комплекс робіт, спрямованих на обґрунтування необхідності автоматизації: визначено сутність завдання, описані основні властивості системи, дано опис всіх існуючих процесів. Також дана характеристика інформаційної архітектури проекту, побудовано об'єктну модель. В основу проектування бази даних лягла інформаційна модель задачі, проведено моделювання «сутність-зв'язок», описана структура полів таблиць, бази даних, проаналізовані всі інформаційні потоки вхідний, оперативної, нормативно-довідкової і результативної інформації.

В першому розділі описано аналіз предметної області та постановку задачі, другий розділ присвячено моделюванню, в третьому розділі описано окремі проектні та програмні рішення. Останній розділ описує отриманий програмний продукт та взаємодію з ним.

Задачі бакалаврської роботи, поставлені у вступі, виконані в повному обсязі.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Palvia, P., Aeron, P., Gupta, P., Mahapatra, D., & Parida, R. (2015). Digital transformation and disruption of the pharmaceutical and healthcare industries. *Information Systems Frontiers*, 17(6), 1201-1210.
2. Shih, W. Y., Chang, H. Y., Chen, C. C., & Huang, C. K. (2017). An intelligent medication management system for inpatient settings: A systematic review. *Journal of Medical Systems*, 41(7), 110.
3. Kukafka, R., Ancker, J. S., Chan, C., Chelico, J. H., Khan, S., Mortoti, S., & Stabile, M. (2011). Redesigning electronic health record systems to support public health. *Journal of Biomedical Informatics*, 44(3), 401-412.
4. Wei, L., Xu, W., & He, J. (2020). Applications of blockchain technology in pharmaceutical supply chains: A systematic review. *International Journal of Environmental Research and Public Health*, 17(21), 7829.
5. Agarwal, A., Garg, P., & Srivastava, R. K. (2017). A review of decision support systems for pharmacy practice. *Expert Systems with Applications*, 83, 81-101.
6. Al-Qirim, N. (2006). The role of information systems in supporting knowledge management: The case of a pharmaceutical company in Jordan. *Information & Management*, 43(3), 366-376.
7. Fakhree, M. A., & Ismail, A. I. (2013). The role of e-business applications in enhancing pharmaceutical supply chain integration. *International Journal of Pharmacy and Pharmaceutical Sciences*, 5(4), 283-290
8. Sabah, S., & Abid, M. (2014). Pharmaceutical marketing: Ethical and legal perspectives. *Advances in Management and Applied Economics*, 4(2), 33-43.
9. Беляков К. І. Інформаційно-правові дослідження походження, становлення, стан та перспективи розвитку. *Інформація і право*. 2011. № 2 (2). С. 4–12.
10. Кубарева І, Пилюга Л. АНАЛІЗ ІНФОРМАЦІЙНО-ПРАВОВОГО ЗАБЕЗПЕЧЕННЯ ФАРМАЦЕВТИЧНОЇ ДІЯЛЬНОСТІ В УКРАЇНІ// І

Кубарева, Л. Пилюга, АДМІНІСТРАТИВНЕ ПРАВО І ПРОЦЕС, №2, -2019, стр. 105

11. Управління підприємницькою діяльністю в галузі охорони здоров'я: за ред. О. В. Баєвої та І. М. Солоненка. – К.: МАУП, 2007. – 376 с.

12. Інформаційні технології у фармації: підручник / І.Є. Булах, Л.П. Войтенко, Л.О.Кухар та ін.; за ред. І.Є. Булах. К.: Медицина, 2014. 224с.

13. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: підручник. — 2-ге вид. — К.: Каравела, 2007. — 640 с.

14. Кривуця В.Г., Барковський В.В. Інформаційні технології у фармації. Практикум: навч. посіб. — К.: Центр навч. л-ри, 2018. — 536 с.

15. Хісамова А.Р. Область застосування та особливості систем автоматизації складської діяльності// А.Р. Хісамова, Матеріали XII Міжнародної студентської наукової конференції «Студентський науковий форум», 2017

16. Винокуров А.Ю. Інформаційні технології: навчальний посібник. – К.: Просвіта, 2012.

17. Гагарина Л.Г. Автоматизированные информационные системы: учеб, пособие. М.: МИЭТ, 2011.

18. Культин А.В. Реляційні бази даних: практичні прийоми ефективних рішень. — К.: ВД «Грифон», 2018 — 400с.:ил;

## Додаток 1 Приклад коду взаємодії АІС з БД

```
using System;
using System.Collections.Generic;
using System.Data.OleDb;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
namespace BgamesShop
{
    static class GetTable
    {
        static OleDbCommand OleDbComm = new OleDbCommand();
        static OleDbConnection OleDbConn = new OleDbConnection();
        static OleDbDataAdapter OleDbAdapter = new OleDbDataAdapter();
        static public DataTable Get(string query)
        {
            OleDbConn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=
BgamesShop.accdb;Persist Security Info=True";
            // OleDbConn.ConnectionString = FormDialogMethods.ConnectionString;
            OleDbComm.Connection = OleDbConn;
            DataTable dt = new DataTable();
            OleDbAdapter.SelectCommand = new OleDbCommand(query, OleDbConn);
            OleDbAdapter.Fill(dt);
            try
            {
                OleDbConn.Open();
            }
            catch
            {
                OleDbConn.Close();
            }
            return dt;
        }
    }
}
```

## Додаток 2 Лістинг фрагмента коду, який відповідає за додавання замовлення на поставку

```

void UpdateStorage()
{
    DataTable dtGoods = new DataTable();
    dtGoods = GetTabele.Get("SELECT Код, Кількість, Ціна FROM [Товар на складі] WHERE "
+
"[Найменування товару]='" + tb_Name.Text + "' AND [Тип товару]='" +
dtType.Rows[cmb_Type.SelectedIndex][0] +
" AND Постачальник=" + dtProvider.Rows[cmb_Provider.SelectedIndex][0]);
    if (dtGoods.Rows.Count > 0)
    {
        GetTabele.Get("UPDATE [Товар на складі] SET Кількість=" +
(Convert.ToInt32(dtGoods.Rows[0][1]) + Convert.ToInt32(tb_Amount.Text)) +
", Ціна=" + (Convert.ToInt32(dtGoods.Rows[0][2]) + Convert.ToInt32(tb_Price.Text))
+
" WHERE Код=" + dtGoods.Rows[0][0]);
    }
    else
    {
        GetTabele.Get("INSERT INTO [Товар на складі] (Постачальник, [Категорія товару],
[Найменування товару], " +
"Кількість, [Тип товару], Ціна) VALUES (" +
dtProvider.Rows[cmb_Provider.SelectedIndex][0] + ", " +
dtCat.Rows[cmb_Cat.SelectedIndex][0] + ", '" + tb_Name.Text + "', " +
tb_Amount.Text + ", " +
dtType.Rows[cmb_Type.SelectedIndex][0] + ", " + tb_Price.Text + ")");
    }
}
if (Convert.ToInt32(tb_StorageNumb.Text) >= 0)
{
    if (tb_Amount.Text.Length > 0 && tb_Cat.Text.Length > 0 && tb_Name.Text.Length > 0
&& tb_Price.Text.Length > 0)
    {
        GetTabele.Get("INSERT INTO [Бланк замовлення] (Постачальник, [Категорія товару],
[Найменування товару], " +
"Ціна, Кількість, Склад, [Дата], [Тип товару]) VALUES (" +
dtProvider.Rows[cmb_Provider.SelectedIndex][0] + ", " +
dtCat.Rows[cmb_Cat.SelectedIndex][0] + ", '" + tb_Name.Text + "', " + tb_Price.Text
+
", " + tb_Amount.Text + ", " + dtStorage.Rows[cmb_StorageNumb.SelectedIndex][0] +
", '" + (DateTime.Now.Day + "/" + DateTime.Now.Month + "/" + DateTime.Now.Year + "', ")
+
dtType.Rows[cmb_Type.SelectedIndex][0] + ")");
    }
}
UpdateStorage();

GetTabele.Get("UPDATE Склад SET [Залишок місця]=" + tb_StorageNumb.Text + " WHERE Код="
+ dtStorage.Rows[cmb_StorageNumb.SelectedIndex][0]);

MessageBox.Show("Дані успішно добавлені", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);

tb_Amount.Clear();
tb_Cat.Clear();
tb_Name.Clear();
tb_Price.Clear();
tb_StorageNumb.Clear();
;

```

```
dtProvider = GetTabele.Get("SELECT p.Код, p.[Найменування фірми], c.Категорія FROM
Постачальник p, Категорія c, p LEFT JOIN c ON p.Категорія=c.Код");
cmb_Provider.DataSource = dtProvider;
cmb_Provider.DisplayMember = "Найменування фірми";
tb_Cat.Text = dtProvider.Rows[cmb_Provider.SelectedIndex][2].ToString();

dtCat = GetTabele.Get("SELECT Код, [Категорія товару] FROM [Категорія товару]");
cmb_Cat.DataSource = dtCat;
cmb_Cat.DisplayMember = "Категорія товару";

StorageName();
}
else MessageBox.Show("Не всі поля заповнені", "", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
}
else MessageBox.Show("Кількість товару перевищує кількість місць осередок складу", "",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
```



### Додаток 3 Головний клас програми ( class F\_Main)

```

public partial class F_Main : Form
{
    F_AddProvider F_AddProvider = new F_AddProvider();
    F_AddCat F_AddCat = new F_AddCat();
    F_AddStorage F_AddStorage = new F_AddStorage();
    F_AddGoods F_AddGoods = new F_AddGoods();
    F_AddOrder F_AddOrder = new F_AddOrder();

    F_Provider F_Provider = new F_Provider();
    F_Parts F_Parts = new F_Parts();
    F_Customer F_Customer = new F_Customer();
    F_Storage F_Storage = new F_Storage();
    F_InventoryLists F_InventoryLists = new F_InventoryLists();
    F_CashReport F_CashReport = new F_CashReport();
    F_Orders F_Orders = new F_Orders();
    F_AddDefectiveGoods F_AddDefectiveGoods = new F_AddDefectiveGoods();
    F_DefectiveGoods F_DefectiveGoods = new F_DefectiveGoods();
    F_ChartGoods F_ChartGoods = new F_ChartGoods();
    F_AddType F_AddType = new F_AddType();
    F_AddDescription F_AddDescription = new F_AddDescription();

    DataTable dtProv = new DataTable();

    String superFilter = "";
    public F_Main()
    {
        InitializeComponent();
    }

    private void поставальникаToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (F_AddProvider.Created == true) F_AddProvider.Close();
        F_AddProvider = new F_AddProvider();
        F_AddProvider.Show();
    }

    private void категория товара ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (F_AddCat.Created == true) F_AddCat.Close();
        F_AddCat = new F_AddCat();
        F_AddCat.Show();
    }

    private void осередокСкладуToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (F_AddStorage.Created == true) F_AddStorage.Close();
        F_AddStorage = new F_AddStorage();
        F_AddStorage.Show();
    }

    private void товарToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (F_AddGoods.Created == true) F_AddGoods.Close();
        F_AddGoods = new F_AddGoods();
        F_AddGoods.Show();
    }

    private void замовленняToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (F_AddOrder.Created == true) F_AddOrder.Close();
        F_AddOrder = new F_AddOrder();
    }
}

```

```

    F_AddOrder.Show();
}

private void поПостачальникамToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Provider.Created == true) F_Provider.Close();
    F_Provider = new F_Provider();
    F_Provider.Show();
}

private void поДеталямToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Parts.Created == true) F_Parts.Close();
    F_Parts = new F_Parts();
    F_Parts.Show();
}

private void поЗамовникамToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Customer.Created == true) F_Customer.Close();
    F_Customer = new F_Customer();
    F_Customer.Show();
}

private void F_Main_Load(object sender, EventArgs e)
{
    dtProv = GetTabele.Get("SELECT Код, [Найменування фірми] FROM Постачальник");
    cmb_Prov.DataSource = dtProv;
    cmb_Prov.DisplayMember = "Найменування фірми";

    dgv_Main.DataSource = GetTabele.Get("SELECT s.Код, р.[Найменування фірми],
к.[Категорія товару], т.[Тип товару], s.[Найменування товару], " +
"s.Кількість FROM Постачальник р, [Категорія товару] к, [Тип товару] т, [Товар на
складі] s, " +
"s LEFT JOIN р ON s.Постачальник=р.Код, s LEFT JOIN к ON s.[Категорія
товару]=к.Код, s LEFT JOIN т ON s.[Тип товару]=т.Код");
    GetCat();

    cmb_Cat.DataSource = FormDialog.dtCat;
    cmb_Cat.DisplayMember = "Категорія товару";
}

void Filter()
{
    superFilter = "";
    if (chb_Amount.Checked == true)
    {
        if (tb_Amount.Text.Length>0)
        superFilter += " AND s.Кількість>" + tb_Amount.Text;
    }
    if (chb_Cat.Checked == true) superFilter += " AND s.[Категорія
товару]="+FormDialog.dtCat.Rows[cmb_Cat.SelectedIndex][0];
    if (chb_Prov.Checked == true) superFilter += " AND s.Постачальник=" +
dtProv.Rows[cmb_Prov.SelectedIndex][0];

    dgv_Main.DataSource = GetTabele.Get("SELECT s.Код, р.[Найменування фірми],
к.[Категорія товару], т.[Тип товару], s.[ Найменування товару], " +
"s.Кількість FROM Поставник р, [Категорія товару] к, [Тип товару] т, [Товар на складі]
s, " +
"s LEFT JOIN р ON s.Постачальник=р.Код, s LEFT JOIN к ON s.[Категорія товару]=к.Код, s
LEFT JOIN т ON s.[Тип товару]=т.Код WHERE s.Ціна>-1" + superFilter);
}

void GetCat()

```

```

{
    FormDialog.dtCat = GetTabele.Get("SELECT Код, [Категорія товару] FROM [Категорія
товару]");
    FormDialog.dtType = GetTabele.Get("SELECT * FROM [Тип товару]");
    FormDialog.dtName = GetTabele.Get("SELECT Код, [Найменування товару],[Тип товару]
FROM [Товар на складі]");
}
private void поСкладуToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Storage.Created == true) F_Storage.Close();
    F_Storage = new F_Storage();
    F_Storage.Show();
}

private void інвентиразиційнаВідомістьToolStripMenuItem_Click(object sender, EventArgs
e)
{
    if (F_InventoryLists.Created == true) F_InventoryLists.Close();
    F_InventoryLists = new F_InventoryLists();
    F_InventoryLists.Show();
}

private void касовийЗвітToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_CashReport.Created == true) F_CashReport.Close();
    F_CashReport = new F_CashReport();
    F_CashReport.Show();
}

private void поЗамовленнямToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Orders.Created == true) F_Orders.Close();
    F_Orders = new F_Orders();
    F_Orders.Show();
}

private void бракованийТоварToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_AddDefectiveGoods.Created == true) F_AddDefectiveGoods.Close();
    F_AddDefectiveGoods = new F_AddDefectiveGoods();
    F_AddDefectiveGoods.Show();
}

private void бракованийТоварToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (F_DefectiveGoods.Created == true) F_DefectiveGoods.Close();
    F_DefectiveGoods = new F_DefectiveGoods();
    F_DefectiveGoods.Show();
}

private void обороткоштівToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_ChartGoods.Created == true) F_ChartGoods.Close();
    F_ChartGoods = new F_ChartGoods();
    F_ChartGoods.Show();
}
}

```

```

private void зачинитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    Close();
}

private void типТоваруToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_AddType.Created == true) F_AddType.Close();
    F_AddType = new F_AddType();
    F_AddType.Show();
}

private void оновитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    dtProv = GetTabele.Get("SELECT Код, [Найменування фірми] FROM Постачальник");
    cmb_Prov.DataSource = dtProv;
    cmb_Prov.DisplayMember = "Найменування фірми";

    dgv_Main.DataSource = GetTabele.Get("SELECT s.Код, p.[Найменування фірми],
k.[Категорія товару], t.[Тип товару], s.[Найменування товару], " +
"s.Кількість FROM Постачальник p, [Категорія товару] k, [Тип товару] t, [Товар на
складі] s, " +
"s LEFT JOIN p ON s.Постачальник=p.Код, s LEFT JOIN k ON s.[Категорія
товару]=k.Код, s LEFT JOIN t ON s.[Тип товару]=t.Код");
    GetCat();

    cmb_Cat.DataSource = FormDialog.dtCat;
    cmb_Cat.DisplayMember = "Категорія товару";

    chb_Amount.Checked = false;
    chb_Cat.Checked = false;
    chb_Prov.Checked = false;
}

private void поПостачальникамToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (F_Provider.Created == true) F_Provider.Close();
    F_Provider = new F_Provider();
    F_Provider.Show();
}

private void поЗамовленнямToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (F_Orders.Created == true) F_Orders.Close();
    F_Orders = new F_Orders();
    F_Orders.Show();
}

private void поТоваруToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Parts.Created == true) F_Parts.Close();
    F_Parts = new F_Parts();
    F_Parts.Show();
}

private void поЗамовникамToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (F_Customer.Created == true) F_Customer.Close();
    F_Customer = new F_Customer();
    F_Customer.Show();
}

private void поСкладськомуПриміщеннюToolStripMenuItem_Click(object sender, EventArgs e)

```

```
{
    if (F_Storage.Created == true) F_Storage.Close();
    F_Storage = new F_Storage();
    F_Storage.Show();
}

private void інвентиразиційнаВідомістьToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (F_InventoryLists.Created == true) F_InventoryLists.Close();
    F_InventoryLists = new F_InventoryLists();
    F_InventoryLists.Show();
}

private void касоваВідомістьToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_CashReport.Created == true) F_CashReport.Close();
    F_CashReport = new F_CashReport();
    F_CashReport.Show();
}

private void оборотКоштівToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (F_ChartGoods.Created == true) F_ChartGoods.Close();
    F_ChartGoods = new F_ChartGoods();
    F_ChartGoods.Show();
}

private void поБракованомуТоваруToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_DefectiveGoods.Created == true) F_DefectiveGoods.Close();
    F_DefectiveGoods = new F_DefectiveGoods();
    F_DefectiveGoods.Show();
}

private void chb_Prov_CheckedChanged(object sender, EventArgs e)
{
    Filter();
}

private void chb_Cat_CheckedChanged(object sender, EventArgs e)
{
    Filter();
}

private void chb_Amount_CheckedChanged(object sender, EventArgs e)
{
    Filter();
}

private void cmb_Prov_SelectedIndexChanged(object sender, EventArgs e)
{
    Filter();
}

private void cmb_Cat_SelectedIndexChanged(object sender, EventArgs e)
{
    Filter();
}

private void btn_AmountSearch_Click(object sender, EventArgs e)
{
    Filter();
}
```

```
private void dgv_Main_SelectionChanged(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    dt = GetTabele.Get("SELECT Зображення, Опис FROM Опис WHERE
Номер="+ (int)dgv_Main.CurrentRow.Cells[0].Value);
    if (dt.Rows.Count > 0)
    {
        rbtn_Text.Text = dt.Rows[0][1].ToString();

        pb_Main.ImageLocation = @"Зображення/"+dt.Rows[0][0].ToString();
        pb_Main.BackgroundImageLayout = ImageLayout.Zoom;
        pb_Main.SizeMode = PictureBoxSizeMode.Zoom;
    }
    else
    {
        rbtn_Text.Clear();

        pb_Main.ImageLocation = null;
    }
}

private void tb_Amount_KeyPress(object sender, KeyPressEventArgs e)
{
    char l = e.KeyChar;
    if (l != '\b' && (l < '0' || l > '9'))
    {
        e.Handled = true;
    }
}

private void описToolStripMenuItem_Click(object sender, EventArgs e)
{
    FormDialog.Index = (int)dgv_Main.CurrentRow.Cells[0].Value;
    FormDialog.Cat = dgv_Main.CurrentRow.Cells[2].Value.ToString();
    FormDialog.Name = dgv_Main.CurrentRow.Cells[4].Value.ToString();
    FormDialog.Prov = dgv_Main.CurrentRow.Cells[3].Value.ToString();

    if (F_AddDescription.Created == true) F_AddDescription.Close();
    F_AddDescription = new F_AddDescription();
    F_AddDescription.Show();
}
}
```

## Додаток 4 Загальний алгоритм обробки даних

