Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматики
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

**Бакалаврська кваліфікаційна робота на тему:**

# ПРОГРАМНИЙ МОДУЛЬ ПРОГНОЗУВАННЯ ОБСЯГУ ЗАКУПІВЛІ КВІТІВ

Виконала: студентка 4 курсу, групи 1КН-21б
спеціальності 122 – Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)
Моклюк Д.М.
(прізвище та ініціали)
Керівник: к.т.н., доцент кафедри КН
Озеранський В.С.
(прізвище та ініціали)
«04» червня 2025 р.
Рецензент: к.т.н., доцент кафедри САІТ
Варчук І.В.
(прізвище та ініціали)
«05» червня 2025 р.

**Допущено до захисту**
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
«06» червня 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Комп'ютерні науки»

**ЗАТВЕРДЖУЮ**
Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
"05" січня 2025 року

# ЗАВДАННЯ
# НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Моклюк Дарії Миколаївні

1. Тема роботи: Програмний модуль прогнозування обсягу закупівлі квітів.
керівник роботи: Озеранський Володимир Сергійович, к.т.н., доцент
затверджені наказом вищого навчального закладу "20" березня 2025 року №97

2. Строк подання студентом роботи 20 січня 2025 р.

3. Вихідні дані до роботи:
мова програмування – об'єктно-орієнтована, має забезпечувати можливість маніпулювання даними та управління базами даних;
період прогнозування – не менше 4 днів;
кількість товару – не менше 10шт;
максимально можлива кількість замовлення – не більше 1000шт;
інтерфейс користувача має бути інтуїтивно зрозумілим.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
вступ, обґрунтування доцільності розробки програмного модуля прогнозування обсягу закупки квітів, розробка програмного модуля прогнозування обсягу закупки квітів, програмна реалізація модуля прогнозування обсягу закупки квітів, висновки, список використаних джерел, додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
схема загального алгоритму функціонування системи; структура програмного модуля; модель функціонування системи прогнозування обсягу закупки квітів; загальна UML-діаграма класів.

6. Консультанти розділів проєкту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| 1-3 | Озеранський В.С., к.т.н., доцент | | |
| | | | |
| | | | |
| | | | |

6. Дата видачі завдання "05" травня 2025 року

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва та зміст етапу | Термін виконання | | Примітка |
|---|---|---|---|---|
| | | початок | закінчення | |
| 1. | Обгрунтування доцільності розробки програмного модуля прогнозування обсягу закупівлі квітів | 05.05.25 | 07.05.2025 | |
| 2. | Розробка програмного модуля прогнозування обсягу закупівлі квітів | 08.05.2025 | 15.05.2025 | |
| 3. | Програмна реалізація модуля прогнозування обсягу закупівлі квітів | 16.05.2025 | 26.05.2025 | |
| 4. | Розробка інструкції користувача. | 27.05.2025 | 29.05.2025 | |
| 5. | Висновки та аналіз отриманих результатів | 30.05.2025 | 01.06.2025 | |
| 6. | Оформлення матеріалів до захисту БКР | 02.06.2025 | 04.06.2025 | |

Студентка _____ (підпис) Моклюк Д.М. (прізвище та ініціа

Керівник проєкту (роботи) _____ (підпис) Озеранський В (прізвище та ініціа

# АНОТАЦІЯ

УДК 621.374.415

Моклюк Д.М. Програмний модуль прогнозування обсягу закупівлі квітів. Бакалаврська кваліфікаційна робота складається з 94 сторінки формату А4, на яких є 18 рисунків, 3 таблиці, список використаних джерел містить 22 найменуваннь.

В роботі обґрунтовано доцільність розробки програмного модуль прогнозування обсягу закупівлі квітів, розглянуто принципи прогнозування та планування обсягу закупки квітів для квіткових магазинів. Проведено аналіз сучасних програм-аналогів, які використовуються для прогнозування закупки і продажу товарів. Розроблено структуру програмного модуля прогнозування обсягу закупівлі квітів. Розроблено алгоритми функціонування програмного модуля, а також наведено алгоритм виведення архіву та алгоритм запису інформації в базу даних.

Програмний модуль розроблено на об’єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність та простоту розробки, в середовищі розробки IntelliJ IDEA.

Ключові слова: IntelliJ IDEA, Java, прогноз, квіти, закупівля, прогнозування.

## ABSTRACT

Moklyuk D. M. Software module of forecasting the volume of flower purchases. Bachelor's qualification work consists of 94 pages of A4 format, which has 18 figures, 3 tables, the list of sources used contains 22 names.

The work substantiates the feasibility of developing a software module for forecasting the volume of flower purchases, the principles of forecasting and planning the volume of purchasing flowers for flower shops are considered. The analysis of modern analog programs used to forecast the purchase and sale of goods has been analyzed. The structure of the software module of forecasting the volume of flower purchases has been developed. Algorithms for the functioning of the software module have been developed, as well as the algorithm of the archive and the algorithm for recording information in the database.

The software module is designed in the Java-oriented Object-Oriented Language, taking into account its cross-platform and ease of development, in the development of Intellij Idea development.



Keywords: IntelliJ IDEA, Java, forecast, flowers, purchase, forecasting.

# ЗМІСТ

# ВСТУП

**Актуальність теми.** На сьогоднішній день прогнозування обсягу закупки квітів із застосування сучасних інформаційних технологій зумовлена активним розвитком та ростом обсягів квіткового бізнесу, його затребуваністю та прибутковістю, а також складністю прогнозування доходів та обсягу закупки квітів вручну, що потребує розробки ефективних і доступних методів її прогнозування.

Квіти – один з головних атрибутів свята. Традиція дарувати квіти й прикрашати ними простір має глибокі культурні та емоційні корені й міцно утвердилася в Україні. Букети супроводжують найважливіші події в житті людини – дні народження, весілля, релігійні свята, професійні урочистості, вшанування пам'яті, а також є важливою частиною корпоративного етикету. Завдяки цьому квітковий бізнес має стабільний попит, навіть попри сезонні коливання чи економічні труднощі.

Флористи та квіткові магазини, як правило, завжди знаходять свого покупця, незалежно від пори року або загального економічного клімату. Однак попит на квіти має свою специфіку: він дуже чутливий до святкових дат, погодних умов, купівельної спроможності населення та змін у поведінці споживачів. З другого боку, обсяги квітів для подарунку є своєрідним відображенням фінансового стану як окремих споживачів, так і підприємств. У бізнес-середовищі кількість та масштаб замовлень квіткової продукції для корпоративних заходів, подарункових композицій чи оформлення офісних просторів можуть слугувати індикатором економічного стану компанії: чим кращі її справи, тим частіше і щедріше вона купує квіти.

Планування збуту – це дуже важливий крок, оскільки від цього залежить постачання квіткової продукції. Як правило, квіти купуються власниками торгових точок оптом. Планування продажів квітникарів базується на сезонності квіткової діяльності. Пік продажів припадає на наступні дати – 14 лютого, 8 березня, 8 травня, 1 вересня. Сьогодні існує велика кількість оптових баз квітів, які готові працювати з усіма споживачами, оскільки вони завжди мають велику кількість товарів. Іншим важливим завданням у цьому плані є налагоджена система доставки квітів. Свіжі

квіти слід купувати кожні 4–5 днів. У сучасних умовах динамічного розвитку ринку квіткової продукції, прогнозування обсягу закупівлі квітів набуває особливого значення. Сезонні коливання попиту, святкові періоди, тренди споживчої поведінки та нестабільність економічної ситуації вимагають від підприємств квіткового бізнесу чіткого планування закупівель. Надлишкові закупівлі призводять до збитків через швидке псування квітів, а дефіцит товару — до втрати прибутку та довіри клієнтів. У зв'язку з цим прогнозування обсягу закупівлі квітів є важливим та актуальним питанням для тих, хто займається таким бізнесом.

**Метою бакалаврської кваліфікаційної роботи** є підвищення точності прогнозування обсягу закупівлі квітів.

**Об'єктом дослідження** є процес прогнозування обсягу закупівлі квітів.

**Предметом дослідження** є програмні засоби прогнозування обсягу закупівлі квітів.

Для досягнення поставленої мети необхідно виконати такі задачі:

- обґрунтувати доцільність розробки програмного модуля прогнозування обсягу закупівлі квітів;
- проаналізувати переваги та недоліки програм-аналогів для прогнозування обсягу закупівлі квітів;
- розробити алгоритм прогнозування обсягу закупівлі квітів;
- здійснити програмну реалізацію модуля прогнозування обсягу закупівлі квітів;
- виконати тестування програмного модуля та провести аналіз отриманих результатів.

**Апробація результатів роботи.** Результати роботи були апробовані на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» (м. Вінниця, Україна, 2025 р.) [1]. Результати дослідження впроваджено в роботу ТОВ «ІТІ».

**Публікації.** За результатами досліджень опубліковано тези доповіді на науково-практичній конференції [1].

# 1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ОБСЯГУ ЗАКУПІВЛІ КВІТІВ

## 1.1 Особливості ринку квітів в Україні

Ринок квітів – один із небагатьох ринків в Україні, який характеризується відсутністю великих власників та мережевих продажів. Незважаючи на зростання продажів, ринок квітів представлений стихійною торгівлею, відсутністю сервісу та сучасних технологій. Таким чином, вітчизняні компанії займають близько 30% внутрішнього ринку, а все інше – імпортне.

Класично ринок квіткової продукції можна розділити на три основні сегменти: ринок зрізаних квітів; горщикової продукції та посадкового матеріалу. Наразі немає відкритої статистики чи іншої кількісної інформації щодо оцінки ринку зрізаних квітів в Україні. За оцінками експертів, більше половини всіх зрізаних квітів, які продаються в Україні, є імпортними. Менша частка – квіти вітчизняного виробництва, їх вирощують як спеціалізовані підприємства (переважно троянди), так і непрофесійно – у господарствах на невеликих ділянках, які потім продаються переважно в невеликих кіосках і на натуральних базарах. У сегменті зрізаних квітів в Україні переважає виробництво троянд, гербер, гвоздик, цибулин. Найпопулярніші квіти на квітковому ринку в горщиках - фікус, орхідея, драцена, спатифіллум, азалія, бегонія. Найбільше їх імпортують з Нідерландів (90%), Польщі, Данії та Італії. Загалом, аналіз квіткового ринку України показує, що його обсяг значною мірою залежить від фінансового благополуччя населення нашої країни. Так було під час кризи 2014-2015 років, ринок квітів впав на 58%, а в наступні два роки, після початку економічного зростання, збільшився на 40% [2]. Сьогодні спостерігається тенденція до зниження ринку квітів, оскільки війна в Україні може знизити популярність ринку квітів, особливо їх імпорту. Тому очікується розвиток вітчизняного виробництва популярних квітів, особливо троянд. У цьому секторі з’являться нові учасники, щоб конкурувати з провідними галузями сучасності, такими як Асканія-Флора, Камелія, Фрезія та Тандем. Троянди є основою продажу

(до 70%) асортименту зрізаних квітів протягом року: це троянди, які дарують на будь-якому заході, тому вони широко використовуються у флористиці. Найпопулярнішою є червона троянда (майже половина всіх троянд), за нею йде біла троянда (25%), троянди інших кольорів часто складають не більше 25% асортименту [3]. Наявність інших квітів в асортименті залежить від пори року. Український ринок досить консервативний, тому асортимент складається переважно з класичних категорій, новинки дуже рідкісні. Структура продажу свіжих зрізаних квітів сильно варіюється в залежності від різних свят.

Сьогодні більшість українців дарує квіти на свята. Найбільші доходи продавці квітів приносять 8 березня та 14 лютого, на які припадає 13,5% і 9,4% річного обсягу продажів відповідно. Ще 5,6% додає 1 вересня традиція ходити до школи з букетом [4]. Структура продажів зрізаних квітів у святкові дні в Україні продемонстрована на рисунку 1.1.

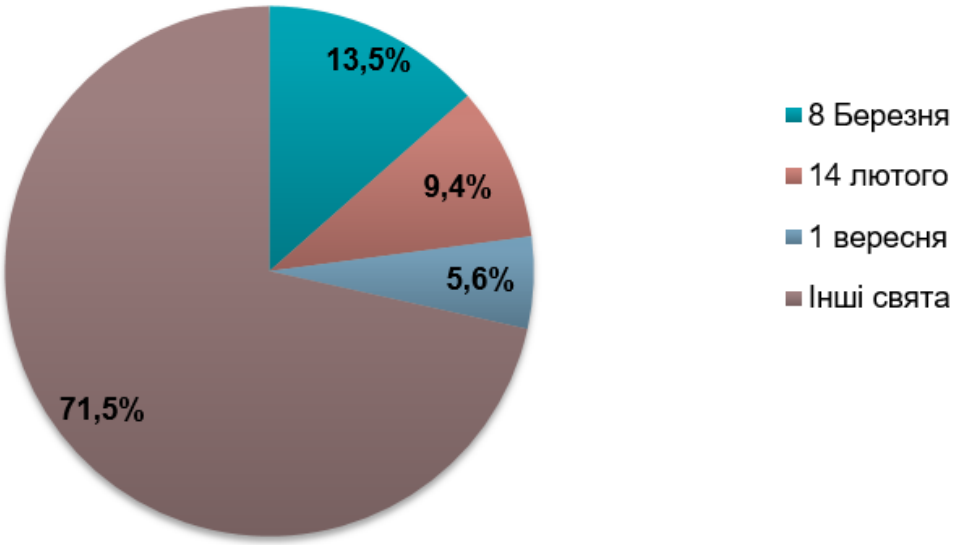



Рисунок 1.1 – Структура продажів зрізаних квітів у святкові дні

Аналізуючи ринок квітів, слід зазначити, що більшість квітів імпортного походження. Вирощують троянди переважно в українських теплицях. Друге місце в структурі займають тюльпани, але їх частка не перевищує 2%.

Для цього ринку характерна сезонність – кількість імпортних квітів більше взимку та навесні, а влітку та восени збільшується кількість квітів від вітчизняних виробників.

Аналіз ринку квітів в Україні показує, що найбільш різноманітний він у столиці. Регіональні оптові бази поповнюються квітами приблизно раз на тиждень, і чим менше місто, тим менше точок і асортимент квітів.

Використовуючи аналіз ринку квітів в Україні, можна виділити наступні основні канали збуту [5]:

- продовольчі ринки та невеликі кіоски - на них припадає 60-70% продажів квітів через відносно низькі ціни, свіжість продукції та зручне розташування на контрольно-пропускних пунктах;

- фірмові флористи – підтримують частку продажів на рівні 20-25%; намагатися сформувати конкурентоспроможне співвідношення ціна-якість шляхом прямого контакту з виробниками та додатковими послугами;

- супермаркети та гіпермаркети, бакалія та сегменти DIY – становлять близько 5% від загального обсягу продажів; перевага віддається вже готовим букетам, які покупці купують на додаток до основного подарунка, придбаного тут;

- інтернет-магазини – працюють як окремо, так і як служби доставки для офлайн-магазинів; дозволяють подарувати квіти кур'єром на відстані.

## 1.2 Дослідження принципів прогнозування та планування обсягу закупівлі квітів для квіткових магазинів

Прогноз продажу – один з найважливіших етапів ведення бізнесу: підприємству повинно бути представлене повідомлення про те, що він продає, за яку суму і з якою рентабельністю. Методи прогнозування продажів здійснюють різні, починаючи від елементарних, закінчуючи тими, які складаються за допомогою потужних математичних інструментів [6].

Планування – безпосередньо поняття завдання, яке ставиться перед менеджером. Прогноз - пропозиція про том, що в якій перспективі магазин продає

певне кількісне співтовариство. Прогноз - це не завдання, яке необхідно виконати, а саме пропонування про те, як досягти його виконання. Прогноз завжди має під собою визначену базу, він ніколи не робить із передбачуваних, пов’язаних, наприклад, із бажанням підприємця отримати вигоду в який-небудь період [7].

Щодо форм прогнозування, то слід зазначити, що існують два різновиди:

- прогноз ринку в цілому, що визначає прогноз збуту окремих товарів, що виробляються всіма підприємствами. Далі розраховується частка ринку компанії і, таким чином, розраховується прогнозна вартість реалізації її товарів;

- пряме прогнозування, при якому підприємство безпосередньо визначає вартість прогнозу, виходячи з інформації про власний продаж своєї продукції. У той же час він не прогнозує ринок в цілому. У більшості випадків маркетологи пропонують таку форму прогнозування, хоча цей підхід менш ефективний. Він недостатньо враховує загальний розвиток збуту на ринку.

Достовірність прогнозів продажів певною мірою залежить від використовуваних принципів. Відомо, що прогнозування збуту – відповідальне і складне завдання. Застосування відповідних принципів може забезпечити належну точність відповідних результатів, отриманих у процесі прогнозування.

Основні принципи прогнозування продажів включають [8]:

- об'єктивність прогнозів. Це забезпечує відповідність стану навколишнього середовища, на якому базується прогноз, та його результатів. Реалізація цього принципу прогнозування передбачає:

а) використання достовірної інформації за минулий період;

б) використання в очікуванні найбільш ефективних методів за цих умов. Нижче буде зазначено, що існують різні методи прогнозування.

У кожній конкретній ситуації необхідно вибрати ту, яка враховує характер і зміст середовища, тип прогнозу тощо.

в) одночасне використання кількох методів прогнозування. Дуже часто в бізнесі паралельно використовуються такі методи, як регресійний аналіз і метод експоненційного згладжування. За допомогою регресійного аналізу досягається більш висока точність прогнозу (коли при прогнозуванні необхідно враховувати

різні фактори). Аналогічно передбачення поєднуються з математичними методами, зокрема з такими, як експоненціальне згладжування [9];

- мінімізація кількості ринкових факторів, що впливають на результати прогнозу. Чим можна пояснити необхідність використання цього принципу? Значна кількість факторів ускладнює процес прогнозування. Перед прогнозистом постає проблема визначення їх реального впливу на величину прогнозу. По-друге, чим більше факторів, тим вище ймовірність дублювання їх впливу на результати прогнозу. І по-третє, модель регресійного аналізу, яка спирається на значну кількість факторів, буде досить складною з точки зору розуміння їх впливу на продажі нерухомості;

- використання різних оцінок результатів прогнозу (оптимістичних, найбільш ймовірних і песимістичних). Це пов'язано з можливістю розвитку різних ринкових ситуацій (сприятливі умови, найгірші ситуації, проміжні прогнози між цими двома крайнощами);

- відповідність прогнозу цілям використання його результатів. Нагадаємо, що з цією метою є обґрунтування поставлених перед збутом товарів цілей. Тому ця кореспонденція в першу чергу стосується термінів, на які робляться прогнози. Вони повинні відповідати умовам, для яких складаються плани збуту. Причому ця відповідність стосується факторів, які впливають як на вартість прогнозу, так і на значення прогнозних цілей продажу товарів. У цьому випадку виходить більша кореляція між прогнозами та планами;

- з урахуванням максимальних можливостей прогнозу. В даному випадку мова йде про точність прогнозів продажів різних товарів. Наприклад, здатність прогнозувати продажі вже існуючих товарів у цьому плані набагато більше, ніж, скажімо, нових товарів. На прогноз продажів останнього можуть вплинути фактори, повний перелік яких важко передбачити. Те саме стосується сезонних продуктів, продуктів, призначених для продажу на зовнішніх ринках. Тому менеджери, розробляючи прогнози збуту цих товарів, повинні враховувати обмеження можливості прогнозування їх збуту.

Прогнозування обсягу закупки та продажу товару, що використовується компаніями, зазвичай різноманітне. Воно зумовлене великою кількістю планів, розроблених менеджерами з маркетингу [10]. Класифікація прогнозування представлена в таблиці 1.1.

Таблиця 1.1 – Класифікація та види прогнозування

| № | Ознаки класифікації прогнозів | Види прогнозів |
|---|---|---|
| 1. | Терміни, на які складаються прогнози | 1. Короткострокові<br>2. Середньострокові<br>3. Довгострокові<br>4. Прогнози, що відповідають тривалості сезону |
| 2. | Об’єкти прогнозування | 1. Прогнозу для всього асортименту<br>2. Прогноз для окремих товарів |
| 3. | Характер ринку | 1. Весь ринок<br>2. Окремі сегменти<br>3. Внутрішній ринок<br>4. Зовнішній ринок<br>5. Торгові зони |
| 4. | Характер процесу прогнозування | 1. Пошукові<br>2. Інтуїтивні<br>3. Нормативні |
| 5. | Точність кількісної оцінки результатів | 1. З однозначною оцінкою<br>2. З інтервальними оцінками |
| 6. | Ступінь імовірності настання результатів | 1. Імовірнісні<br>2. Інваріативні |

З точки зору прогнозування обсягу закупки квітів, необхідно звернути увагу на короткострокові прогнози, адже закупка квітів відбувається в основному на період часу від 4 до 10 днів.

## 1.3 Існуючі технології, що застосовуються для реалізації поставленої задачі

Для прогнозування обсягу закупки квітів доцільно використовувати кількісні методи прогнозування.

Кількісні методи, на відміну від якісних методів прогнозування збуту, спираються на точний розрахунок обсягу продажів, який спирається на використання набору статистичних даних. Такі методи зазвичай використовують аналіз часових рядів, що передбачає врахування обсягу закупки минулого року в такий же період наступного року.

Їх застосовують у випадках, коли закономірні тенденції розвитку збуту, характерні для попереднього періоду, залишаються незмінними для прогнозного періоду. Іншими словами, доцільно використовувати методи кількісного прогнозування при стабільності економічної ситуації як у попередні роки, так і в прогнозний період. Це дозволяє екстраполювати тенденцію продажів, яка склалася за минулий період, на прогнозні роки.

Серед найбільш поширених методів кількісного прогнозування виділяють:

- прогноз за ковзними середніми;
- експоненційне згладжування;
- прогнозування найменших квадратів;
- графічний метод;
- метод Бокса-Дженкінса.

Метод прогнозування з використанням ковзних середніх є досить поширеним завдяки своїй простоті. За допомогою цього методу прогнозована вартість продажів у майбутньому буде дорівнювати його середньому обсягу минулих років. Такий підхід пояснюється тим, що фактори, які діяли як у прогнозному періоді, так і в майбутньому, незмінні [7].

Однак з часом вони можуть змінюватися. Тому необхідно взяти дані з кількох минулих періодів, усереднювати їх і визначити прогноз продажів на майбутнє за національною ознакою. Чим більше років минулого періоду буде включено в розрахунок, тим точнішим буде прогноз продажів.

Середня поточна вартість і-го року визначається як частка від ділення суми продажів за останні роки на кількість років, за які продажі враховуються при розрахунку середніх оборотних значень.

Такий підхід до прогнозування збуту доцільний за умови, що його обсяг з роками не змінюється. У випадках, коли спостерігається тенденція до зростання обсягів продажів за рік, навіть незначне, прогноз збуту може бути нижчим за його фактичне значення в окремі роки минулого періоду. Це нелогічно. Більше того, в умовах стабільної економіки закономірно динамічно збільшувати продажі. Щоб уникнути такої ситуації, необхідно змінити підхід до використання ковзних середніх у прогнозі. Прогнозна вартість продажів буде розрахована в такому порядку:

- збирається статистична інформація про фактичні обсяги реалізації минулих років;

- визначаються ковзні середні (бажано три);

- значення цих величин систематично попарно порівнюють і встановлюють їх зростання;

- обчислює середнє збільшення цих значень за досліджуваний період як відношення їх суми до величини;

- шляхом додавання цього збільшення до фактичного обсягу реалізації останнього року досліджуваного періоду та визначає значення прогнозу на наступний рік.

У випадках, коли статистика минулих років «змішана», тобто одні з них більш змістовні та точні, а інші – навпаки, використовується метод прогнозування, який називають експоненційним згладжуванням. Цей метод незалежно відкрили Браун (1958) і Холт (1957). Цей тип прогнозу використовує значення числового ряду попередніх років. Його суть полягає в застосуванні константи згладжування. У цьому випадку розрахункова вартість продажів розраховується за такою формулою [12]:

$$ОП_{t+1} = ax_t + (1 - a)ОП_t \quad (1.1)$$

де $ОП_{t+1}$ – прогнозована величина продажу часового періоду $t + 1$;

$a$ – константа згладжування;

$x_t$ – фактичний обсяг продажу періоду $t$;

$ОП_t$ – прогнозований обсяг продажу періоду $t$.

Вибір конкретного значення константи згладжування, який виконується довільно, впливає на значення прогнозу продажів. Якщо він великий, це означає, що фактичний продаж минулого періоду стає більше. Константа меншого значення збільшує вплив на прогнози продажів на більш віддалені періоди. Тому важливу роль у прогнозуванні збуту тут грає правильне визначення значення константи згладжування [13].

На практиці фахівець, який займається прогнозом, приймає свою цінність виходячи з:

- вивчення інформації, що використовується в прогнозі;

- його інтуїція, що сформувалася з часом;

- врахування подібності умов, що мали місце в минулому і прогнозованих періодах;

- умови прогнозу.

Щоб знайти константи згладжування, пропонується використовувати квазіньютоновський алгоритму для максимізації правдоподібності (ймовірності).

Експоненціальне згладжування як метод прогнозування продажів має як переваги, так і недоліки.

Переваги:

- простота використання;

- здатність прогнозиста визначати вплив на точність прогнозу на певний період.

Однак у нього є деякі недоліки. До них належать:

- нездатність передбачити зростання або спад ринку;

- довільний вибір значення константи згладжування, яке синоптик визначає самостійно.

Незважаючи на ці недоліки, ряд компаній досягли успіху використовуючи експоненціальне згладжування в практиці прогнозування продажів.

Теорії і практиці відомий також такий метод прогнозування збуту, як метод найменших квадратів. Він відноситься до методів екстраполяції [14].

Його суть полягає в мінімізації суми квадратів відхилень між спостережуваними значеннями і розрахунковими значеннями. Прогноз продажів буде точнішим, коли відомі між цими параметрами менші. Прогноз продажів для цього методу визначається за формулою:

$$y_{t+1} = a + bx_n, \tag{1.2}$$

де $y_{t+1}$ – прогнозований обсяг продажу;

$t + 1$ – період, на який здійснювався прогноз;

$a$ і $b$ – коефіцієнти;

$x_n$ – умовне позначення часу (номер періоду, на який здійснювався прогноз);

$$a = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \cdot \sum_{i}^{n} x_i - \left(\sum_{i}^{n} x_i\right)^2} \tag{1.3}$$

$$b = \frac{n \cdot \sum_{i=1}^{n} x_i y_i - \sum_{i}^{n} x_i \sum_{i}^{n} y_i}{n \cdot \sum_{i}^{n} x_i^2 - \left(\sum_{i}^{n} x_i\right)^2} \tag{1.4}$$

де $n$ – кількість років, за які взята статистична інформація для прогнозування;

$x_i$ – номер $i$-го року;

$y_i$ – обсяг продажу в цьому році.

Можна також використовувати графічний метод з побудовою діаграми для прогнозування продажів. Це досить просто. Суть цього методу полягає в наступному [5]:

- збирається інформація про фактичні обсяги реалізації за попередні роки;

- будується система координат. Періоди часу відкладаються на осі х, а обсяги продажів – на осі у;

- перетин перпендикулярів, опущених уздовж двох осей, фіксується точкою;

- через ці точки проходить пряма, рівновіддалена від усіх цих точок;

- від точки, розташованої на осі абсцис і відповідної прогнозованому році, перпендикуляр до перетину з проведеною лінією, яка фіксується точкою;

- з цієї точки падає перпендикулярно до осі у. Точка перетину і буде визначати вартість прогнозованих продажів.

Точність прогнозу збуту за допомогою графічного методу залежить від уміння прогнозиста правильно провести (рівновіддалену від усіх точок) пряму лінію.

Наступним розглянемо метод Бокса-Дженкінса. Він поєднує моделі авторегресії та ковзного середнього, щоб утворити змішану модель ARIMA, яка є основною моделлю серед інших моделей цього класу. Модель авторегресії – це модель часового ряду [16].

У ньому значення часового ряду в даний момент часу лінійно залежить від значень того самого ряду, який їм передує. Часовий ряд – це набір даних певного показника, в даному випадку обсягів продажів, за відповідну кількість послідовних періодів. Останні можуть описувати різні часові ряди - як стаціонарні, так і нестаціонарні.

Стаціонарні часові ряди характеризуються постійним середнім значенням, а їх дані коливаються навколо нього з відносно постійною дисперсією. Модель ARIMA відноситься до лінійних моделей класу.

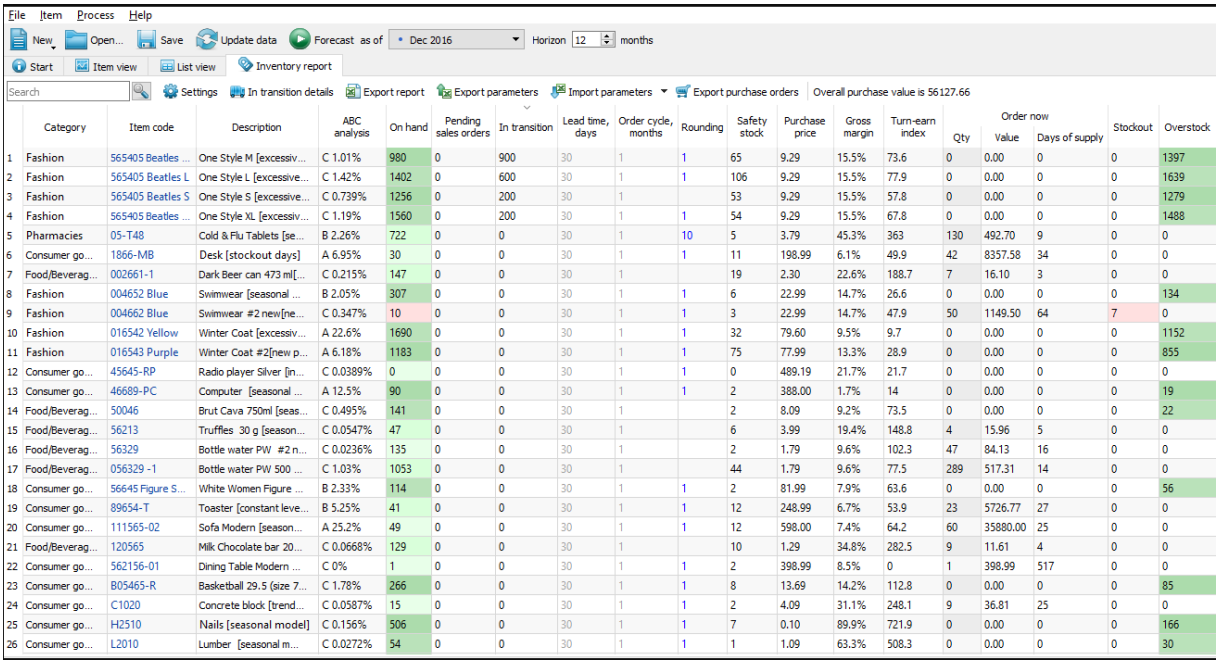
Прогнозування продажів за методом ARIMA не передбачає чіткої моделі. Тут визначається лише їхній загальний клас. А також наявний опис часового ряду. Це дозволяє визначити поточний стан змінної. Після цього алгоритм самостійно вибирає необхідні моделі прогнозування.

Для прогнозування обсягу закупки квітів існує потреба у врахуванні тренду та сезонності, однак є квіти, що продаються протягом всього року, а є ті, що можна вважати несезонними товарами. Для цього необхідно поєднати метод ковзного середнього та експонентного згладжування для сезонних та несезонних квітів відповідно.

## 1.4 Аналіз відомих програмних реалізацій для прогнозування обсягу закупки квітів

Кількість запасів є одним з головних критеріїв, що гарантують стійке функціонування компанії. Крім того, відсутність та надлишок товарних запасів на складі спричиняє підприємство зазнати збитків. Щоб цього не сталося, необхідно правильно організувати систему закупівель, що неможливо без своєчасного планування.

Досвід показує, що найбільш прогресивним варіантом планування закупівель є застосування спеціалізованих програм, наприклад, Streamline [10]. Розглянемо дану програму детальніше (рис. 1.2).




| | Category | Item code | Description | ABC analysis | On hand | Pending sales orders | In transition | Lead time, days | Order cycle, months | Rounding | Safety stock | Purchase price | Gross margin | Turn-earn index | Order now | | | Stockout | Overstock |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | Qty | Value | Days of supply | | |
| 1 | Fashion | 565405 Beatles ... | One Style M [excessiv... | C 1.01% | 980 | 0 | 900 | 30 | 1 | 1 | 65 | 9.29 | 15.5% | 73.6 | 0 | 0.00 | 0 | 0 | 1397 |
| 2 | Fashion | 565405 Beatles L | One Style L [excessive... | C 1.42% | 1402 | 0 | 600 | 30 | 1 | 1 | 106 | 9.29 | 15.5% | 77.9 | 0 | 0.00 | 0 | 0 | 1639 |
| 3 | Fashion | 565405 Beatles S | One Style S [excessive... | C 0.739% | 1256 | 0 | 200 | 30 | 1 | | 53 | 9.29 | 15.5% | 57.8 | 0 | 0.00 | 0 | 0 | 1279 |
| 4 | Fashion | 565405 Beatles ... | One Style XL [excessiv... | C 1.19% | 1560 | 0 | 200 | 30 | 1 | 1 | 54 | 9.29 | 15.5% | 67.8 | 0 | 0.00 | 0 | 0 | 1488 |
| 5 | Pharmacies | 05-T48 | Cold & Flu Tablets [se... | B 2.26% | 722 | 0 | 0 | 30 | 1 | 10 | 5 | 3.79 | 45.3% | 363 | 130 | 492.70 | 9 | 0 | 0 |
| 6 | Consumer go... | 1866-MB | Desk [stockout days] | A 6.95% | 30 | 0 | 0 | 30 | 1 | 1 | 11 | 198.99 | 6.1% | 49.9 | 42 | 8357.58 | 34 | 0 | 0 |
| 7 | Food/Beverag... | 002661-1 | Dark Beer can 473 ml[... | C 0.215% | 147 | 0 | 0 | 30 | 1 | | 19 | 2.30 | 22.6% | 188.7 | 7 | 16.10 | 3 | 0 | 0 |
| 8 | Fashion | 004652 Blue | Swimwear [seasonal ... | B 2.05% | 307 | 0 | 0 | 30 | 1 | 1 | 6 | 22.99 | 14.7% | 26.6 | 0 | 0.00 | 0 | 0 | 134 |
| 9 | Fashion | 004662 Blue | Swimwear #2 new[ne... | C 0.347% | 10 | 0 | 0 | 30 | 1 | 1 | 3 | 22.99 | 14.7% | 47.9 | 50 | 1149.50 | 64 | 7 | 0 |
| 10 | Fashion | 016542 Yellow | Winter Coat [excessiv... | A 22.6% | 1690 | 0 | 0 | 30 | 1 | 1 | 32 | 79.60 | 9.5% | 9.7 | 0 | 0.00 | 0 | 0 | 1152 |
| 11 | Fashion | 016543 Purple | Winter Coat #2[new p... | A 6.18% | 1183 | 0 | 0 | 30 | 1 | 1 | 75 | 77.99 | 13.3% | 28.9 | 0 | 0.00 | 0 | 0 | 855 |
| 12 | Consumer go... | 45645-RP | Radio player Silver [in... | C 0.0389% | 0 | 0 | 0 | 30 | 1 | 1 | 0 | 489.19 | 21.7% | 21.7 | 0 | 0.00 | 0 | 0 | 0 |
| 13 | Consumer go... | 46689-PC | Computer [seasonal ... | A 12.5% | 90 | 0 | 0 | 30 | 1 | 1 | 2 | 388.00 | 1.7% | 14 | 0 | 0.00 | 0 | 0 | 19 |
| 14 | Food/Beverag... | 50046 | Brut Cava 750ml [seas... | C 0.495% | 141 | 0 | 0 | 30 | 1 | | 2 | 8.09 | 9.2% | 73.5 | 0 | 0.00 | 0 | 0 | 22 |
| 15 | Food/Beverag... | 56213 | Truffles 30 g [season... | C 0.0547% | 47 | 0 | 0 | 30 | 1 | | 6 | 3.99 | 19.4% | 148.8 | 4 | 15.96 | 5 | 0 | 0 |
| 16 | Food/Beverag... | 56329 | Bottle water PW #2 n... | C 0.0236% | 135 | 0 | 0 | 30 | 1 | | 2 | 1.79 | 9.6% | 102.3 | 47 | 84.13 | 16 | 0 | 0 |
| 17 | Food/Beverag... | 056329 -1 | Bottle water PW 500 ... | C 1.03% | 1053 | 0 | 0 | 30 | 1 | | 44 | 1.79 | 9.6% | 77.5 | 289 | 517.31 | 14 | 0 | 0 |
| 18 | Consumer go... | 56645 Figure S... | White Women Figure ... | B 2.33% | 114 | 0 | 0 | 30 | 1 | 1 | 2 | 81.99 | 7.9% | 63.6 | 0 | 0.00 | 0 | 0 | 56 |
| 19 | Consumer go... | 89654-T | Toaster [constant leve... | B 5.25% | 41 | 0 | 0 | 30 | 1 | 1 | 12 | 248.99 | 6.7% | 53.9 | 23 | 5726.77 | 27 | 0 | 0 |
| 20 | Consumer go... | 111565-02 | Sofa Modern [season... | A 25.2% | 49 | 0 | 0 | 30 | 1 | 1 | 12 | 598.00 | 7.4% | 64.2 | 60 | 35880.00 | 25 | 0 | 0 |
| 21 | Food/Beverag... | 120565 | Milk Chocolate bar 20... | C 0.0668% | 129 | 0 | 0 | 30 | 1 | | 10 | 1.29 | 34.8% | 282.5 | 9 | 11.61 | 4 | 0 | 0 |
| 22 | Consumer go... | 562156-01 | Dining Table Modern ... | C 0% | 1 | 0 | 0 | 30 | 1 | 1 | 2 | 398.99 | 8.5% | 0 | 1 | 398.99 | 517 | 0 | 0 |
| 23 | Consumer go... | B05465-R | Basketball 29.5 (size 7... | C 1.78% | 266 | 0 | 0 | 30 | 1 | 1 | 8 | 13.69 | 14.2% | 112.8 | 0 | 0.00 | 0 | 0 | 85 |
| 24 | Consumer go... | C1020 | Concrete block [trend... | C 0.0587% | 15 | 0 | 0 | 30 | 1 | 1 | 2 | 4.09 | 31.1% | 248.1 | 9 | 36.81 | 25 | 0 | 0 |
| 25 | Consumer go... | H2510 | Nails [seasonal model] | C 0.156% | 506 | 0 | 0 | 30 | 1 | 1 | 7 | 0.10 | 89.9% | 721.9 | 0 | 0.00 | 0 | 0 | 166 |
| 26 | Consumer go... | L2010 | Lumber [seasonal m... | C 0.0272% | 54 | 0 | 0 | 30 | 1 | 1 | 1 | 1.09 | 63.3% | 508.3 | 0 | 0.00 | 0 | 0 | 30 |

Рисунок 1.2 – Приклад роботи програми Streamline

Програмне забезпечення Streamline поєднує в собі високу продуктивність та ефективність, дозволяючи вам зосередитися на стратегічному розвитку та довгострокових цілях вашого бізнесу. Двостороння інтеграція забезпечує імпорт даних із вашої системи продажів до Streamline, а також автоматичний експорт прогнозів замовлень назад у вашу ERP-систему. Для успішного впровадження

необхідна узгоджена робота багатьох складових. Команда Streamline має глибокі знання про сучасні системи продажів і ERP, що дозволяє забезпечити безперешкодний запуск і повну готовність вашої команди. Система управління запасами повинна відповідати вашим бізнес-процесам і цілям. При виборі рішення варто враховувати такі аспекти, як загальна вартість володіння, надійність, якість технічної підтримки та можливість повноцінного тестування функціоналу перед остаточним вибором.

Як діяти, якщо стратегія поповнення за принципом мінімум/максимум, вбудована у вашу ERP-систему, подає сигнал на закупівлю лише для одного артикулу, тоді як інші товари того ж постачальника ще не потребують поповнення? Адже ERP генерує сигнали замовлення на рівні окремого SKU, тоді як бізнес зазвичай оформлює закупівлі по постачальнику в цілому. У результаті ви або ігноруєте сигнал і ризикуєте залишитись без товару, або замовляєте надмірно, щоб заповнити контейнер. На відміну від традиційного підходу ERP, Streamline формує сигнали закупівлі на рівні постачальника. Програма прогнозує всі потреби в закупівлі на майбутній цикл, використовуючи моделювання дискретних подій. Це дозволяє заздалегідь формувати замовлення, забезпечуючи стабільний процес закупівель — незалежно від того, чи йдеться про фіксовані цикли, закупівлю повного контейнера або економічно обґрунтований обсяг замовлення (EOQ). Замість статичних формул, Streamline застосовує симуляцію дискретних подій із щоденним рівнем деталізації. Це дозволяє точно моделювати реальні потоки запасів і ефективно справлятися з комплексними ситуаціями в ланцюгу постачань – з якими Excel впоратись не здатен.

На відміну від інших рішень, які часто спрощують обчислення, не враховуючи реальні події, Streamline формує часову шкалу з щоденною деталізацією, на яку наносяться всі заплановані дії. Потім система відтворює хронологію подій, забезпечуючи максимально точне відображення змін у запасах з точністю до одного дня. У деяких випадках це просто більш точний підхід, ніж традиційні формули поповнення, але часто це єдиний метод, здатний врахувати складність реального ланцюга постачань.

Оцінки сезонності, цінової еластичності чи прогнози на основі загальних тенденцій більше не є достатніми. Сучасний ринок швидко змінюється, і складно сказати, наскільки актуальна історія продажів для поточної ситуації та чи підходить вона для прогнозування майбутнього. У цьому контексті ми використовуємо власні алгоритми штучного інтелекту, які визначають доцільність застосування методів прогнозування часових рядів, зміни рівнів і трендів — так, якби ви щодня аналізували кожен окремий артикул вручну.

Ви користуєтеся EOQ у своїй роботі? Якщо ні – варто звернути на нього увагу, адже ця концепція управління запасами дозволяє суттєво скоротити витрати на зберігання та оформлення замовлень. Проте традиційний підхід до EOQ передбачає розрахунок для кожного SKU окремо, тоді як у реальному житті закупівлі зазвичай охоплюють групи, а іноді й сотні товарних позицій.

Streamline, окрім підтримки класичного EOQ, пропонує вдосконалений груповий варіант, який адаптовано до реалій закупівель із кількома SKU в одному замовленні. Завдяки функції синхронізації дат замовлень для груп товарів, Streamline може гнучко зміщувати межу синхронізації вперед або назад, щоб визначити оптимальний цикл замовлення. Це дозволяє автоматично знайти найвигідніше поєднання витрат на зберігання й замовлення для всієї групи SKU.

Anaplan – це хмарна платформа для фінансового та операційного планування й моделювання бізнес-процесів. На ринку бізнес-планування, де довгий час домінувала "велика четвірка" – IBM, Oracle, SAP і Microsoft, компанія з’явилася відносно нещодавно. Головний технічний директор компанії Майкл Гулд створив Anaplan як альтернативу традиційним системам, таким як Oracle Hyperion Planning, IBM Cognos TM1 та SAP BPC. В основі технології Anaplan лежить єдиний хмарний обчислювальний центр, у якому бізнес-користувачі можуть створювати, розгортати, використовувати й спільно керувати бізнес-моделями без необхідності володіння спеціальними IT-знаннями [11].

Компанія Anaplan має штаб-квартиру в Сан-Франциско та офіси у Великобританії, Франції, Швеції, країнах Бенілюксу, Сінгапурі.

Використовуючи хмарні технології, архітектуру даних in-memory та обчислювальну систему, всі дані якої зберігаються в оперативній пам’яті (технологія HyperBlock™), Anaplan створив гнучку платформу, що надається кінцевим користувачам у форматі Software-as-a-Service — програмного забезпечення як послуги (рис. 1.3).


| | Q1 FY15 | Q2 FY15 | Q3 FY15 | Q4 FY15 | FY15 |
|---|---|---|---|---|---|
| TOTAL REVENUE | 25,432,046 | 41,603,432 | 48,576,660 | 50,385,794 | 165,997,932 |
| TOTAL COST OF SALES | (19,212,883) | (30,508,255) | (35,351,642) | (36,148,240) | (121,221,021) |
| GROSS PROFIT | 6,219,162 | 11,095,177 | 13,225,018 | 14,237,554 | 44,776,910 |
| Gross Margin % | 24.45% | 26.67% | 27.23% | 28.26% | 26.97% |
| OPERATING EXPENSES | (4,400,278) | (4,539,487) | (4,642,801) | (4,520,794) | (18,103,360) |
| OPERATING INCOME | 1,818,884 | 6,555,690 | 8,582,217 | 9,716,760 | 26,673,551 |
| Operating Margin % | 7.15% | 15.76% | 17.67% | 19.28% | 16.07% |

Рисунок 1.3 – Приклад роботи програми AnaPlan

Технологія HyperBlock™ – це архітектура, яка поєднує характеристики реляційних, колонкових (вертикальних) та OLAP-баз даних з обчисленнями та зберіганням даних в оперативній пам’яті. У традиційних електронних таблицях для планування кожна клітинка має унікальну формулу, результат якої залежить від інших клітинок. Натомість HyperBlock™ автоматично оновлює зміни на будь-якому рівні, змінюючи лише пов’язані клітинки. Незалежно від обсягу даних, кількості користувачів або складності завдань, користувачі можуть миттєво оновлювати або змінювати моделі будь-якого розміру.

Унікальна технологія «Living Blueprint» є мозковим центром платформи — єдиним репозиторієм бізнес-логіки, обчислень, налаштувань та інших параметрів

моделі, що дозволяє в режимі реального часу вносити зміни та забезпечує гнучке масштабування й адаптацію моделей до динамічного бізнес-середовища.

Рішення Anaplan – це програмне забезпечення як послуга (Software-as-a-Service), побудоване на основі хмарних обчислень. Такий підхід дозволяє користувачам самостійно розвивати моделі без залучення консультантів і відповідних витрат, а також не потребує інвестицій в IT-інфраструктуру чи спеціалістів. Крім того, це забезпечує можливість «розгортання з нуля» – користувачі можуть отримати доступ до платформи майже з будь-якого пристрою та в будь-якому місці.

DALLAS – (BUSINESS WIRE) – Компанія o9, провідний постачальник програмного забезпечення на базі штучного інтелекту для трансформації планування та прийняття рішень, сьогодні оголосила про вдосконалення своєї платформи Digital Brain шляхом інтеграції генеративних композитних агентів, створених на основі великих мовних моделей (LLM). Ці агенти нового покоління покликані докорінно змінити підхід до виконання складних завдань у плануванні, забезпечуючи подальший розвиток можливостей інтегрованого бізнес-планування.

Композитні агенти o9 побудовані на базі атомарних агентів – систем на основі штучного інтелекту, які здатні виконувати завдання, отримувати інформацію та генерувати відповіді на запити, виходячи з конкретних вхідних даних. Вони поєднуються з великими мовними моделями (LLM), навченими на інформації, специфічній для конкретного бізнесу.

У межах основного компоненту платформи o9 Digital Brain – корпоративного графа знань (Enterprise Knowledge Graph, EKG), композитні агенти можуть поєднувати послідовність атомарних агентів для виконання складніших задач, які використовуються у міжфункціональних процесах планування – таких як створення прогнозів або післяопераційний аналіз. Це відбувається шляхом послідовного збору та синтезу даних для досягнення певного результату – наприклад, аналізу змін прогнозів з місяця в місяць або проведення післяопераційного аналізу попереднього кварталу з порівнянням прогнозу та фактичних результатів і визначення причин неточностей.

Крім того, композитні агенти навчені на «рецептах» – шаблонах дій, яких дотримуються експерти з планування в організації, – і здатні постійно вдосконалюватись завдяки зворотному зв’язку. Це дозволяє агентам з часом дедалі краще досягати бажаних результатів.

EKG (Enterprise Knowledge Graph) у складі Digital Brain забезпечує міжфункціональну взаємодію, охоплюючи ланцюг постачань, фінанси, закупівлі, комерційний блок, клієнтів і постачальників. Він об’єднує знання з усіх цих напрямів — і постійно накопичує нову інформацію.

o9 Solutions – провідна платформа на основі штучного інтелекту для інтегрованого бізнес-планування та прийняття рішень у масштабах підприємства. Незалежно від того, йдеться про формування попиту, узгодження попиту та пропозиції чи оптимізацію комерційних ініціатив – будь-який процес планування можна зробити швидшим і розумнішим завдяки цифровим рішенням o9, що працюють на базі AI (рис. 1.4).

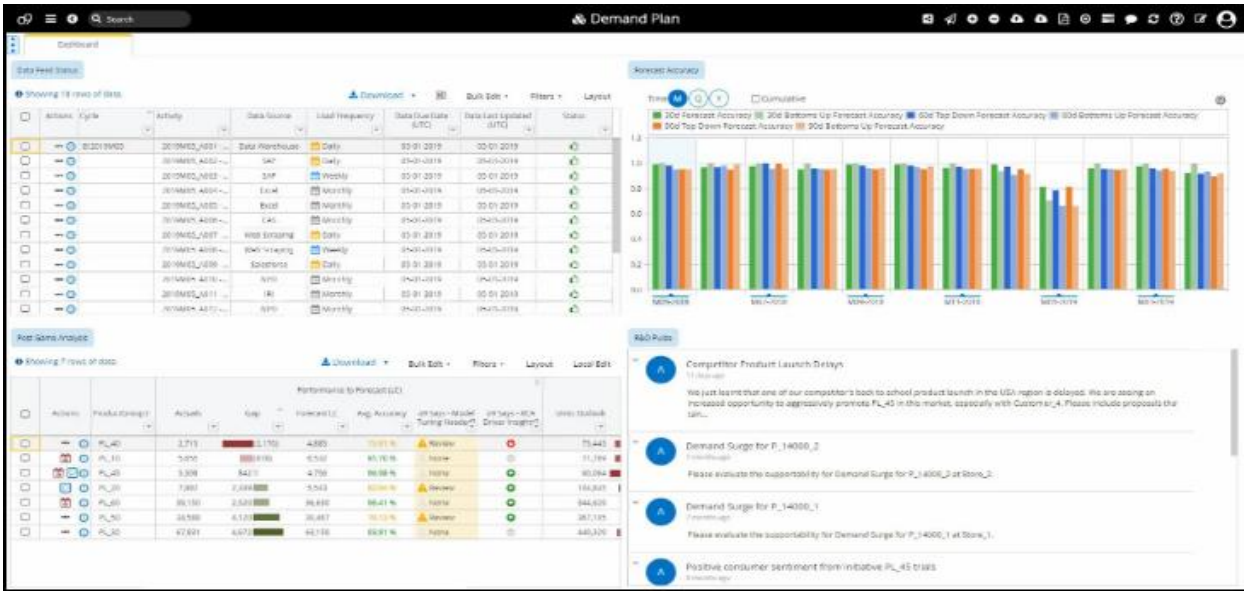



Рисунок 1.4 – Приклад роботи o9 Solutions Digital Brain Platform

Компанія o9 об’єднує інноваційні технології – такі як моделювання підприємства на основі графів, аналітика великих даних, просунуті алгоритми для сценарного планування, портали для співпраці, зручні інтерфейси та хмарна доставка – в єдину платформу [12].

Однак багато функцій є зайвими для вирішення проблеми прогнозування закупки квітів, що може перевантажити магазин та є недоцільним вкладенням коштів. Також вагомим недоліком для магазину квітів є те, що програма більше орієнтована не на прогнозування обсягу закупок, а на їх автоматизацію.

Серед проаналізованих програмних систем, «Streamline», «o9 Solutions Digital Brain Platform» являються багатофункціональними та дорогими програмами, що мають вагомі переваги для підприємств у сфері продажу, однак непотрібні для прогнозування обсягу закупки квітів. Дані програми містять інструмент для вирішення нашої задачі, проте їх недоцільно використовувати, зважаючи на ціну.

Наближеним по функціональних характеристиках є програма «AnaPlan», але вона теж має недолік у вигляді ціни, а також недостатню точність прогнозування для такого товару як зрізані квіти, адже навіть незначні похибки можуть призвести до великих втрат. Саме тому в подальшому обираємо як прототип цю систему, однак вдосконалимо точність прогнозування для запобігання неприємних ситуацій у вигляді неправильного прогнозу та, як наслідок, втрати коштів.

## 1.5 Висновок

В даному розділі було досліджено особливості ринку квітів в Україні. Також розглянуто принципи прогнозування та планування обсягу закупки квітів для квіткових магазинів. Було проведено аналіз сучасних програм-аналогів, які використовуються для прогнозування закупки і продажу товарів. Було наведено короткий опис основних функцій, які виконують дані програми. Також було досліджено методи, що можуть бути використані для прогнозування обсягів продаж, для чого запропоновано використати методи аналізу часових рядів – метод ковзного середнього та експонентного згладжування.

# 2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ОБСЯГУ ЗАКУПІВЛІ КВІТІВ

## 2.1 Обґрунтування вибору методу прогнозування обсягу закупки квітів

Прогнозування на основі часових рядів є одним із найпопулярніших підходів до прогнозування розвитку економічних процесів, обсягів торгових операцій, обсягів виробництва та накопичення продукції на складах, оцінки альтернативних економічних стратегій, формування бюджетів підприємства та держави, прогнозування та управління економічними та фінансовими ризиками та інші.

Аналіз часового ряду починається з побудови та вивчення його графіка. Якщо нестаціонарність часового ряду очевидна, то перш за все необхідно виділити нестаціонарну складову ряду. Процес виявлення тренду та інших складових ряду, які призводять до порушення стаціонарності, може проходити в кілька етапів. На кожному з них розглядається ряд залишків, отриманий в результаті розрахунку з вихідного ряду обраної моделі тренду, або результат диференціальних та інших перетворень ряду. Крім графіків, ознаками нестаціонарності часового ряду можуть бути автокореляційна функція, яка не прямує до нуля (за винятком дуже великих значень лагів) і наявність на періодограмі яскраво виражених піків на низьких частотах. За допомогою функції автокореляції також досліджуються внутрішні зв’язки між елементами часового ряду [17].

У вибіркових дослідженнях найпростіші числові характеристики описової статистики (середнє, медіана, дисперсія, стандартне відхилення, коефіцієнти асиметрії та ексцесу) зазвичай дають достатньо інформативну картину вибірки. При цьому графічні методи представлення та аналізу вибірок відіграють лише допоміжну роль, дозволяючи краще зрозуміти локалізацію та концентрацію даних, закон їх розподілу. Зовсім інша роль графічних методів в аналізі часових рядів. Табличне представлення часових рядів і описова статистика в цілому не дозволяють зрозуміти природу процесу, тоді як з графіка часових рядів можна зробити досить багато висновків.

Метод прогнозування з використанням ковзних середніх досить поширений завдяки своїй простоті. При цьому методі очікуваний обсяг закупки у майбутньому періоді дорівнюватиме середньому його обсягу попередніх років. Такий підхід пояснюється тим, що фактори, які діяли як у прогнозному періоді, так і в майбутньому, залишаються незмінними.

Проте з часом вони можуть змінюватися. Тому необхідно взяти дані кількох минулих періодів, усереднити їх і на основі цього визначити прогноз закупки квітів на майбутнє. Чим більше років минулого періоду буде включено в розрахунок, тим точніше буде прогноз закупки.

Середня ковзна вартість і-го року визначається як частка суми обсягів закупок за останні роки, поділена на кількість років, обсяг закупки яких враховується при розрахунку середніх ковзних величин.

Такий підхід до прогнозування закупки сезонних квітів є доцільним за умови, що їх обсяги мало змінюються від року до року. У цих випадках, коли спостерігається тенденція зростання обсягу закупки сезонних квітів протягом багатьох років, навіть якщо вона незначна, може виявитися, що прогнози закупок будуть нижчими за реальні потреби в окремі минулі періоди. Крім того, в умовах стабільної економіки природно динамічно нарощувати обсяг продажів. Для уникнення такої ситуації, необхідно трохи змінити підхід до використання ковзних середніх в прогнозі. При цьому прогнозне значення закупки слід розраховувати в такому порядку:

— збір статистичної інформації про фактичні обсяги закупки та продажу минулих років;

— визначення ковзних середніх;

— послідовне порівняння величин цих значень та визначення їх зростання;

— обрахунок середнього приросту цих значень за досліджуваний період як відношення їх суми до кількості;

— додавання приросту до фактичного обсягу реалізації за останній рік досліджуваного періоду, визначення значення прогнозу на наступний період.

Отже, метод ковзного середнього доцільно використовувати для прогнозування обсягу закупки сезонних квітів.

Що стосується несезонних квітів, доречно розглянути метод експотенціального згладжування.

Експоненціальне згладжування – це метод згладжування часових рядів, процедура розрахунку якого включає обробку всіх попередніх спостережень з урахуванням старіння інформації з віддаленням від прогнозного періоду. Іншими словами, чим «старше» спостереження, тим менше воно повинно впливати на значення прогнозної оцінки. Ідея експоненціального згладжування полягає в тому, що в міру старіння даних спостережень їм призначають зменшувальні ваги. [2]

Цей метод прогнозування вважається дуже ефективним. Основні переваги методу полягають у можливості врахування ваг вихідної інформації, у простоті розрахункових операцій, у гнучкості опису динаміки різних процесів. Метод експоненціального згладжування дає змогу отримати оцінку параметрів тренду, які характеризують не середній рівень процесу, а тренд, що склався під час останнього спостереження. Найбільшого поширення метод отримав при виконанні середньострокових прогнозів. Для експоненціального методу згладжування основним є вибір параметра згладжування (константи згладжування) і початкових умов.

## 2.2 Розробка структури програмного модуля прогнозування обсягу закупівлі квітів

Прогнозування обсягу закупки квітів потребує великої обчислювальної потужності. Вхідними даними є період часу, на який необхідно зробити прогноз, враховуючи архівні дані замовлень закупки квітів. Оскільки імпорт вхідних даних виконується через базу даних, користувачеві потрібно лише визначити термін прогнозування. Після отримання вхідних даних – періоду прогнозування – відбувається встановлення зв’язку з базою даних, де міститься інформація про обсяги закупки попередніх місяців та відповідного періоду часу минулих років.

Завдяки цій інформації формується обсяг закупки квітів та виводиться прогнозований результат. Програма здатна автоматично опрацьовувати всю доступну вхідну інформацію, введену користувачем та наявну в базі даних, а також вивести результат у доступному форматі, забезпечивши зручність та зрозумілість подання інформації.

В розробленій програмі будуть представлені наступні модулі: модуль інтерфейсу, модуль прогнозування обсягу закупки квітів, модуль роботи з базою даних. Модель функціонування програми прогнозування обсягу закупки квітів наведена на рисунку 2.1.

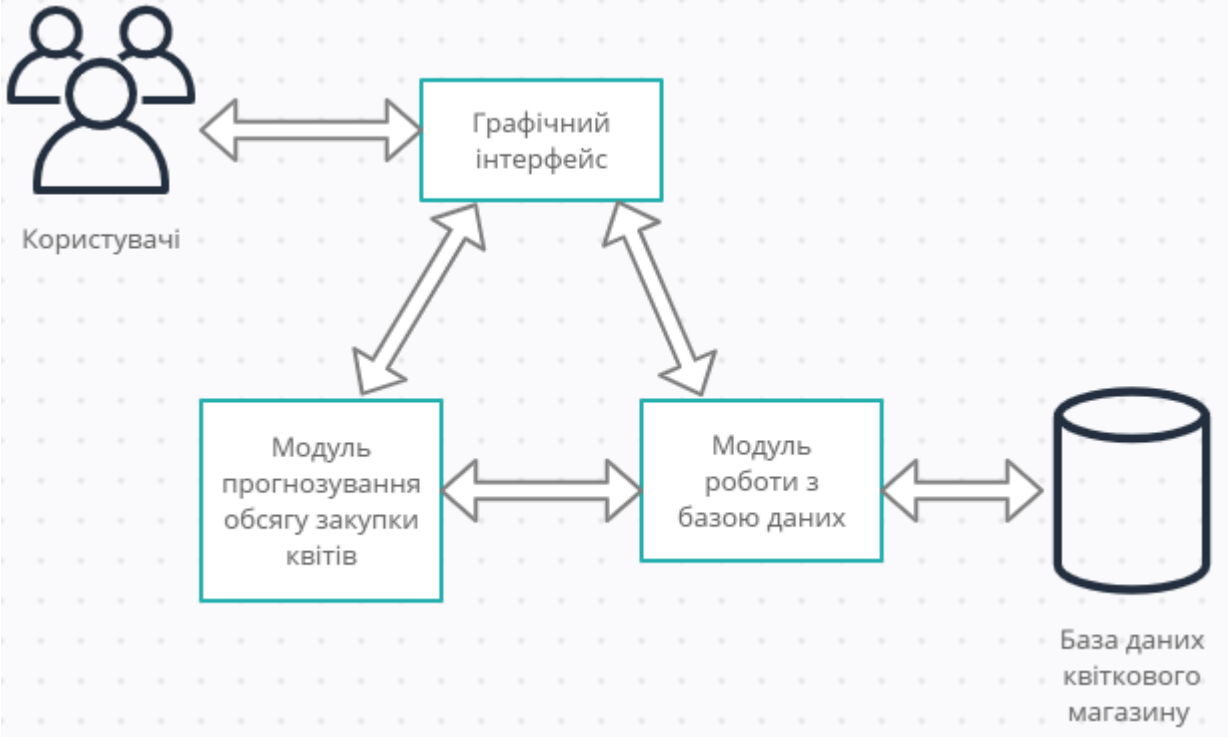



Рисунок 2.1 – Модель функціонування програмного модуля прогнозування обсягу закупівлі квітів

Завдяки графічному інтерфейсу, що пов'язаний з модулями прогнозування та роботи з базою даних користувач отримує всю необхідну йому інформацію. Він відповідає за коректне відображення на екрані результатів прогнозування, а також за введення та виведення відповідної інформації. Користувач може додавати

інформацію, переглядати архіви, вводити проміжки часу, на які здійснюватиметься прогноз, переглядати прогнозовані значення завдяки модулю інтерфейсу.

Модуль прогнозування пов'язаний з модулем роботи з базою даних. Цей модуль є головним модулем програми та містить інтелектуальний алгоритм прогнозування обсягу закупівлі квітів. Прогнозування відбується при застосуванні методу аналізу часових рядів.

Модуль роботи з базою даних пов'язаний з модулем прогнозування та модулем інтерфейсу. Даний модуль забезпечує зв'язок з базою даних, в якій міститься дані квіткового магазину. Цей модуль містить запити на мові SQL, які дозволяють переглядати архів, що містить інформацію про закупку квітів минулих місяців. SQL є основою багатьох баз даних, оскільки відповідає за фізичну структуру і запис даних на диск, а також читання даних з диска, дозволяє отримувати SQL-запити від інших компонентів бази даних і користувацьких програм [5]. Таким чином, SQL є потужним інструментом, який надає користувачам, програмам і комп'ютерним системам доступ до інформації в реляційних базах даних. База даних повинна бути синхронізованою з програмний модулем прогнозування обсягу закупки квітів та забезпечувати швидкий імпорт та автоматичний експорт усієї інформації, необхідної для прогнозування.

## 2.3 Розробка алгоритмів функціонування програмного модуля прогнозування обсягу закупівлі квітів

Розглянемо загальний алгоритм роботи додатку, що реалізує програмний модуль прогнозування обсягу закупки квітів (рис. 2.2).

Даний алгоритм складається з наступних кроків:

1. На першому кроці відбувається встановлення зв'язку з базою даних.

2. На другому етапі відбувається виведення головного меню програми, в якому відображено три варіанти подальших дій: показ архіву, додавання інформації та прогнозування.

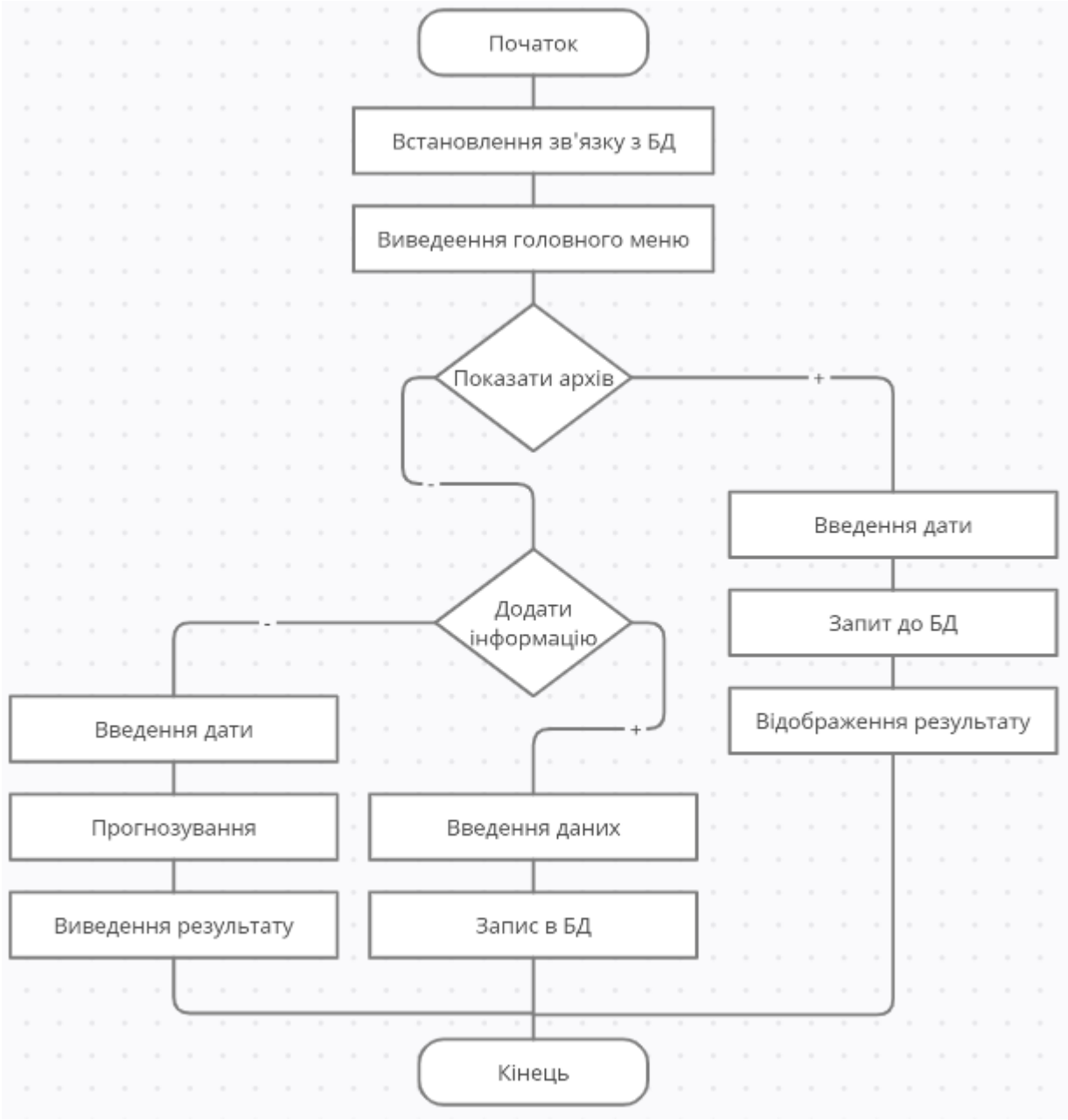



Рисунок 2.2 – Схема загального алгоритму функціонування програмного модуля прогнозування обсягу закупівлі квітів

3. Перевірка на необхідність відображення архіву.

3.1. Введення початкової та кінцевої дати для показу архіву.

3.2. На даному етапі виконується запит до бази даних щодо виведення обсягів закупки квітів в обраний період.

3.3. Відображення результату. Перехід до пункту 8.

4. Перевірка на необхідність додавання інформації.

4.1. Введення даних щодо обсягу закупки квітів певної дати у запропоноване вікно.

4.2. Запис введених даних до бази даних.

5. Введення початкової та кінцевої дати – часового проміжку, на який має здійснюватися прогнозування.

6. Прогнозування обсягу закупки квітів на вказаний період.

7. Виведення результату прогнозування.

8. Закінчення діалогу з системою.

На рисунку 2.3 зображено алгоритм виведення архіву, що містить інформацію про попередні закупки квітів.

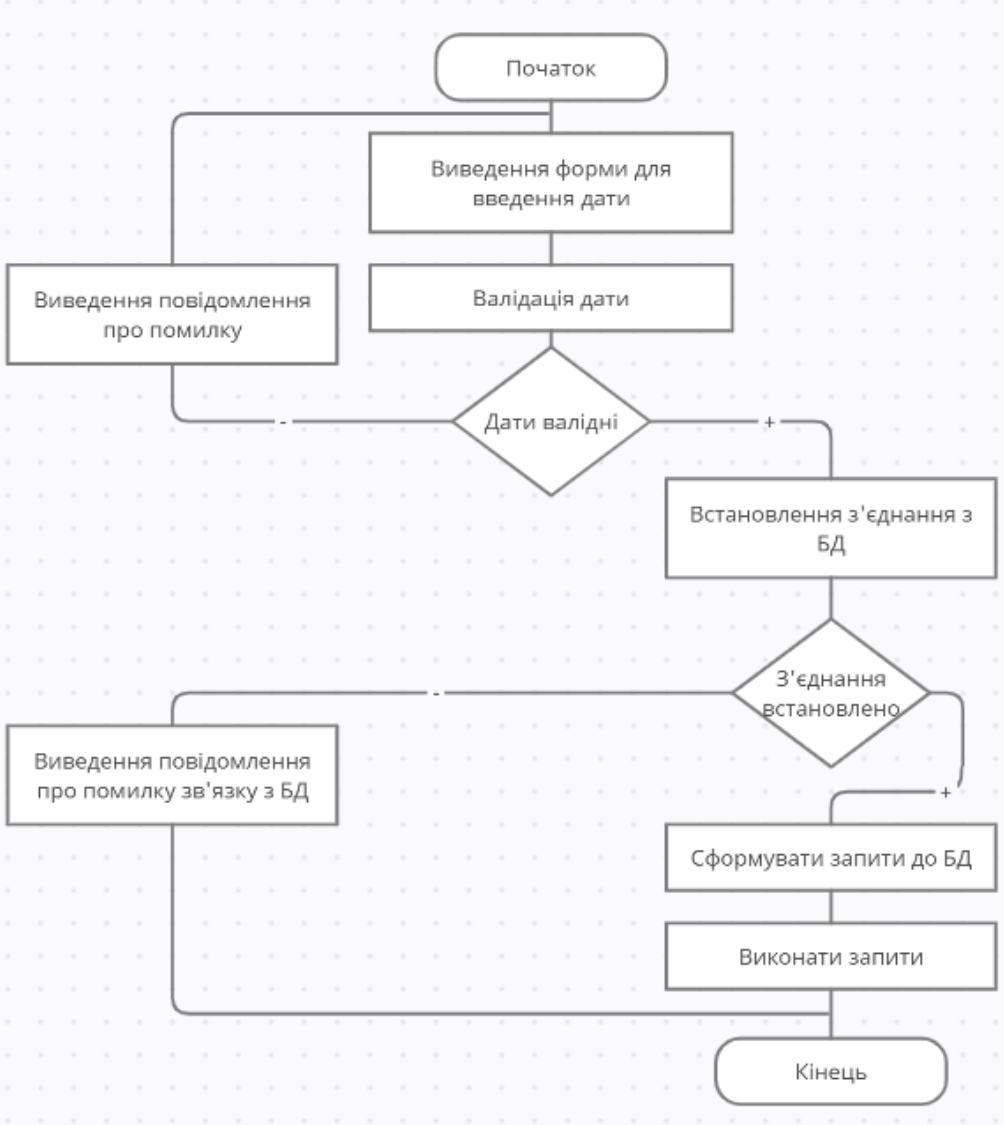



Рисунок 2.3 – Схема алгоритму виведення архіву

Даний алгоритм містить наступні кроки:

1. Виведення форми, в яку користувач вводить дату початку та дату кінця періоду, на який необхідно здійснити прогноз.

2. На даному епапі відбувається валідація дат.

3. Перевірка валідності дат. Якщо дати валідні – перехід до кроку 5.

4. Якщо дані не пройшли перевірку на валідність – виводиться повідомлення про помилку. Після цього користувач має змогу пройти заново алгоритм, перейшовши до його першого кроку.

5. Якщо дані валідні – відбувається встановлення зв’язку з базою даних.

6. Перевірка на встановлення зв’язку. При успішному з’єднанні з базою даних відбувається перехід до кроку 8.

7. Виведення повідомлення про відсутність зв’язку з базою даних, завершення діалогу з системою.

8. Формування запиту до бази даних.

9. Виконання запитів, завершення діалогу з системою.

На рисунку 2.4 зображено алгоритм запису інформації щодо поточної закупки квітів в архів.

Даний алгоритм містить наступні кроки:

1. Встановлення зв’язку з базою даних.

2. Введення нових даних.

3. Валідація введених даних.

4. Перевірка на валідність даних, якщо дані невалідні – перехід до 6.

5. Створення нового об’єкту архіву.

6. Додавання введених даних до бази даних.

7. Перевірка на успішність додавання даних. Якщо додавання успішне – завершення діалогу.

8. Виведення повідомлення про помилку. Завершення діалогу з системою.

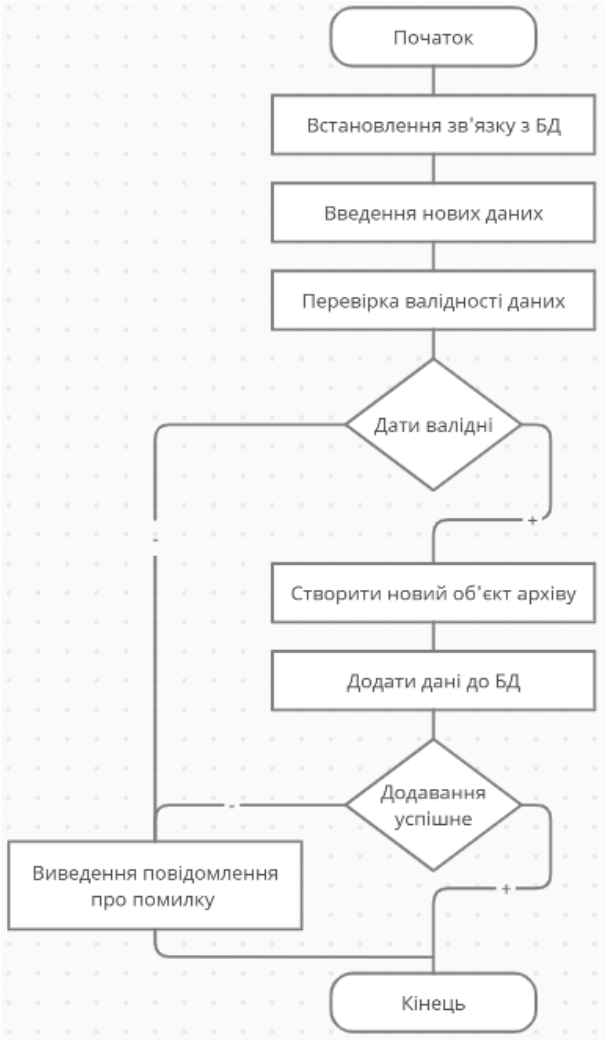



Рисунок 2.4 - Схема алгоритму запису інформації

## 2.4 Висновок

В другому розділі обґрунтовано використання методів прогнозування часових рядів, перевагами яких є те, що вони не накладають обмежень на довжину часового ряду, швидко адаптуються до динаміки змін часового ряду, не вимагають складних математичних розрахунків. Розроблено структуру програмного модуля прогнозування обсягу закупівлі квітів. Розроблено алгоритми функціонування програмного модуля, а також наведено алгоритм виведення архіву та алгоритм запису інформації в базу даних.

# 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ОБСЯГУ ЗАКУПІВЛІ КВІТІВ

## 3.1 Обґрунтування вибору мови та середовища програмування

Розглянемо популярні мови програмування для реалізації програмного модуля прогнозування обсягу закупки квітів. Java і Python є двома найпопулярнішими мовами програмування сьогодні. Хоча Java була найпопулярнішою мовою програмування з моменту її випуску в 1995 році, Python також набирає популярності з кожним роком. Розглянемо більш детально кожну з них.

Мова програмування Java – це універсальна, узгоджена, суворо типізована, об'єктно-орієнтована мова на основі класів. Зазвичай вона компілюється в набір інструкцій байт-коду та двійковий формат, визначений у специфікації віртуальної машини Java [26].

Хорошими прикладами програмного забезпечення мовою Java є банківські системи – великі розподілені програми, які успішно працюють протягом десятиліть. Програма, написана 10 років тому, буде успішно працювати на новітній версії платформи Java (JVM – Java Virtual Machine), не вимагаючи великих ресурсів та інвестицій. Платформа Java спочатку була розроблена для роботи на сильно завантажених системах з мільйонами користувачів, і досі досягає успіху. Доказом цього є активне використання Java такими гігантами IT-індустрії, як Google, Twitter і Amazon.

Об’єктно орієнтована мова програмування Java надзвичайно портативна. Одна й та сама програма Java працюватиме однаково на будь-якому комп’ютері, незалежно від апаратних можливостей чи операційної системи, якщо в ній є інтерпретатор Java. Крім портативності, ще однією з вагомих переваг Java є набір функцій безпеки, що захищають ПК, на якому запущена програма Java, не лише від проблем, спричинених помилковим кодом, але й від шкідливих програм (наприклад, вірусів). Ви можете безпечно запускати аплет Java, завантажений з Інтернету, оскільки функції безпеки Java перешкоджають цим типам аплетів отримати доступ

до жорсткого диска ПК або мережевих підключень. Аплет, як правило, є невеликою програмою на Java, яка вбудована в сторінку HTML.

Мову Java можна називати як компільованою, так і інтерпретованою мовою, так як її вихідний код спершу компілюється у двійковий байт-код. Цей байт-код виконується на віртуальній машині Java (JVM), що частіше за все є програмним інтерпретатором. Використання скомпільованого байт-коду дозволяє інтерпретатору (віртуальній машині) бути невеликим і ефективним (і майже таким же швидким, як і процесор, який виконує власний скомпільований код). Крім того, цей байт-код надає Java переносимість: він працюватиме на будь-якій JVM, яка правильно реалізована, незалежно від конфігурації апаратного чи програмного забезпечення комп’ютера. Більшість веб-браузерів (наприклад, Microsoft Internet Explorer або Netscape Communicator) містять JVM для запуску Java-аплетів.

Наведемо переваги мови Java [18].

1. Простота. Java є простою мовою програмування, тому що її легко вивчити і легко зрозуміти. Його синтаксис заснований на C++ і використовує автоматичне збирання сміття; тому нам не потрібно видаляти з пам'яті об'єкти без посилання. Java також вилучила такі функції, як явні покажчики, перевантаження операторів тощо, що полегшує читання та запис.

2. Об'єктно-орієнтованість. Java використовує об'єктно-орієнтовану парадигму, що робить її більш практичною. Усе в Java є об’єктом, який піклується як про дані, так і про поведінку. Java використовує об’єктно-орієнтовані концепції, такі як об’єкт, клас, успадкування, інкапсуляція, поліморфізм та абстракція.

3. Безпечність. Java є безпечною мовою програмування, оскільки не використовує явних покажчиків. Крім того, програми Java запускаються в пісочниці віртуальної машини. JRE також надає завантажувач класів, який використовують для динамічного завантаження класу в JVM. Він відокремлює пакунки класів з локальної файлової системи від імпортованиих з мережі.

4. Надійність. Java є надійною мовою програмування, оскільки вона використовує потужне управління пам'яттю. Ми також можемо обробляти винятки за допомогою коду Java. Крім того, ми можемо використовувати перевірку типів,

щоб зробити наш код безпечнішим. Він не надає явних покажчиків, щоб програміст не міг отримати доступ до пам'яті безпосередньо з коду.

5. Незалежність від платформи. Java-код може працювати безпосередньо на кількох платформах, тобто нам не потрібно компілювати його щоразу. Це лише один раз, запустіть в будь-якому місці (WORA) мову, яку можна перетворити на байтовий код під час компіляції. Байт-код – це незалежний від платформи код, який може виконуватися на кількох платформах.

6. Багатопотоковість. Java використовує багатопотокове середовище, де більшу задачу можна перетворити на кілька потоків і виконати окремо. Основна перевага багатопоточності полягає в тому, що нам не потрібно надавати пам'ять кожному запущеному потоку.

Далі розглянемо мову Python. Python – це високорівнева, універсальна і дуже популярна мова програмування. Мова програмування Python використовується у веб-розробці, програмах машинного навчання, а також у всіх передових технологіях в індустрії програмного забезпечення [19]. Мова програмування Python дуже добре підходить для початківців, а також досвідчених програмістів з іншими мовами програмування, такими як C++ і Java.

На даний момент Python є найбільш широко використовуваною багатоцільовою мовою програмування високого рівня. Python дозволяє програмувати в об'єктно-орієнтованих і процедурних парадигмах. Програми на Python, як правило, менші, ніж інші мови програмування, такі як Java. Програмістам доводиться вводити відносно менше, а вимога до відступів мови робить їх читаними весь час. Мова Python використовується майже всіма технологічними гігантами, такими як Google, Amazon, Facebook, Instagram, Dropbox, Uber тощо. Найбільшою перевагою Python є величезна колекція стандартних бібліотек

Python використовується у двох основних сферах: наукових обчисленнях та науковій інформації. Він має широкий вибір бібліотек, ресурсів і ресурсів, корисних для дослідження.

Python має бути однією з найпростіших мов, оскільки він підкреслює читабельність коду. Його бібліотеки та база коду для читання допомагають

програмістам оновлювати й підтримувати інформацію про програмне забезпечення без додаткових зусиль і витрат на курси розробки програмного забезпечення.

Концепція машинного навчання включає в себе класифікацію кількох речей, таких як фінансові послуги, поведінка розпізнавання голосу і навіть вибір місць на Netflix. Python використовується для машинного навчання через свої зрозумілі бібліотеки та фреймворки машинного навчання PyTorch, які включають дві з найпопулярніших – TensorFlow і scikit learn для кластеризації та вибору моделі.

Python є досконалою мовою в індустрії інвестиційного банкінгу та фінансових хедж-фондів. Він не має вбудованої підтримки алгебри лінійної попередньої обробки, але натомість має досить простий спосіб написання коду. Аналітики хедж-фондів використовують найбільш зрозумілий спосіб представлення набору матриць і векторів у вигляді списків і вкладених списків.

Існує багато переваг використання Python як мови програмування, тому наведемо деякі з них [29]:

Прискорення роботи програми за допомогою спеціальних моделей виконання.

Підтримка модулів і пакетів

Модулі використовують Інтернет-протоколи та надають допомогу в супутніх технологіях

Заохочує модульність програми та повторне використання коду

Забезпечує підвищення ефективності продуктів або проектів

Підтримує функціональну та процедурну мову програмування

Відкритий вихідний код та його розповсюдження можна вільно поширювати

Сумісний з платформами, базою даних, продуктами та програмами

У порівнянні з Python, Java начебто стоїть між C++ і Python. Програми написані на Java зазвичай працюють швидше, ніж відповідні програми на Python, і повільніше, ніж C++. Як і C++, Java виконує статичну перевірку типів, але Python ні. Розглянувши мови програмування Python та Java, можна зробити висновок, що завдяки перевагам Java саме її необхідно вибрати для розробки програмного модуля прогнозування обсягу закупки квітів. Тому доцільно розглянути середовище розробки мовою програмування Java.

Для написання програми мовою програмування Java необхідно розглянути деякі середовища, де розробник може реалізувати коди та програми. Для цього використовують інтегроване середовище розробки Java (Java IDE). Потреба в Java IDE виникла, коли у розробників виникли проблеми з кодуванням величезної програми.

Великі програми будуть мати багато класів і файлів, тому їх важко налаштувати. Використовуючи IDE можна підтримувати належне керування проектами. Він надає вказівки щодо завершення коду, синтаксичних помилок тощо.

Інтегроване середовище розробки (IDE) – це програмний застосунок, що дозволяє розробникам використовувати платформу з великою кількістю функцій та інструментів для розробки комп'ютерних програм, веб-сторінок, інструментів, служб тощо. Інструмент IDE включає текстові редактори, налагоджувачі, компілятори спільно з деякими функціями та інструментами, що допомагають автоматизувати, тестувати та аналізувати робочий процес розробки додатків.

З метою розробки серверних програм Java зазвичай використовують три середовища розробки, а саме IntelliJ IDEA, Eclipse і NetBeans.

IntelliJ IDEA – це середовище розробки для створення програмних застосунків за допомогою Java. IntelliJ IDEA розроблялася компанією JetBrains [30]. Він доступний у вигляді ліцензованої версії спільноти Apache 2 і власної комерційної версії. Обидва середовища можна використовувати для прогнозування обсягу закупки квітів. Однак саме IntelliJ IDEA було вибрано для реалізації програмного модуля прогнозування обсягу закупки квітів. Приклад роботи в IntelliJ IDEA наведено на рисунку 3.1.

IntelliJ IDEA надає рекомендації щодо завершення коду, аналізу коду та перевірених інструментів рефакторингу. Середовище має такі інструменти, як контроль версій, зв'язок з багатьма мовами і фреймворками. Воно може стежити за контекстом розробника і самостійно запускати потрібні інструменти.

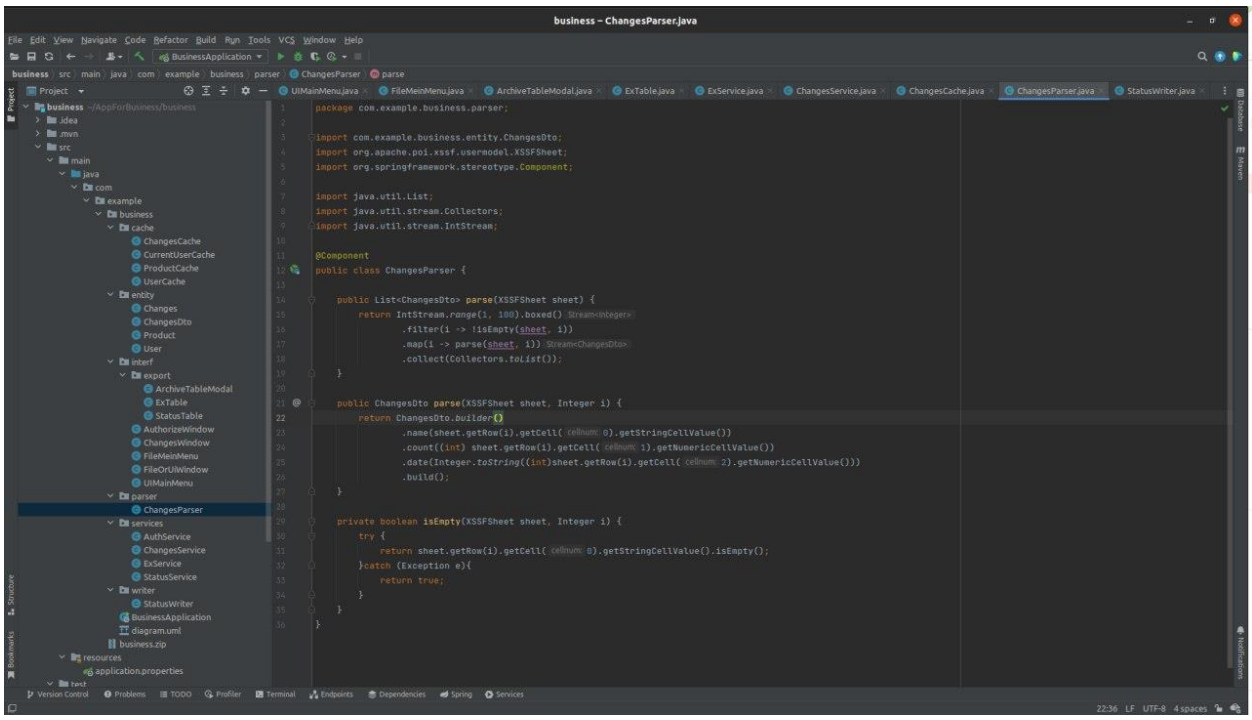



Рисунок 3.1 – Середовище розробки IntelliJ IDEA

Особливості IntelliJ IDEA [19]:

Розумний висновок: у ньому перераховані найбільш релевантні символи, які стосуються поточного контексту. Він безперервно переміщує новітні класи, методи тощо, збільшує список пропозицій. Відповідно, заповнення коду проводиться швидше.

Аналіз потоку даних: дане середовище може аналізувати потік даних та передбачати можливий символ при виконанні.

Мовне введення: можна легко інтегрувати фрагменти іншої мови у свій код Java, наприклад SQL.

IntelliJ рекомендує глибокий та ефективний рефакторинг, так як він має необхідні знання про використання символів.

Середовище пропонується з широким набором вбудованих інструментів, як наприклад GIT, контроль версій, декомпілятор, покриття, база даних SQL тощо.

Наявний потужний компілятор, що може виявляти дублікати, запах коду тощо.

Має надійну інтеграцію з серверами додатків.

Отже, завдяки наведеним перевагам, було вибрано мову програмування Java та середовище розробки IntelliJ IDEA.

### 3.2 Обґрунтування вибору бібліотек та фреймворків

Розробка програмного модуля прогнозування обсягу закупки квітів здійснюється мовою програмування Java. Розглянемо деякі бібліотеки, що використовуються в розробленому програмному модулі.

Swing – інструмент для створення графічного інтерфейсу користувача (GUI) на мові програмування Java. Це частина бібліотеки класів Java Foundation (JFC) [32].

Swing був розроблений, щоб надати більш функціональний набір програмних компонентів для створення графічного інтерфейсу користувача, ніж попередні інструменти AWT. Компоненти Swing підтримують специфічні модулі зовнішнього вигляду, які динамічно підключаються. Вони дозволяють емулювати графічний інтерфейс платформи (тобто компонент може бути динамічно підключений до іншого, специфічного для цього типу та поведінки операційної системи). Вагомим недоліком таких компонентів є відносно повільна робота, хоча це нещодавно не було підтверджено через зростаючу потужність персональних комп'ютерів. Перевага - універсальність інтерфейсу створюваних програм на всіх платформах.

У наведеному нижче коді наведено приклад використання Swing, що відображає вікно початкової активності – головне вікно програми, яке містить кнопки «Показати архів», «Додати інформацію», «Прогнозування».

```
package java1.interf;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
public class MainMenu {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;

    public MainMenu() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Головне меню");
        frame.setSize(400, 350);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);


        JButton exFlowers = new JButton();
        exFlowers.setText("Показати архів");
        exFlowers.setFont(font1);
        label.add(exFlowers, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
```

```
        exFlowers.addActionListener(new exFlowersListener());

        JButton newFlowers = new JButton();
        newFlowers.setText("Додати інформацію");
        newFlowers.setFont(font1);
        label.add(newFlowers, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
        newFlowers.addActionListener(new newFlowersListener());

        JButton newForecast = new JButton();
        newForecast.setText("Прогнозування");
        newForecast.setFont(font1);
        label.add(newForecast, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
        newForecast.addActionListener(new newForecastListener());

        frame.add(label);
        frame.setVisible(true);
    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);

    }
```

Java DataBase Connectivity (JDBC) – це інтерфейс програмування Java, який визначає методи, за допомогою яких програмне забезпечення на Java отримує доступ до бази даних. JDBC – це незалежний від платформи галузевий стандарт для

взаємодії програм Java з різними базами даних, реалізований у формі пакета java.sql, який є частиною Java SE [33].

Перевагами JDBC є:

- простота розробки: розробник може не знати особливостей бази даних, з якою працює;

- при переході компанії до іншої бази даних код не змінюється;

- немає необхідності встановлювати громіздку клієнтську програму;

- можливе підключитися до будь-якої бази даних за допомогою легко описаної URL-адреси.

- Java DataBase Connectivity – це стандартний API для самостійного підключення мови програмування Java до різних баз даних [20].

JDBC вирішує такі завдання:

- створення підключення до бази даних;

- створення виразів SQL;

- виконання запитів SQL;

- перегляд та модифікація отриманих записів.

Загалом, JDBC – це бібліотека, яка забезпечує цілий набір інтерфейсів для доступу до різних баз даних.

Для доступу до кожної конкретної бази даних потрібен спеціальний драйвер JDBC, який є адаптером програми Java до бази даних.

Бібліотека JDBC, що забезпечує цілий набір інтерфейсів для доступу до різних баз даних використовується в наступному фрагменті коду:

```
package java1.db;

import java.sql.*;
import java.util.ArrayList;

public class FlowerDAO {
```

```
        private static final String mySQL = "INSERT INTO flowers (dateS, rrl, rrm, rye, rw, rr, ro, ow, flowers.or, oy, ch) " +
                "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        private static final String myGetSQL = "SELECT * FROM flowers where dateS > ? and dateS < ?";

        public void addNewFl(db.Flower flower){
            PreparedStatement pstmt = null;
            ResultSet rs = null;
            Connection con = null;
            DBManager dbManager = DBManager.getInstance();
            try {
                con = dbManager.getConnection();
                pstmt = con.prepareStatement(mySQL, Statement.RETURN_GENERATED_KEYS);
                pstmt.setString(1, flower.getDate());
                pstmt.setInt(2, flower.getRoseRedLong());
                pstmt.setInt(3, flower.getRoseRedMini());
                pstmt.setInt(4, flower.getRoseYellow());
                pstmt.setInt(5, flower.getRoseWhite());
                pstmt.setInt(6, flower.getRosePink());
                pstmt.setInt(7, flower.getRoseOrange());
                pstmt.setInt(8, flower.getOrchidWhite());
                pstmt.setInt(9, flower.getOrchidPink());
                pstmt.setInt(10, flower.getOrchidYellow());
                pstmt.setInt(11, flower.getChrysanthemums());
                System.out.println(pstmt);
                pstmt.executeUpdate();
                rs = pstmt.getGeneratedKeys();
                if (rs.next()) {
```

```
                flower.setId(rs.getInt(1));
            }
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            close(rs);
            close(pstmt);
            close(con);
        }
    }
```

### 3.3 Опис процесу написання програмного коду

Розробимо діаграму діяльності програмного модуля прогнозування обсягу закупівлі квітів при введенні нових даних.

Наведений нижче код описує зображену на рисунку 3.2 діаграму:

```
JButton newForecast = new JButton();
        newForecast.setText("Прогнозування");
        newForecast.setFont(font1);
        label.add(newForecast, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
        newForecast.addActionListener(new newForecastListener());
        frame.add(label);
        frame.setVisible(true);
    }
    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);
```

```
    }
    private class newFlowersListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageForNewFlowers().createWindow();
        }
    }
    private class exFlowersListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageShowOldFlower().createWindow();
        }
    }
    private class newForecastListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageForNewForecast().createWindow();
        }
}
```

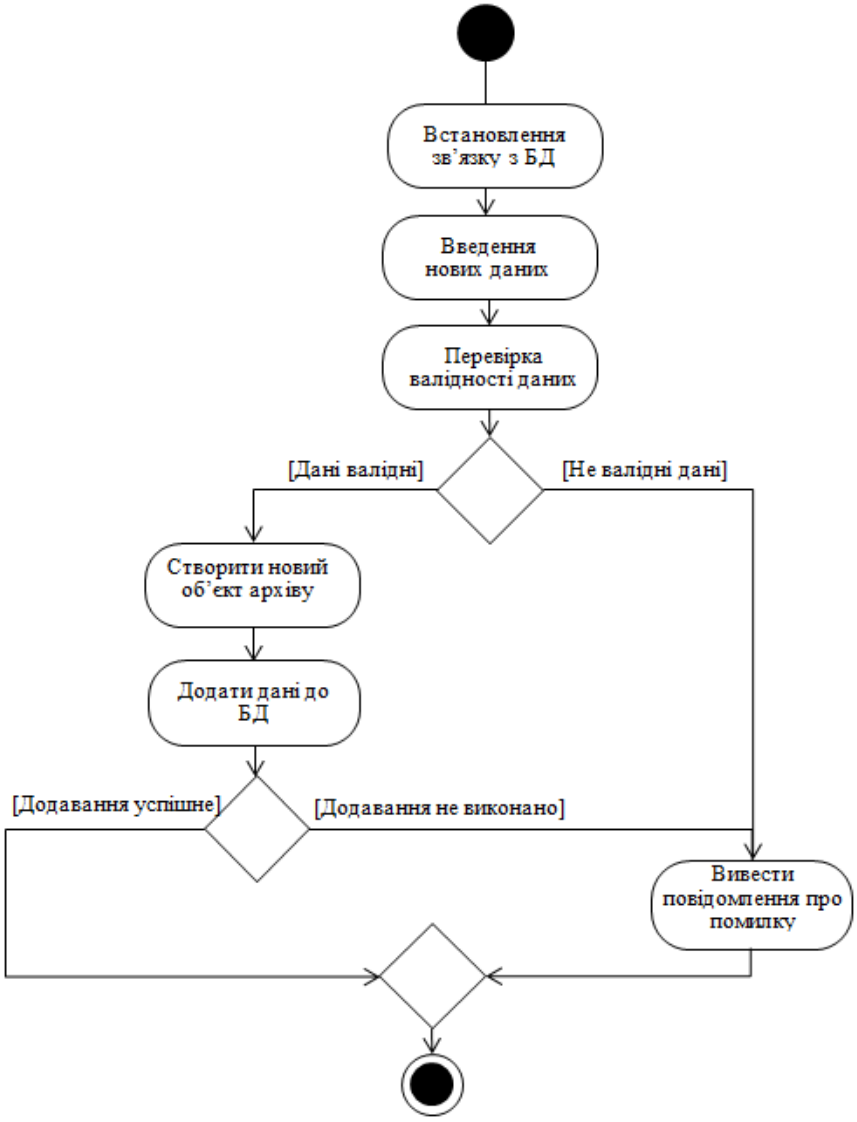



Рисунок 3.2 – Діаграма діяльності системи при введенні нових даних

Діаграма діяльності програмного модуля прогнозуванні обсягу закупки квітів відображає алгоритм прогнозування закупки квітів. Після вибору користувачем роботи з прогнозуванням відбувається встановлення зв'язку з базою даних та отримання даних на період, вказаний у запиті користувача. При отриманні необхідних даних відбувається перехід до розрахунку прогнозованого результату.

Діаграма діяльності програмного модуля при прогнозуванні обсягу закупки квітів зображена на рисунку 3.3.

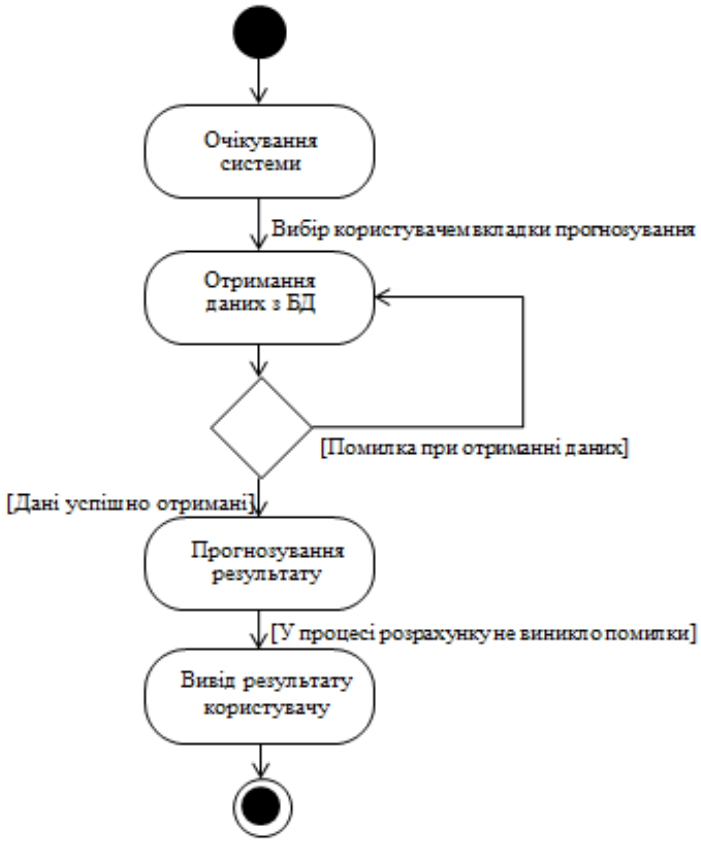



Рисунок 3.3 – Діаграма діяльності системи при прогнозуванні

В протилежному випадку відбувається перехід до попереднього стану. Якщо прогнозування пройшло успішно і при його обрахунку не виникло помилок, користувач отримує результат прогнозування обсягу закупки квітів на період часу, який зацікавив користувача.

## 3.4 Опис класів і функцій програмного модуля прогнозування обсягу закупки квітів

На рисунку 3.4 зображено загальну UML-діаграму класів програмного модуля прогнозування обсягу закупки квітів.

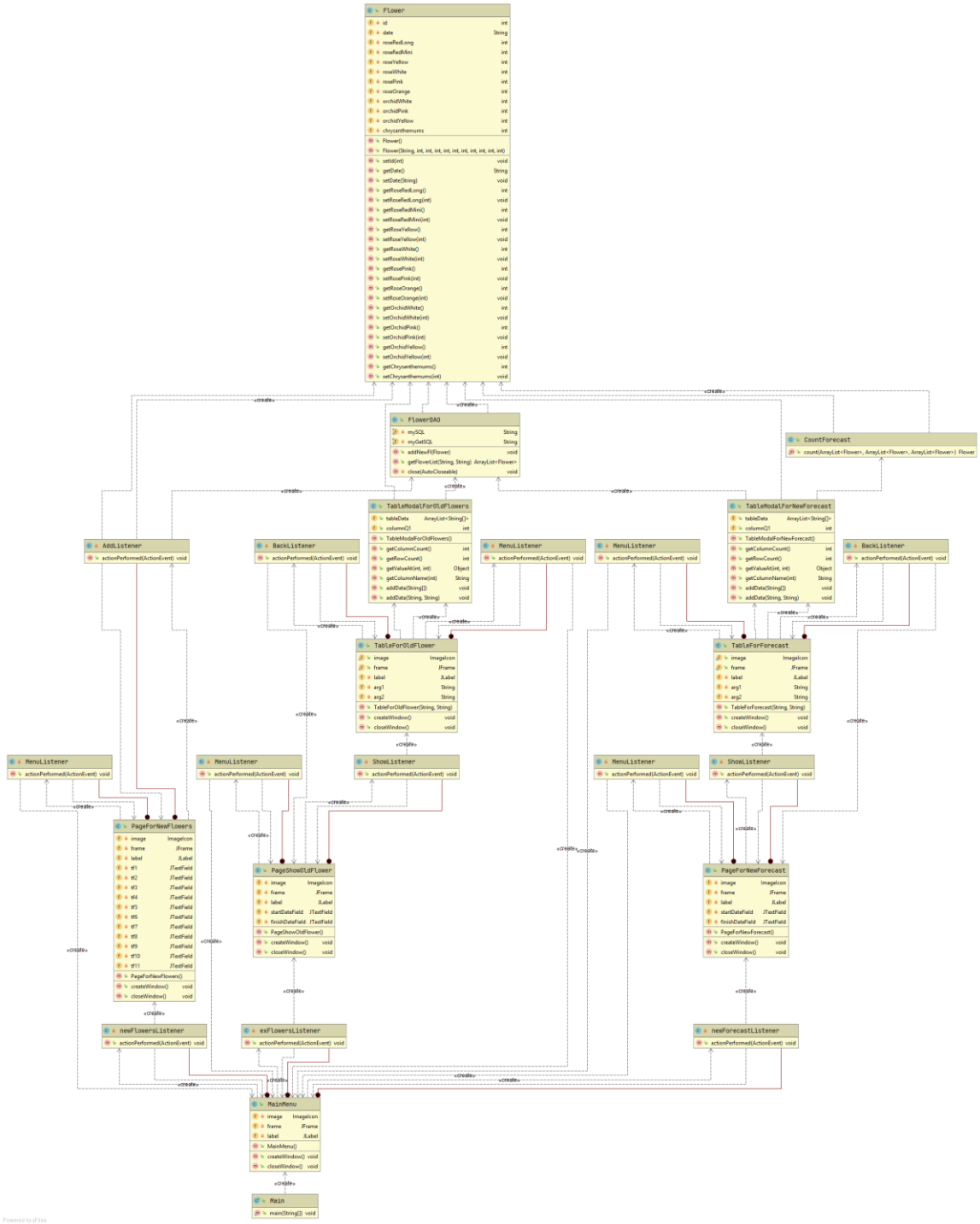



Рисунок 3.4 – Загальна UML-діаграма класів

Загальна UML-діаграма класів містить розроблені класи Flower, FlowerDAO, CountForecast, DBManager, MainMenu, PageForNewFlovers, PageForNewForecast, PageShowOldFlower, TableForForecast та інші.

Клас Flower призначений для отримання даних про квіти, що знаходяться в базі даних.

За допомогою класу FlowerDAO відбувається виконання запитів до бази даних.

Клас DBManager призначений для здійснення доступу до бази даних. В даному класі здійснюється підключитися до розробленої бази даних за допомогою легко описаної URL-адреси:

private static final String URL = "jdbc:mysql://localhost:3306/flowersdb";

Клас CountForecast – клас з основною логікою програми, в ньому відбувається прогнозування обсягу закупки квітів на основі методу аналізу часових рядів.

Класи PageForNewFlovers, PageForNewForecast, PageShowOldFlower, TableForForecast відповідають за відображення графічного інтерфейсу програми з допомогою інструменту для створення графічного інтерфейсу користувача Swing. Кожен клас відповідає за відображення та функціонування окрмих вікон.

### 3.5 Тестування програмного модуля прогнозування обсягу закупівлі квітів

Провівши 100 запусків додатку, було протестовано можливості його роботи, здійснено оцінку функціональних можливостей. Тестування програмного модуля прогнозування обсягу закупівлі квітів підтвердило коректність його роботи.

Початок роботи з додатком починається з відкриття головного меню програми. Дане меню включає наступні пункти – «Показати архів», «Додати інформацію», «Прогнозування» (рис. 3.5). При натисканні на кожен з пуктів відбувається перехід до наступного діалогового вікна. Отже, на першому кроці необхідно натистути кнопку у вікні початкової активності, що відповідає пункту меню, який цікавить користувача.

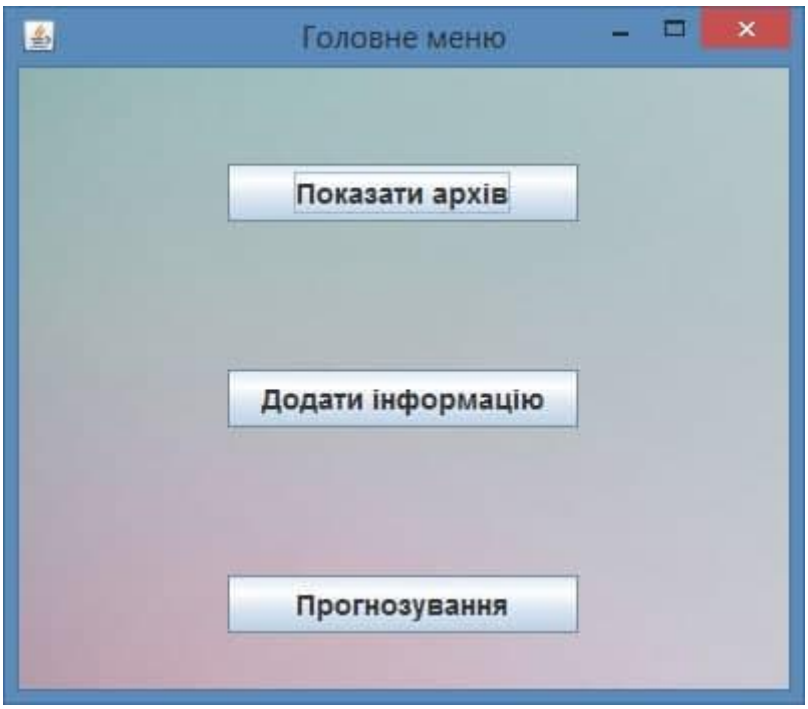



Рисунок 3.5 – Вікно головного меню

Якщо користувач вибрав перший пункт головного меню програми – «Показати архів» – відбувається перехід до вікна архіву. Дане вікно містить поля для введення початкової та кінцевої дати, тобто конкретний проміжок часу, за який було здійснено закупку квітів, що цікавить користувача в даний момент (рис. 3.6).

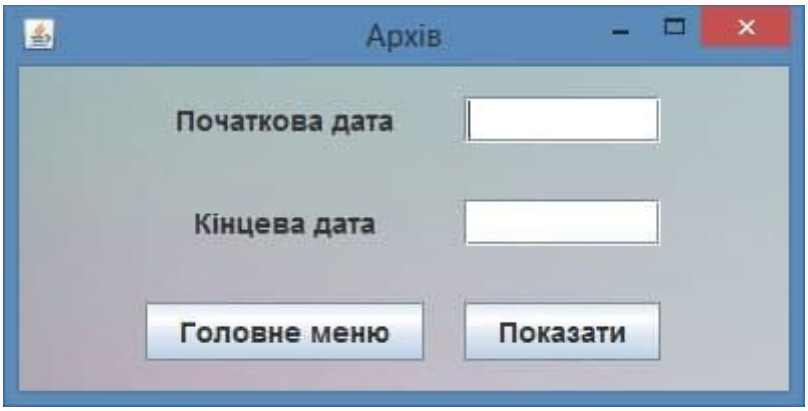
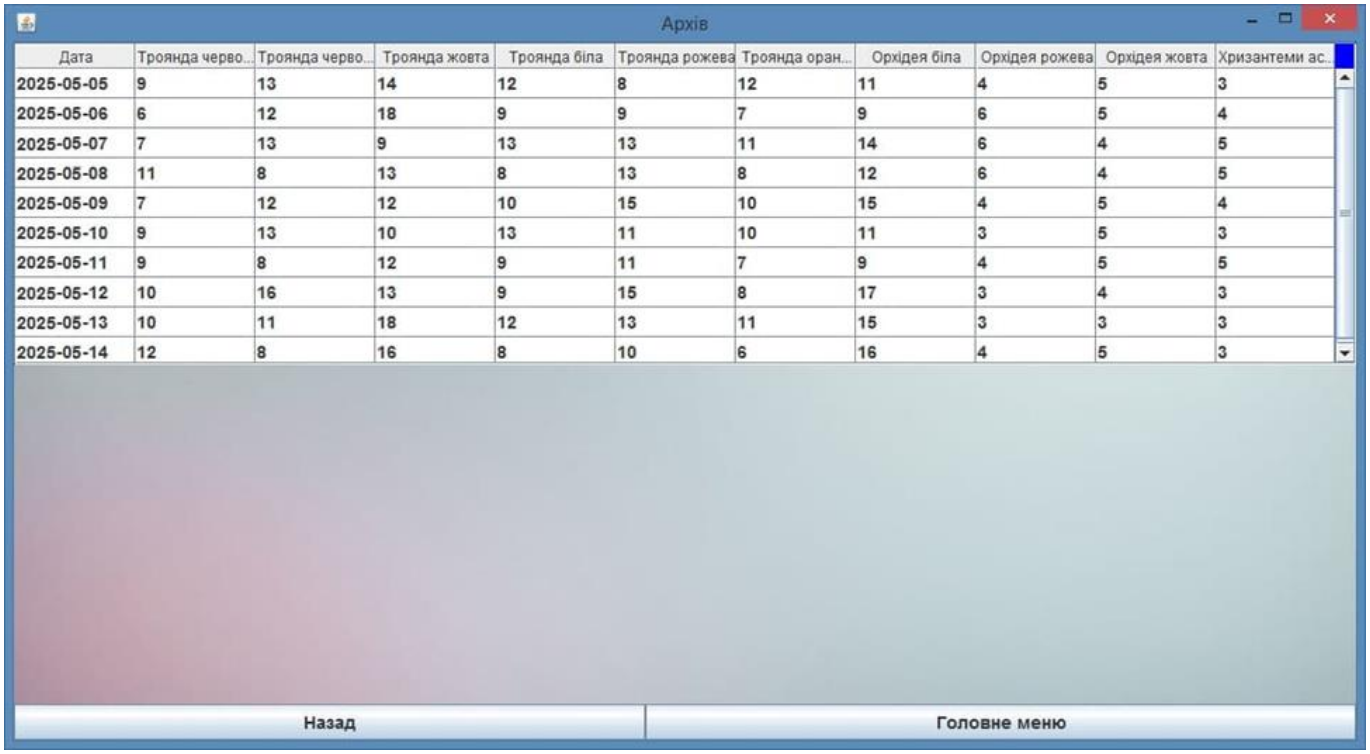



Рисунок 3.6 – Вікно введення проміжку часу

Дане вікно містить також кнопки «Головне меню» та «Показати». Натискання кнопки «Головне меню» приводить до попереднього вікна з вибором пунктів для подальшої роботи. При введенні необхідного проміжку часу та натискання кнопки «Показати» відбувається перехід до вікна архіву, де міститься таблиця з закупками

квітів на вказаний користувачем проміжок часу (рис. 3.7). Таблиця формується шляхом імпорту значень з бази даних.




| Дата | Троянда черво... | Троянда черво... | Троянда жовта | Троянда біла | Троянда рожева | Троянда оран... | Орхідея біла | Орхідея рожева | Орхідея жовта | Хризантеми ас... |
|---|---|---|---|---|---|---|---|---|---|---|
| 2025-05-05 | 9 | 13 | 14 | 12 | 8 | 12 | 11 | 4 | 5 | 3 |
| 2025-05-06 | 6 | 12 | 18 | 9 | 9 | 7 | 9 | 6 | 5 | 4 |
| 2025-05-07 | 7 | 13 | 9 | 13 | 13 | 11 | 14 | 6 | 4 | 5 |
| 2025-05-08 | 11 | 8 | 13 | 8 | 13 | 8 | 12 | 6 | 4 | 5 |
| 2025-05-09 | 7 | 12 | 12 | 10 | 15 | 10 | 15 | 4 | 5 | 4 |
| 2025-05-10 | 9 | 13 | 10 | 13 | 11 | 10 | 11 | 3 | 5 | 3 |
| 2025-05-11 | 9 | 8 | 12 | 9 | 11 | 7 | 9 | 4 | 5 | 5 |
| 2025-05-12 | 10 | 16 | 13 | 9 | 15 | 8 | 17 | 3 | 4 | 3 |
| 2025-05-13 | 10 | 11 | 18 | 12 | 13 | 11 | 15 | 3 | 3 | 3 |
| 2025-05-14 | 12 | 8 | 16 | 8 | 10 | 6 | 16 | 4 | 5 | 3 |

Рисунок 3.7 – Вікно архіву

Вікно архіву містить таблицю з атрибутами «Дата» та назвами квітів, що закуповує магазин. В таблицю виводяться дати, коли здійснювалась закупка квітів обраного періоду, а також кількість квітів різних видів, що були закуплені відповідного дня.

У розглянутому вікні присутні кнопки, що забезпечують перехід до попередніх вікон. Завдяки кнопці «Назад» відбувається перехід до вінка введення періоду часу, необхідного для перегляду архівних даних. Кнопка «Головне меню» повертає користувача до вибору пунктів головного меню та подальшої роботи з ними.

При виборі пункту головного меню «Додати інформацію» виводиться вікно для введення поточної закупівлі, що має бути збережена в базу даних (рис. 3.8).

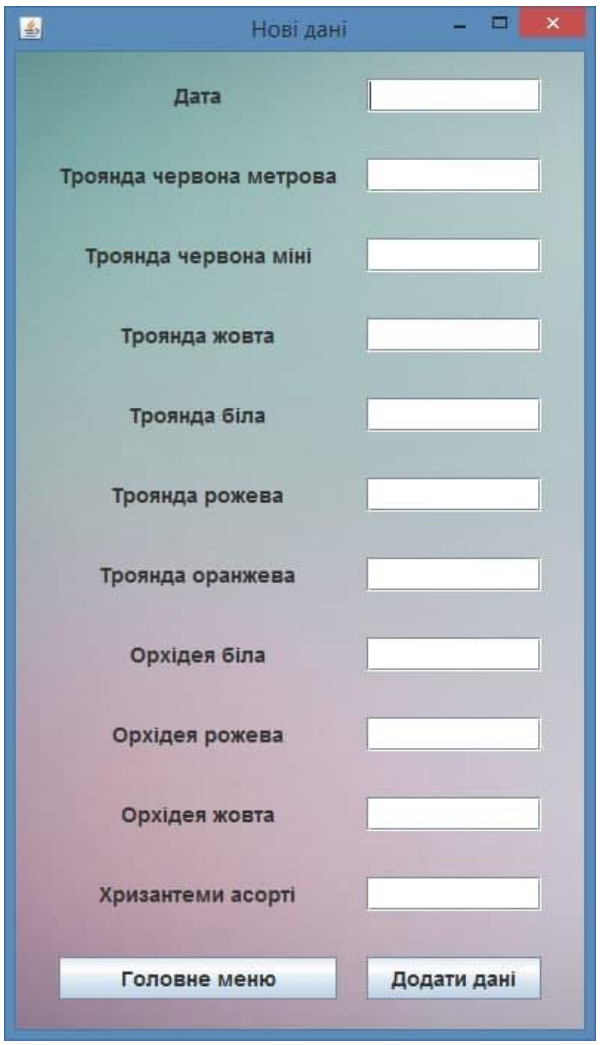



Рисунок 3.8 – Вікно введення нових даних

Наведене на рисунку 3.8 вікно містить поля для введення дати закупівлі та кількості квітів відповідно до їх виду. Вікно містить кнопку «Додати дані», що забезпечує запис введених значень, заповнених у відповідні поля користувачем, для їх подальшого збереження в базі даних. З даного вікна також можливий перехід до вікна початкової активності – головного меню, що здійснюється за допомогою кнопки «Головне меню».

При виборі останнього пункту головного меню – «Прогнозування» – відбувається перехід до вікна прогнозування. Дане вікно містить поля для введення початкової та кінцевої дати, тобто конкретний проміжок часу, на який потрібно здійснити прогноз закупівлі квітів, що цікавить користувача в даний момент (рис. 3.9).

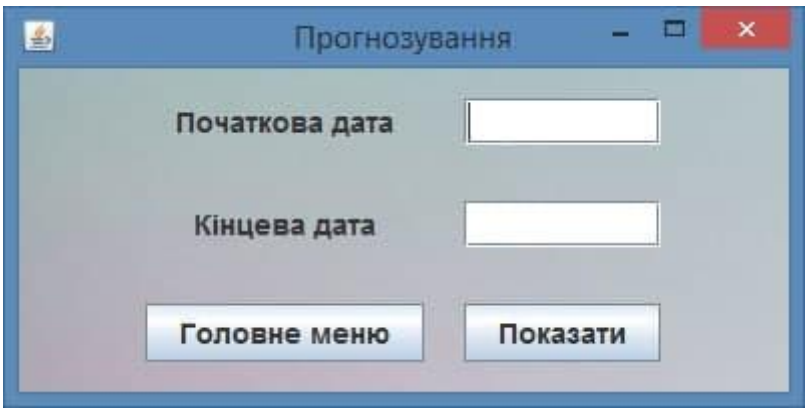



Рисунок 3.9 – Вікно введення проміжку часу для прогнозування

Дане вікно містить кнопку для повернення до вікна початкової активності, якщо користувач на попередньому кроці помилково вибрав пункт «Прогнозування».

Після введення початкової та кінцевої дати користувач натискає кнопку «Показати». Після цього програма аналізує дати на присутність сезонних чинників та здійснює прогнозування обсягу закупівлі сезонних квітів методом ковзного середнього. Для несезонних квітів прогнозування відбувається завдяки методу експотенціального згладжування. Після проведення програмним модулем необхідних обчислень відбувається перехід до вікна, що містить таблицю прогнозованих значень закупівлі квітів на обраний користувачем період часу (рис. 3.10).

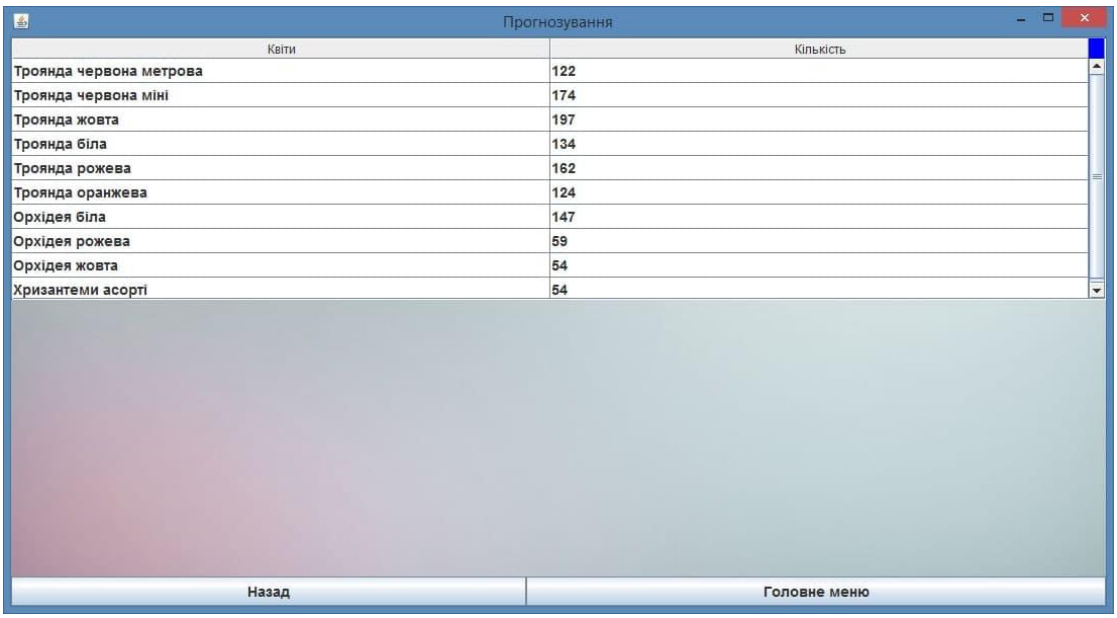



Рисунок 3.10 – Вікно виведення результатів прогнозування

Таблиця результатів прогнозування містить атрибути «Квіти» та «Кількість». В ній наведені прогнозована кількість квітів певного виду, що доцільно закупити на обраний період часу.

Проведемо тестування програмного забезпечення на відповідність прогнозованих результатів реальним значенням продажу, тобто оцінимо точність прогнозування. Для цього обиремо проміжок часу з 02 травня 2025 року до 11 травня 2025 року (рис. 3.11).

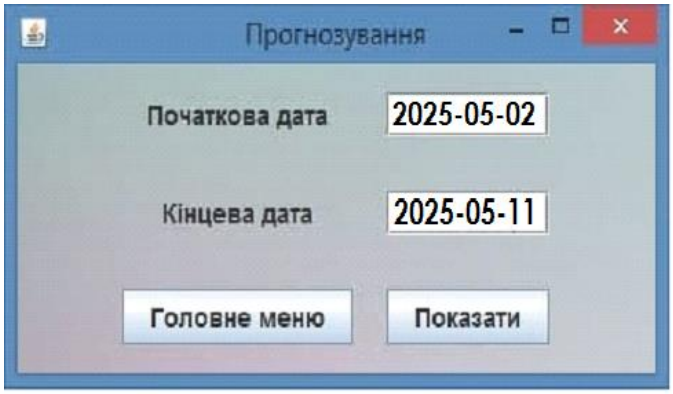



Рисунок 3.11 – Вікно введення проміжку часу для прогнозування

Після натискання кнопки «Показати» отримаємо результати прогнозування, обраховані за допомогою програмного модуля (рис. 3.12).

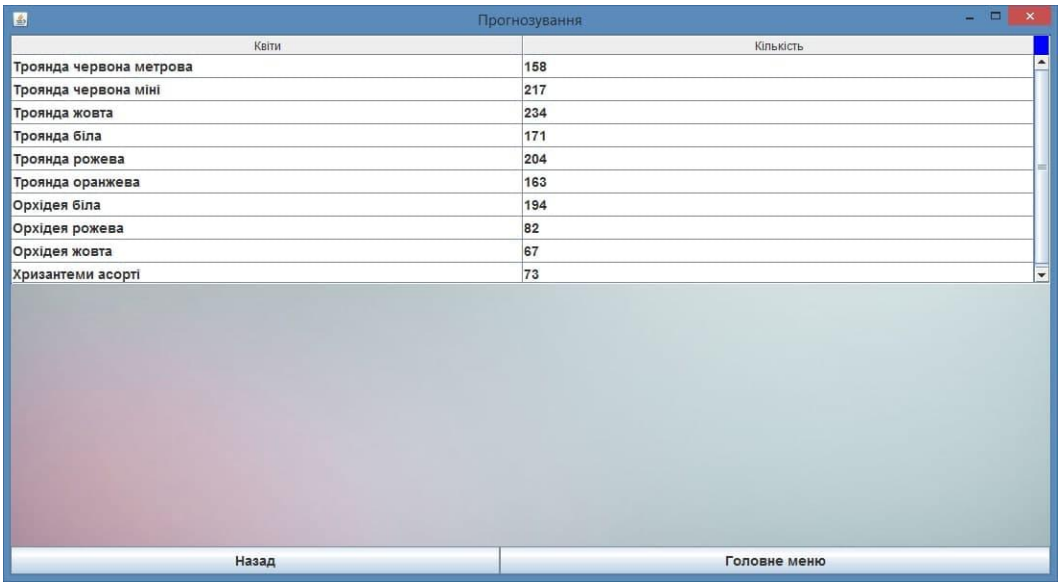



Рисунок 3.10 – Вікно виведення результатів прогнозування

В таблиці 3.1 наведемо реальні значення продажу квітів, здійснені в проміжок часу з 02 травня 2025 року до 11 травня 2025 року та прогнозовані значення розробленим додатком.

Таблиця 3.1 – Тестування розробленого додатку

| Квіти | Пронозована кількість | Реальна кількість | Похибка результату |
|---|---|---|---|
| Троянда червона метрова | 158 | 150 | 5% |
| Троянда червона міні | 217 | 205 | 5,6% |
| Троянда жовта | 234 | 200 | 15% |
| Троянда біла | 171 | 165 | 4% |
| Троянда рожева | 204 | 198 | 3% |
| Троянда оранжева | 163 | 152 | 7% |
| Орхідея біла | 194 | 180 | 7,3% |
| Орхідея рожева | 82 | 50 | 39% |
| Орхідея жовта | 67 | 60 | 10,5% |
| Хризантеми асорті | 73 | 70 | 4,2% |

З наведених в таблиці 3.1 даних видно, що похибка розрахунку в окремому випадку досягає 39%, однак, зважаючи на те, що в інших випадках вона не досягає 15%  можна зробити висновок, що програмний модуля прогнозування обсягу закупівлі квітів працює коректно та з достатньою точністю.

Аналогічні розрахунки були проведено за допомогою аналогів «AnaPlan», та «Streamline». В таблиці 3.2 наведено результати порівняння роботи розробленого додатку з наявними рішеннями.

Таблиця 3.2 – Порівняння результатів роботи програм

| Програма | Точність прогнозування |
|---|---|
| Розроблений додаток | 89,94% |
| AnaPlan | 86,83% |
| Streamline | 83,28% |

Таким чином, із таблиці 3.2 видно, що розроблений модуль має на 3,11% кращу точність прогнозування ніж програми-аналоги, а отже мета роботи досягнута.

### 3.6 Висновок

В третьому розділі було здійснено обґрунтуваня вибору мови програмування для реалізації програмного модуля прогнозування обсягу закупівлі квітів. Програмний модуль розроблено на об’єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність та простоту розробки, в середовищі розробки IntelliJ IDEA, що має переваги у вигляді глибокого розуміння коду, задоволенні усіх вимог розробника, розумного доповнення коду.

Розроблено та описано діаграму класів, розглянуто методи та атрибути розроблених класів. Проведено тестування програмного модуля прогнозування обсягу закупівлі квітів, яке довело, що розроблене програмне забезпечення має на 3,11% кращу точність прогнозування ніж програми-аналоги.

# ВИСНОВКИ

Всі завдання, поставлені в бакалаврській кваліфікаційній роботі були виконані в повному обсязі. Обґрунтувано доцільність розробки програмного модуля прогнозування обсягу закупівлі квітів, розглянуто принципи прогнозування та планування обсягу закупки квітів для квіткових магазинів. Проведено аналіз сучасних програм-аналогів, які використовуються для прогнозування закупки і продажу товарів. Обґрунтовано використання методу прогнозування часових рядів, перевагами яких є те, що вони не накладають обмежень на довжину часового ряду, швидко адаптуються до динаміки змін часового ряду, не вимагають складних математичних розрахунків. Розроблено структуру програмного модуля прогнозування обсягу закупівлі квітів. Розроблено алгоритми функціонування програмного модуля, а також наведено алгоритм виведення архіву та алгоритм запису інформації в базу даних.

Здійснено обґрунтуваня вибору мови програмування для реалізації програмного модуля прогнозування обсягу закупівлі квітів. Програмний модуль розроблено на об’єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність та простоту розробки, в середовищі розробки IntelliJ IDEA.

Розроблено та описано діаграму класів, розглянуто методи та атрибути розроблених класів. Проведено тестування програмного модуля прогнозування обсягу закупівлі квітів, яке довело, що розроблене програмне забезпечення має на 3,11% кращу точність прогнозування ніж програми-аналоги.

Отже всі поставлені задачі виконано, мету роботи досягнуто.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сімончук С.В., Озеранський В.С., Моклюк Д.М. Прогнозування обсягу закупівлі квітів. Матеріали Міжнародної науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» − [Електронний ресурс]. https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/25527/21144

2. Савицька Н.Л. Управління попитом: навч.-метод посібник / Н.Л. Савицька, О.М. Прядко. - Х.: ХДУХТ, 2016. 197 с.

3. Антонова О. А. Ринок квітів та декоративних рослин в Україні. Проект. К.: Кристал, 2019. 26 с.

4. Лігоненко Л.О., Гуляєва Н.М., Гриню Н.А., Ситник Г.В., Докієнко Л.М. Фінанси підприємства: Підручник/Київський національний торговельно-економічний ун-т. – К., 2007. – 491 с.

5. Бідюк П. І., Половцев О. В. Аналіз та моделювання економічних процесів перехідного періоду. — Київ: НТУУ «КПІ», 1999. — 230 с.

6. Бідюк П. І, Гожий О. П. Ймовірнісно-статистичні методи моделювання і прогнозування : монографія. Миколаїв : ЧДУ ім. П. Могили, 2014, 440 с.

7. Бідюк П.І., Савенков О.І., Баклан І.В. Часові ряди: моделювання і прогнозування : монографія. Київ: ЕКМО, 2003. 144 с.

8. Лук’яненко І. Г., Городніченко Ю. О. Сучасні економічні методи у фінансах : навч. посіб. Київ : Літера ЛТД, 2002. 352 с.

9. Томашевський О. В., Рисіков В. П. Комп’ютерні технології статистичної обробки даних : навч. посіб. Запоріжжя : ЗНТУ, 2015. 175 с.

10. Цеслів О. В. Коломієць А. С. Технологія проектування та адміністрування баз даних та сховищ даних : навч. посіб. Київ : КПІ ім. І. Сікорського, 2017. 290 с.

11. Streamline – [Електронний ресурс]. – Режим доступу: https://www.streamlineplan.com/

12. Anaplan – [Електронний ресурс]. – Режим доступу:

https://www.anaplan.com/

13. o9 Solutions Digital Brain Platform – [Електронний ресурс]. – Режим доступу: www.o9solutions.com.

14. Kress G.J., Shyder J. Forecasting and Market Analysis Techniques: A Practical Approach. — Hardcover, 1994.

15. Wheelwright, S. and Makridakis, S. Forecasting Methods for Management. — 4th ed. — John Wiley & Sons, Canada, 1985.

16. Кадомський К. К., Ніколюк П. К. Java. Теорія і практика : навч. посіб. Вінниця : Донну, 2019. 197 с.

17. Копитко М. Ф., Іванків К. С. Основи програмування мовою Java : тексти лекцій. Львів : ЛНУ ім. Івана Франка, 2002. 83с.

18. Яковенко А. В. Основи програмування. Python. Частина 1 : навч. посіб. Київ : КПІ ім. І. Сікорського, 2018. 195 с.

19. Юрченко І. В., Сікора В. С. Програмування мовою Python : навч. посіб. Чернівці : ЧНУ, 2022. 104 с.

20. Java-програмування: комп’ютерний практикум : навч. посіб. Київ: КПІ ім. І. Сікорського, 2021. 95 с.

21. Java Swing Project: Properties, Advantages & Frameworks. URL: https://www.upgrad.com/blog/java-swing-project/ (дата звернення: 29.04.2025).

22. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп’ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. – Вінниця : ВНТУ, 2023. – (PDF, 54 с.).

## ДОДАТОК А (обов'язковий)
## ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Програмний модуль прогнозування обсягу закупівлі квітів

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра комп'ютерних наук
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 18,76 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

✓ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Яровий А.А., зав. каф. КН _____
(прізвище, ініціали, посада) (підпис)

Колесницький О.К., проф. каф КН _____
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку _____ Озеранський В.С.
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Озеранський В.С.
(підпис) (прізвище, ініціали, посада)

Здобувач _____ Моклюк Д.М.
(підпис) (прізвище, ініціали)

# ДОДАТОК Б (довідниковий)
# Інструкція користувача

Початок роботи з додатком починається з відкриття головного меню програми. Дане меню включає наступні пункти – «Показати архів», «Додати інформацію», «Прогнозування» (рис. Б.1). При натисканні на кожен з пуктів відбувається перехід до наступного діалогового вікна. Отже, на першому кроці необхідно натистути кнопку у вікні початкової активності, що відповідає пункту меню, який цікавить користувача.

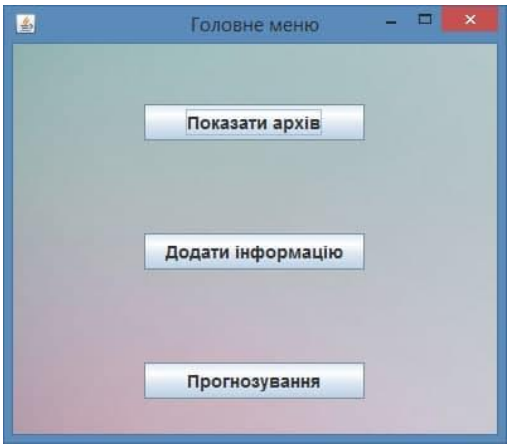



Рисунок Б.1 – Вікно головного меню

Якщо користувач вибрав перший пункт головного меню програми – «Показати архів» – відбувається перехід до вікна архіву. Дане вікно містить поля для введення початкової та кінцевої дати, тобто конкретний проміжок часу, за який було здійснено закупку квітів, що цікавить користувача в даний момент (рис. Б.2).

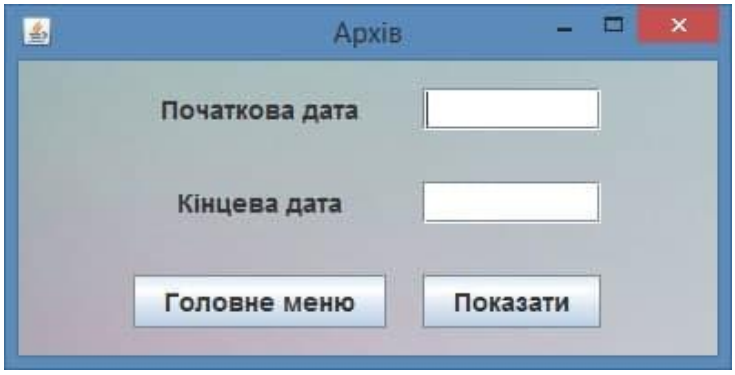



Рисунок Б.2 – Вікно введення проміжку часу

Дане вікно містить також кнопки «Головне меню» та «Показати». Натискання кнопки «Головне меню» приводить до попереднього вікна з вибором пунктів для подальшої роботи. При введенні необхідного проміжку часу та натискання кнопки «Показати» відбувається перехід до вікна архіву, де міститься таблиця з закупками квітів на вказаний користувачем проміжок часу (рис. Б.3). Таблиця формується шляхом імпорту значень з бази даних.

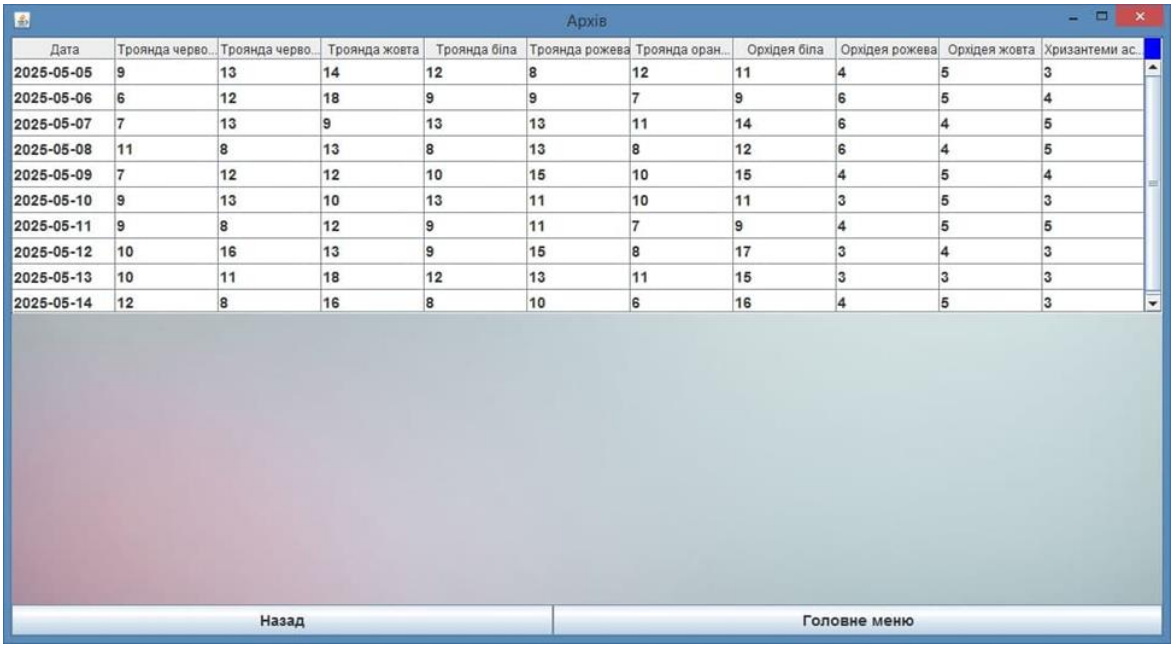



Рисунок Б.3 – Вікно архіву

Вікно архіву містить таблицю з атрибутами «Дата» та назвами квітів, що закуповує магазин. В таблицю виводяться дати, коли здійснювалась закупка квітів обраного періоду, а також кількість квітів різних видів, що були закуплені відповідного дня.

У розглянутому вікні присутні кнопки, що забезпечують перехід до попередніх вікон. Завдяки кнопці «Назад» відбувається перехід до вінка введення періоду часу, необхідного для перегляду архівних даних. Кнопка «Головне меню» повертає користувача до вибору пунктів головного меню та подальшої роботи з ними.

При виборі пункту головного меню «Додати інформацію» виводиться вікно для введення поточної закупівлі, що має бути збережена в базу даних (рис. Б.4).

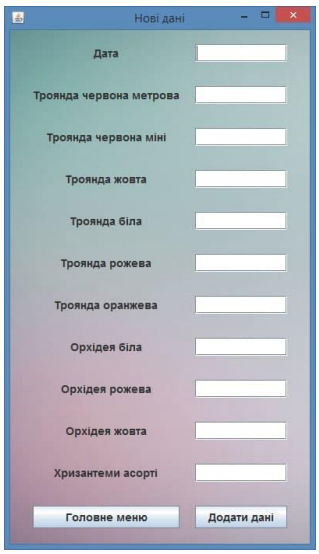



Рисунок Б.4 – Вікно введення нових даних

Вікно містить кнопку «Додати дані», що забезпечує запис введених значень, заповнених у відповідні поля користувачем, для їх подальшого збереження в базі даних. З даного вікна також можливий перехід до вікна початкової активності – головного меню, що здійснюється за допомогою кнопки «Головне меню».

При виборі останнього пункту головного меню – «Прогнозування» – відбувається перехід до вікна прогнозування. Дане вікно містить поля для введення початкової та кінцевої дати, тобто конкретний проміжок часу, на який потрібно здійснити прогноз закупівлі квітів, що цікавить користувача в даний момент (рис. Б.5).

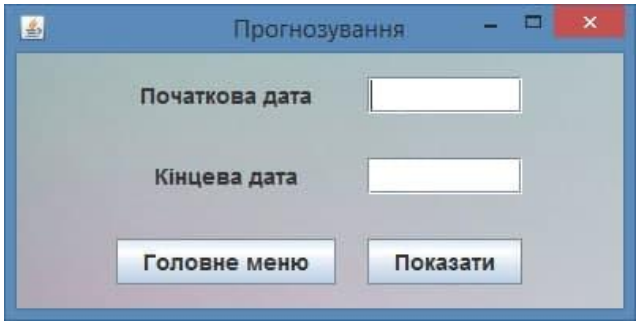



Рисунок Б.5 – Вікно введення проміжку часу для прогнозування

Дане вікно містить кнопку для повернення до вікна початкової активності, якщо користувач на попередньому кроці помилково вибрав пункт «Прогнозування».

Після введення початкової та кінцевої дати користувач натискає кнопку «Показати». Після цього програма аналізує дати на присутність сезонних чинників та здійснює прогнозування обсягу закупівлі сезонних квітів методом ковзного середнього. Для несезонних квітів прогнозування відбувається завдяки методу експотенціального згладжування. Після проведення програмним модулем необхідних обчислень відбувається перехід до вікна, що містить таблицю прогнозованих значень закупівлі квітів на обраний користувачем період часу (рис. Б.6).

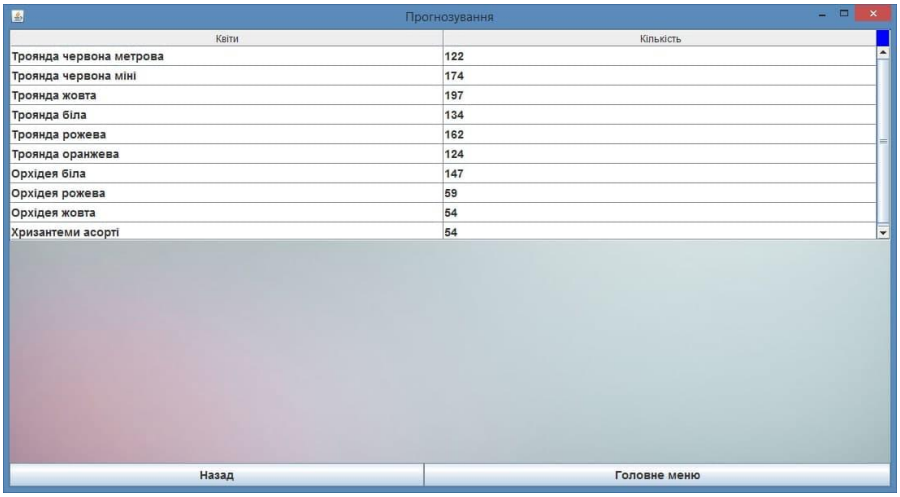



Рисунок Б.6 – Вікно виведення результатів прогнозування

Таблиця результатів прогнозування містить атрибути «Квіти» та «Кількість». В ній наведені прогнозована кількість квітів певного виду, що доцільно закупити на обраний період часу.

# ДОДАТОК В (обов'язковий)
# Лістинг програми

### Клас MainMenu

```
package java1.interf;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MainMenu {
   private ImageIcon image;
   private JFrame frame;
   private JLabel label;

   public MainMenu() {
      image = new ImageIcon(getClass().getResource("1.jpg"));
   }

   public void createWindow() {
      frame = new JFrame();
      frame.setTitle("Головне меню");
      frame.setSize(400, 350);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setLocationRelativeTo(null);

      Font font1 = new Font("TimesRoman", Font.BOLD, 14);
      label = new JLabel();
      label.setLayout(new GridBagLayout());
      label.setVisible(true);
      label.setBackground(Color.BLUE);
      label.setIcon(image);

      JButton exFlowers = new JButton();
      exFlowers.setText("Показати архів");
      exFlowers.setFont(font1);
      label.add(exFlowers, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
      exFlowers.addActionListener(new exFlowersListener());

      JButton newFlowers = new JButton();
      newFlowers.setText("Додати інформацію");
      newFlowers.setFont(font1);
      label.add(newFlowers, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
      newFlowers.addActionListener(new newFlowersListener());

      JButton newForecast = new JButton();
```

```
        newForecast.setText("Прогнозування");
        newForecast.setFont(font1);
        label.add(newForecast, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
        newForecast.addActionListener(new newForecastListener());

        frame.add(label);
        frame.setVisible(true);
    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);

    }

    private class newFlowersListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageForNewFlowers().createWindow();
        }
    }

    private class exFlowersListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageShowOldFlower().createWindow();
        }
    }

    private class newForecastListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageForNewForecast().createWindow();
        }
    }
}
```

**Клас Flower**

```
package db;

public class Flower {
    private int id;
    private String date;
```

```
    private int roseRedLong;
    private int roseRedMini;
    private int roseYellow;
    private int roseWhite;
    private int rosePink;
    private int roseOrange;
    private int orchidWhite;
    private int orchidPink;
    private int orchidYellow;
    private int chrysanthemums;

    public Flower() {
    }

    public Flower(String date, int roseRedLong, int roseRedMini, int roseYellow, int roseWhite, int
rosePink, int roseOrange, int orchidWhite, int orchidPink, int orchidYellow, int chrysanthemums) {
        this.date = date;
        this.roseRedLong = roseRedLong;
        this.roseRedMini = roseRedMini;
        this.roseYellow = roseYellow;
        this.roseWhite = roseWhite;
        this.rosePink = rosePink;
        this.roseOrange = roseOrange;
        this.orchidWhite = orchidWhite;
        this.orchidPink = orchidPink;
        this.orchidYellow = orchidYellow;
        this.chrysanthemums = chrysanthemums;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public int getRoseRedLong() {
        return roseRedLong;
    }

    public void setRoseRedLong(int roseRedLong) {
        this.roseRedLong = roseRedLong;
    }

    public int getRoseRedMini() {
```

```
    return roseRedMini;
}

public void setRoseRedMini(int roseRedMini) {
    this.roseRedMini = roseRedMini;
}

public int getRoseYellow() {
    return roseYellow;
}

public void setRoseYellow(int roseYellow) {
    this.roseYellow = roseYellow;
}

public int getRoseWhite() {
    return roseWhite;
}

public void setRoseWhite(int roseWhite) {
    this.roseWhite = roseWhite;
}

public int getRosePink() {
    return rosePink;
}

public void setRosePink(int rosePink) {
    this.rosePink = rosePink;
}

public int getRoseOrange() {
    return roseOrange;
}

public void setRoseOrange(int roseOrange) {
    this.roseOrange = roseOrange;
}

public int getOrchidWhite() {
    return orchidWhite;
}

public void setOrchidWhite(int orchidWhite) {
    this.orchidWhite = orchidWhite;
}

public int getOrchidPink() {
    return orchidPink;
}
```

```
    public void setOrchidPink(int orchidPink) {
        this.orchidPink = orchidPink;
    }

    public int getOrchidYellow() {
        return orchidYellow;
    }

    public void setOrchidYellow(int orchidYellow) {
        this.orchidYellow = orchidYellow;
    }

    public int getChrysanthemums() {
        return chrysanthemums;
    }

    public void setChrysanthemums(int chrysanthemums) {
        this.chrysanthemums = chrysanthemums;
    }
}
```

**Клас FlowerDAO**

```
package java1.db;

import java.sql.*;
import java.util.ArrayList;

public class FlowerDAO {

    private static final String mySQL = "INSERT INTO flowers (dateS, rrl, rrm, rye, rw, rr, ro, ow,
flowers.or, oy, ch) " +
            "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
    private static final String myGetSQL = "SELECT * FROM flowers where dateS > ? and dateS < ?";

    public void addNewFl(db.Flower flower){
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        Connection con = null;
        DBManager dbManager = DBManager.getInstance();
        try {
            con = dbManager.getConnection();
            pstmt = con.prepareStatement(mySQL, Statement.RETURN_GENERATED_KEYS);
            pstmt.setString(1, flower.getDate());
            pstmt.setInt(2, flower.getRoseRedLong());
            pstmt.setInt(3, flower.getRoseRedMini());
            pstmt.setInt(4, flower.getRoseYellow());
            pstmt.setInt(5, flower.getRoseWhite());
            pstmt.setInt(6, flower.getRosePink());
            pstmt.setInt(7, flower.getRoseOrange());
            pstmt.setInt(8, flower.getOrchidWhite());
```

```
            pstmt.setInt(9, flower.getOrchidPink());
            pstmt.setInt(10, flower.getOrchidYellow());
            pstmt.setInt(11, flower.getChrysanthemums());
            System.out.println(pstmt);
            pstmt.executeUpdate();
            rs = pstmt.getGeneratedKeys();
            if (rs.next()) {
                flower.setId(rs.getInt(1));
            }
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            close(rs);
            close(pstmt);
            close(con);
        }

    }

    public ArrayList<db.Flower> getFloverList(String arg1, String arg2){
        ArrayList<db.Flower> result = new ArrayList<>();
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        Connection con = null;
        try {
            con = DBManager.getInstance().getConnection();
            pstmt = con.prepareStatement(myGetSQL);
            pstmt.setString(1, arg1);
            pstmt.setString(2, arg2);
            System.out.println(pstmt);
            rs = pstmt.executeQuery();
            while (rs.next()) {
                db.Flower flower = new db.Flower();
                flower.setDate(rs.getString("dateS"));
                flower.setRoseRedLong(rs.getInt("rrl"));
                flower.setRoseRedMini(rs.getInt("rrm"));
                flower.setRoseYellow(rs.getInt("rye"));
                flower.setRoseWhite(rs.getInt("rw"));
                flower.setRosePink(rs.getInt("rr"));
                flower.setRoseOrange(rs.getInt("ro"));
                flower.setOrchidWhite(rs.getInt("ow"));
                flower.setOrchidPink(rs.getInt("or"));
                flower.setOrchidYellow(rs.getInt("oy"));
                flower.setChrysanthemums(rs.getInt("ch"));
                result.add(flower);
            }
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            close(rs);
```

```
            close(pstmt);
            close(con);
        }
        return result;
    }
    private void close(AutoCloseable forClose) {
        if (forClose != null) {
            try {
                forClose.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

**Клас DBManager**

```
package java1.db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBManager {

    private static DBManager instance;
    private static final String URL = "jdbc:mysql://localhost:3306/flowersdb";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "root";

    public static synchronized DBManager getInstance() {
        if (instance == null)
            instance = new DBManager();
        return instance;

    }

    private DBManager() {
    }


    public Connection getConnection() throws SQLException {
        Connection con = null;
        try {
            con = DriverManager.getConnection(URL, USERNAME, PASSWORD);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return con;
    }
```

```
    public void commitAndClose(Connection con) {
        try {
            if (con != null) {
                con.commit();
                con.close();
            }
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
}
```

**Клас PageForNewFlowers**

```
package java1.interf;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import db.Flower;
import java1.db.FlowerDAO;

public class PageForNewFlowers {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;

    private JTextField tf1, tf2, tf3, tf4, tf5, tf6, tf7, tf8, tf9, tf10, tf11;

    public PageForNewFlowers() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Нові дані");
        frame.setSize(400, 700);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);

        JLabel dateLabel = new JLabel();
```

```
dateLabel.setText("Дата");
dateLabel.setFont(font1);
label.add(dateLabel, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf1 = new JTextField();
tf1.setFont(font1);
label.add(tf1, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel = new JLabel();
roseLabel.setText("Троянда червона метрова");
roseLabel.setFont(font1);
label.add(roseLabel, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf2 = new JTextField();
tf2.setFont(font1);
label.add(tf2, new GridBagConstraints(1, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel1 = new JLabel();
roseLabel1.setText("Троянда червона міні");
roseLabel1.setFont(font1);
label.add(roseLabel1, new GridBagConstraints(0, 3, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf3 = new JTextField();
tf3.setFont(font1);
label.add(tf3, new GridBagConstraints(1, 3, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel2 = new JLabel();
roseLabel2.setText("Троянда жовта");
roseLabel2.setFont(font1);
label.add(roseLabel2, new GridBagConstraints(0, 4, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf4 = new JTextField();
tf4.setFont(font1);
label.add(tf4, new GridBagConstraints(1, 4, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel3 = new JLabel();
roseLabel3.setText("Троянда біла");
roseLabel3.setFont(font1);
label.add(roseLabel3, new GridBagConstraints(0, 5, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf5 = new JTextField();
```

```
tf5.setFont(font1);
label.add(tf5, new GridBagConstraints(1, 5, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel4 = new JLabel();
roseLabel4.setText("Троянда рожева");
roseLabel4.setFont(font1);
label.add(roseLabel4, new GridBagConstraints(0, 6, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf6 = new JTextField();
tf6.setFont(font1);
label.add(tf6, new GridBagConstraints(1, 6, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel5 = new JLabel();
roseLabel5.setText("Троянда оранжева");
roseLabel5.setFont(font1);
label.add(roseLabel5, new GridBagConstraints(0, 7, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf7 = new JTextField();
tf7.setFont(font1);
label.add(tf7, new GridBagConstraints(1, 7, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel6 = new JLabel();
roseLabel6.setText("Орхідея біла");
roseLabel6.setFont(font1);
label.add(roseLabel6, new GridBagConstraints(0, 8, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf8 = new JTextField();
tf8.setFont(font1);
label.add(tf8, new GridBagConstraints(1, 8, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel7 = new JLabel();
roseLabel7.setText("Орхідея рожева");
roseLabel7.setFont(font1);
label.add(roseLabel7, new GridBagConstraints(0, 9, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

tf9 = new JTextField();
tf9.setFont(font1);
label.add(tf9, new GridBagConstraints(1, 9, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

JLabel roseLabel8 = new JLabel();
roseLabel8.setText("Орхідея жовта");
```

```
    roseLabel8.setFont(font1);
    label.add(roseLabel8, new GridBagConstraints(0, 10, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

    tf10 = new JTextField();
    tf10.setFont(font1);
    label.add(tf10, new GridBagConstraints(1, 10, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

    JLabel roseLabel9 = new JLabel();
    roseLabel9.setText("Хризантеми асорті");
    roseLabel9.setFont(font1);
    label.add(roseLabel9, new GridBagConstraints(0, 11, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

    tf11 = new JTextField();
    tf11.setFont(font1);
    label.add(tf11, new GridBagConstraints(1, 11, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

    JButton menuBtn = new JButton();
    menuBtn.setText("Головне меню");
    menuBtn.setFont(font1);
    label.add(menuBtn, new GridBagConstraints(0, 12, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

    JButton addBtn = new JButton();
    addBtn.setText("Додати дані");
    addBtn.setFont(font1);
    label.add(addBtn, new GridBagConstraints(1, 12, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

    menuBtn.addActionListener(new MenuListener());
    addBtn.addActionListener(new AddListener());

    frame.add(label);
    frame.setVisible(true);

}

public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);

}

private class MenuListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
```

```
            closeWindow();
            new MainMenu().createWindow();

        }
    }

    private class AddListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();

            Flower flower = new Flower(
                    tf1.getText(),
                    Integer.parseInt(tf2.getText()),
                    Integer.parseInt(tf3.getText()),
                    Integer.parseInt(tf4.getText()),
                    Integer.parseInt(tf5.getText()),
                    Integer.parseInt(tf6.getText()),
                    Integer.parseInt(tf7.getText()),
                    Integer.parseInt(tf8.getText()),
                    Integer.parseInt(tf9.getText()),
                    Integer.parseInt(tf10.getText()),
                    Integer.parseInt(tf11.getText())
            );

            new FlowerDAO().addNewFl(flower);
        }
    }
}
```

**Клас PageShowOldFlower**

```
package java1.interf;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class PageShowOldFlower {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;

    private JTextField startDateField, finishDateField;


    public PageShowOldFlower() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }
```

```
public void createWindow() {
    frame = new JFrame();
    frame.setTitle("Архів");
    frame.setSize(400, 200);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLocationRelativeTo(null);

    Font font1 = new Font("TimesRoman", Font.BOLD, 14);
    label = new JLabel();
    label.setLayout(new GridBagLayout());
    label.setVisible(true);
    label.setBackground(Color.BLUE);
    label.setIcon(image);

    JLabel date1Label = new JLabel();
    date1Label.setText("Початкова дата");
    date1Label.setFont(font1);
    label.add(date1Label, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

    startDateField = new JTextField();
    startDateField.setFont(font1);
    label.add(startDateField, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

    JLabel date2Label = new JLabel();
    date2Label.setText("Кінцева дата");
    date2Label.setFont(font1);
    label.add(date2Label, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(10, 10, 10, 10), 0, 0));

    finishDateField = new JTextField();
    finishDateField.setFont(font1);
    label.add(finishDateField, new GridBagConstraints(1, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));


    JButton menuBtn = new JButton();
    menuBtn.setText("Головне меню");
    menuBtn.setFont(font1);
    label.add(menuBtn, new GridBagConstraints(0, 5, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

    JButton showBtn = new JButton();
    showBtn.setText("Показати");
    showBtn.setFont(font1);
    label.add(showBtn, new GridBagConstraints(1, 5, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
```

```
        menuBtn.addActionListener(new MenuListener());
        showBtn.addActionListener(new ShowListener());

        frame.add(label);
        frame.setVisible(true);

    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);

    }

    private class MenuListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new MainMenu().createWindow();
        }
    }

    private class ShowListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new TableForOldFlower(startDateField.getText(), finishDateField.getText()).createWindow();
        }
    }
}
```

**Клас TableForForecast**

```
package java1.interf;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class TableForForecast {
    public static ImageIcon image;
    public static JFrame frame;
    private JLabel label;
    private String arg1, arg2;

    public TableForForecast(String arg1, String arg2) {
        this.arg1 = arg1;
        this.arg2 = arg2;
```

```
    image = new ImageIcon(getClass().getResource("1.jpg"));
}

public void createWindow() {
    frame = new JFrame();
    frame.setTitle("Прогнозування");
    frame.setSize(1100, 600);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLocationRelativeTo(null);
    Font font = new Font("TimesRoman", Font.BOLD, 14);
    label = new JLabel();
    label.setLayout(new GridBagLayout());
    label.setVisible(true);
    label.setIcon(image);
    frame.add(label);
    frame.setVisible(true);

    TableModalForNewForecast tmq2 = new TableModalForNewForecast();
    JTable table = new JTable(tmq2);

    JScrollPane tableScrollPane = new JScrollPane(table);
    tmq2.addData(arg1, arg2);
    int shur = 0;
    if ((19 + tmq2.getRowCount() * 24) > 500) {
        shur = 500;
    } else {
        shur = 19 + tmq2.getRowCount() * 24;
    }
    tableScrollPane.setPreferredSize(new Dimension(1080, shur));
    System.out.println(tmq2.getRowCount());
    tableScrollPane.setBackground(Color.BLUE);
    table.setFont(font);
    table.setRowHeight(24);

    label.add(tableScrollPane, new GridBagConstraints(0, 0, 4, 1, 1, 1, GridBagConstraints.NORTH,
            GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    JButton backBtn = new JButton();
    backBtn.setFont(font);
    backBtn.setText("Назад");
    label.add(backBtn, new GridBagConstraints(0, 0, 2, 1, 1, 1, GridBagConstraints.SOUTH,
            GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    JButton menuBtn = new JButton();
    menuBtn.setFont(font);
    menuBtn.setText("Головне меню");
    label.add(menuBtn, new GridBagConstraints(2, 0, 2, 1, 1, 1, GridBagConstraints.SOUTH,
            GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    backBtn.addActionListener(new BackListener());
```

```
        menuBtn.addActionListener(new MenuListener());

    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);
    }
    private class BackListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageForNewForecast().createWindow();
        }
    }
    private class MenuListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new MainMenu().createWindow();
        }
    }
}
```

**Клас CountForecast**

```
package java1.interf;

import db.Flower;
import java.util.ArrayList;
public class CountForecast {

    public static db.Flower count(ArrayList<db.Flower> f2018, ArrayList<db.Flower> f2019,
ArrayList<db.Flower> f2020) {
        db.Flower flower = new Flower();
        for (Flower flow : f2018) {
            flower.setRoseRedLong(flower.getRoseRedLong() + flow.getRoseRedLong());
            flower.setRoseRedMini(flower.getRoseRedMini() + flow.getRoseRedMini());
            flower.setRoseYellow(flower.getRoseYellow() + flow.getRoseYellow());
            flower.setRoseWhite(flower.getRoseWhite() + flow.getRoseWhite());
            flower.setRosePink(flower.getRosePink() + flow.getRosePink());
            flower.setRoseOrange(flower.getRoseOrange() + flow.getRoseOrange());
            flower.setOrchidWhite(flower.getOrchidWhite() + flow.getOrchidWhite());
            flower.setOrchidPink(flower.getOrchidPink() + flow.getOrchidPink());
            flower.setOrchidYellow(flower.getOrchidYellow() + flow.getOrchidYellow());
            flower.setChrysanthemums(flower.getChrysanthemums() + flow.getChrysanthemums());
        }
        for (Flower flow : f2019) {
            flower.setRoseRedLong(flower.getRoseRedLong() + flow.getRoseRedLong()*2);
```

```
            flower.setRoseRedMini(flower.getRoseRedMini() + flow.getRoseRedMini()*2);
            flower.setRoseYellow(flower.getRoseYellow() + flow.getRoseYellow()*2);
            flower.setRoseWhite(flower.getRoseWhite() + flow.getRoseWhite()*2);
            flower.setRosePink(flower.getRosePink() + flow.getRosePink()*2);
            flower.setRoseOrange(flower.getRoseOrange() + flow.getRoseOrange()*2);
            flower.setOrchidWhite(flower.getOrchidWhite() + flow.getOrchidWhite()*2);
            flower.setOrchidPink(flower.getOrchidPink() + flow.getOrchidPink()*2);
            flower.setOrchidYellow(flower.getOrchidYellow() + flow.getOrchidYellow()*2);
            flower.setChrysanthemums(flower.getChrysanthemums() + flow.getChrysanthemums()*2);
        }
        for (Flower flow : f2020) {
            flower.setRoseRedLong(flower.getRoseRedLong() + flow.getRoseRedLong()*4);
            flower.setRoseRedMini(flower.getRoseRedMini() + flow.getRoseRedMini()*4);
            flower.setRoseYellow(flower.getRoseYellow() + flow.getRoseYellow()*4);
            flower.setRoseWhite(flower.getRoseWhite() + flow.getRoseWhite()*4);
            flower.setRosePink(flower.getRosePink() + flow.getRosePink()*4);
            flower.setRoseOrange(flower.getRoseOrange() + flow.getRoseOrange()*4);
            flower.setOrchidWhite(flower.getOrchidWhite() + flow.getOrchidWhite()*4);
            flower.setOrchidPink(flower.getOrchidPink() + flow.getOrchidPink()*4);
            flower.setOrchidYellow(flower.getOrchidYellow() + flow.getOrchidYellow()*4);
            flower.setChrysanthemums(flower.getChrysanthemums() + flow.getChrysanthemums()*4);
        }
        flower.setRoseRedLong(flower.getRoseRedLong() / 7);
        flower.setRoseRedMini(flower.getRoseRedMini() / 7);
        flower.setRoseYellow(flower.getRoseYellow() / 7);
        flower.setRoseWhite(flower.getRoseWhite() / 7);
        flower.setRosePink(flower.getRosePink() / 7);
        flower.setRoseOrange(flower.getRoseOrange() / 7);
        flower.setOrchidWhite(flower.getOrchidWhite() / 7);
        flower.setOrchidPink(flower.getOrchidPink() / 7);
        flower.setOrchidYellow(flower.getOrchidYellow() / 7);
        flower.setChrysanthemums(flower.getChrysanthemums() / 7);
        return flower;
    }
}
```

**Клас TableModalForNewForecast**

```
package java1.interf;

import java1.db.FlowerDAO;
import javax.swing.table.AbstractTableModel;
import java.util.ArrayList;

public class TableModalForNewForecast extends AbstractTableModel {
    public ArrayList<String[]> tableData;
    public int columnQ1;

    public TableModalForNewForecast() {
        tableData = new ArrayList<String[]>();
        for (int i = 0; i < tableData.size(); i++) {
```

```
            tableData.add(new String[getColumnCount()]);
        }
        this.columnQ1 = 2;
    }

    @Override
    public int getColumnCount() {
        return columnQ1;
    }

    @Override
    public int getRowCount() {
        return tableData.size();
    }

    @Override
    public Object getValueAt(int arg0, int arg1) {
        String[] rows = tableData.get(arg0);
        return rows[arg1];
    }

    @Override
    public String getColumnName(int colomnIndex) {
        switch (colomnIndex) {
            case 0:
                return "Квіти";
            case 1:
                return "Кількість";
        }
        return null;

    }

    public void addData(String[] row) {
        String[] rowTable = new String[getColumnCount()];
        rowTable = row;
        tableData.add(row);
    }

    public void addData(String arg1, String arg2) {
        StringBuilder sbS2018 = new StringBuilder("2018-");
        sbS2018.append(arg1.substring(5));
        StringBuilder sbS2019 = new StringBuilder("2019-");
        sbS2019.append(arg1.substring(5));
        StringBuilder sbS2020 = new StringBuilder("2020-");
        sbS2020.append(arg1.substring(5));

        StringBuilder sbF2018 = new StringBuilder("2018-");
        sbF2018.append(arg2.substring(5));
        StringBuilder sbF2019 = new StringBuilder("2019-");
```

```
            sbF2019.append(arg2.substring(5));
            StringBuilder sbF2020 = new StringBuilder("2020-");
            sbF2020.append(arg2.substring(5));
            ArrayList<db.Flower> arrayList2018 = new FlowerDAO().getFloverList(sbS2018.toString(),
sbF2018.toString());
            ArrayList<db.Flower> arrayList2019 = new FlowerDAO().getFloverList(sbS2019.toString(),
sbF2019.toString());
            ArrayList<db.Flower> arrayList2020 = new FlowerDAO().getFloverList(sbS2020.toString(),
sbF2020.toString());

            db.Flower rez = CountForecast.count(arrayList2018, arrayList2019, arrayList2020);

            String[] row = new String[]{"Троянда червона метрова", Integer.toString(rez.getRoseRedLong())};
            addData(row);

            row = new String[]{"Троянда червона міні", Integer.toString(rez.getRoseRedMini())};
            addData(row);

            row = new String[]{"Троянда жовта", Integer.toString(rez.getRoseYellow())};
            addData(row);

            row = new String[]{"Троянда біла", Integer.toString(rez.getRoseWhite())};
            addData(row);

            row = new String[]{"Троянда рожева", Integer.toString(rez.getRosePink())};
            addData(row);

            row = new String[]{"Троянда оранжева", Integer.toString(rez.getRoseOrange())};
            addData(row);

            row = new String[]{"Орхідея біла", Integer.toString(rez.getOrchidWhite())};
            addData(row);

            row = new String[]{"Орхідея рожева", Integer.toString(rez.getOrchidPink())};
            addData(row);

            row = new String[]{"Орхідея жовта", Integer.toString(rez.getOrchidYellow())};
            addData(row);

            row = new String[]{"Хризантеми асорті", Integer.toString(rez.getChrysanthemums())};
            addData(row);

        }
    }
```

**Клас TableModalForOldFlowers**

```
package java1.interf;

import java1.db.FlowerDAO;

import javax.swing.table.AbstractTableModel;
```

```
import java.util.ArrayList;

public class TableModalForOldFlowers extends AbstractTableModel {
    public ArrayList<String[]> tableData;
    public int columnQ1;

    public TableModalForOldFlowers() {
        tableData = new ArrayList<String[]>();
        for (int i = 0; i < tableData.size(); i++) {
            tableData.add(new String[getColumnCount()]);
        }
        this.columnQ1 = 11;
    }

    @Override
    public int getColumnCount() {
        return columnQ1;
    }

    @Override
    public int getRowCount() {
        return tableData.size();
    }

    @Override
    public Object getValueAt(int arg0, int arg1) {
        String[] rows = tableData.get(arg0);
        return rows[arg1];
    }

    @Override
    public String getColumnName(int colomnIndex) {
        switch (colomnIndex) {
            case 0:
                return "Дата";
            case 1:
                return "Троянда червона метрова";
            case 2:
                return "Троянда червона міні";
            case 3:
                return "Троянда жовта";
            case 4:
                return "Троянда біла";
            case 5:
                return "Троянда рожева";
            case 6:
                return "Троянда оранжева";
            case 7:
                return "Орхідея біла";
            case 8:
```

```
            return "Орхідея рожева";
        case 9:
            return "Орхідея жовта";
        case 10:
            return "Хризантеми асорті";
    }
    return null;

}
public void addData(String[] row) {
    String[] rowTable = new String[getColumnCount()];
    rowTable = row;
    tableData.add(row);
}

public void addData(String arg1, String arg2) {
    ArrayList<db.Flower> flowerList = new FlowerDAO().getFloverList(arg1, arg2);
    for (db.Flower flower : flowerList) {
        String[] row = new String[]{flower.getDate(),
                Integer.toString(flower.getRoseRedLong()),
                Integer.toString(flower.getRoseRedMini()),
                Integer.toString(flower.getRoseYellow()),
                Integer.toString(flower.getRoseWhite()),
                Integer.toString(flower.getRosePink()),
                Integer.toString(flower.getRoseOrange()),
                Integer.toString(flower.getOrchidWhite()),
                Integer.toString(flower.getOrchidPink()),
                Integer.toString(flower.getOrchidYellow()),
                Integer.toString(flower.getChrysanthemums())
        };
        addData(row);
    }
}
}
```

**Клас TableForOldFlower**

```
package java1.interf;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class TableForOldFlower {
    public static ImageIcon image;
    public static JFrame frame;
    private JLabel label;
    private String arg1, arg2;

    public TableForOldFlower(String arg1, String arg2) {
        this.arg1 = arg1;
        this.arg2 = arg2;
```

```
    image = new ImageIcon(getClass().getResource("1.jpg"));
}
public void createWindow() {
    frame = new JFrame();
    frame.setTitle("Архів");
    frame.setSize(1100, 600);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLocationRelativeTo(null);
    Font font = new Font("TimesRoman", Font.BOLD, 14);
    label = new JLabel();
    label.setLayout(new GridBagLayout());
    label.setVisible(true);
    label.setIcon(image);
    frame.add(label);
    frame.setVisible(true);
    TableModalForOldFlowers tmq2 = new TableModalForOldFlowers();
    JTable table = new JTable(tmq2);

    JScrollPane tableScrollPane = new JScrollPane(table);
    tmq2.addData(arg1, arg2);
    int shur = 0;
    if ((19 + tmq2.getRowCount() * 24) > 500) {
        shur = 500;
    } else {
        shur = 19 + tmq2.getRowCount() * 24;
    }
    tableScrollPane.setPreferredSize(new Dimension(1080, shur));
    System.out.println(tmq2.getRowCount());
    tableScrollPane.setBackground(Color.BLUE);
    table.setFont(font);
    table.setRowHeight(24);

    label.add(tableScrollPane, new GridBagConstraints(0, 0, 4, 1, 1, 1, GridBagConstraints.NORTH,
            GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
    JButton backBtn = new JButton();
    backBtn.setFont(font);
    backBtn.setText("Назад");
    label.add(backBtn, new GridBagConstraints(0, 0, 2, 1, 1, 1, GridBagConstraints.SOUTH,
            GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    JButton menuBtn = new JButton();
    menuBtn.setFont(font);
    menuBtn.setText("Головне меню");
    label.add(menuBtn, new GridBagConstraints(2, 0, 2, 1, 1, 1, GridBagConstraints.SOUTH,
            GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    backBtn.addActionListener(new BackListener());
    menuBtn.addActionListener(new MenuListener());
}
public void closeWindow() {
```

```
        frame.setVisible(false);
        frame.remove(label);
    }

    private class BackListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new PageShowOldFlower().createWindow();
        }
    }
    private class MenuListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new MainMenu().createWindow();
        }
    }
}
```

**ДОДАТОК Г (обов'язковий)**

# ГРАФІЧНА ЧАСТИНА

# «ПРОГРАМНИЙ МОДУЛЬ ПРОГНОЗУВАННЯ ОБСЯГУ ЗАКУПІВЛІ КВІТІВ»

Виконала: студентка 4 курсу, групи 1КН-21б
спеціальності 122 – Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)

Моклюк Д.М.
(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри КН
Озеранський В.С.
(прізвище та ініціали)

3 червня 2025р.

Вінниця ВНТУ - 2025 рік

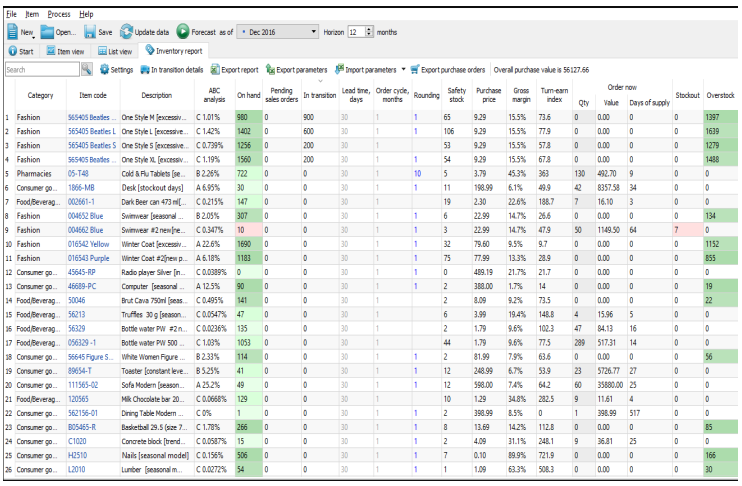


Рисунок Г.1 – Приклад роботи програми Streamline

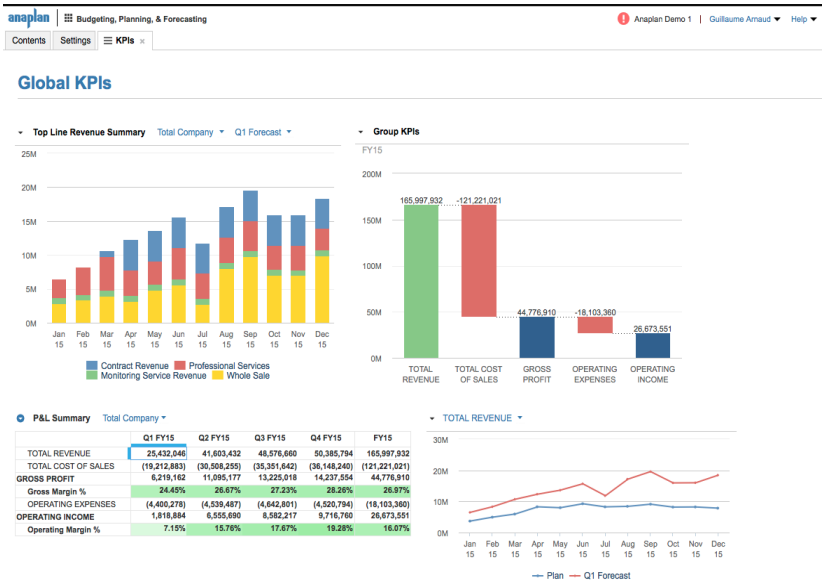



Рисунок Г.2 – Приклад роботи програми AnaPlan

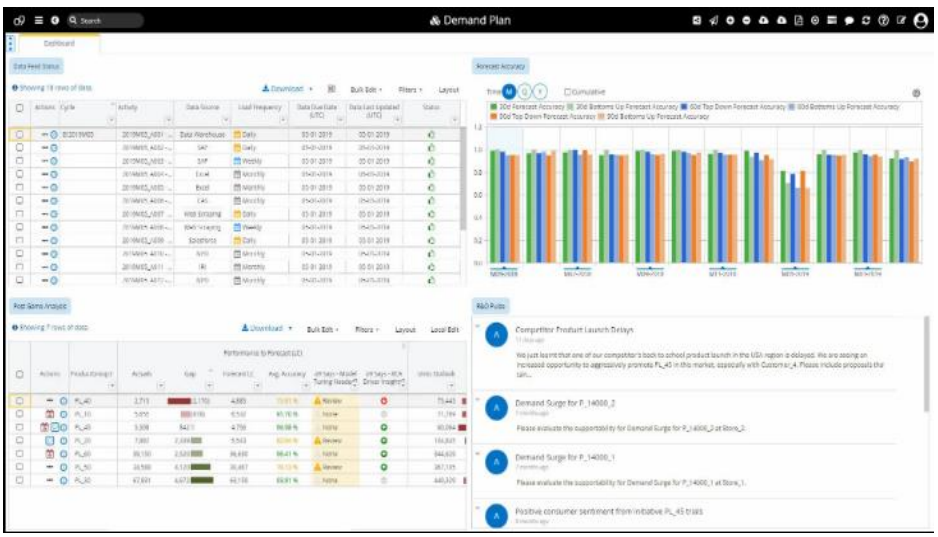



Рисунок Г.3 – Приклад роботи o9 Solutions Digital Brain Platform

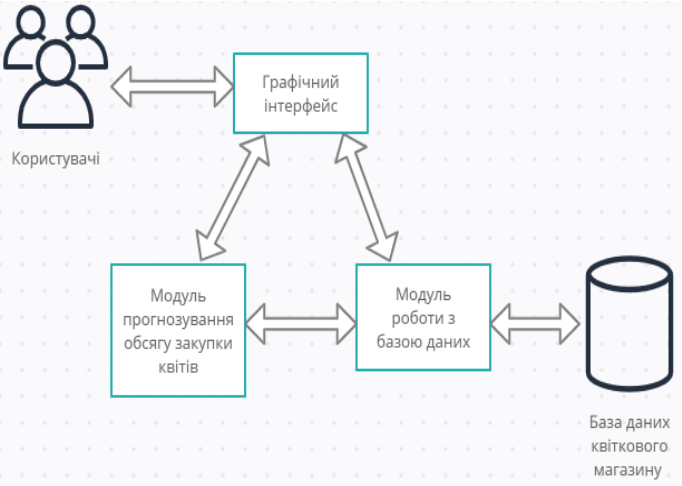



Рисунок Г.4 – Модель функціонування програмного модуля прогнозування обсягу закупівлі квітів

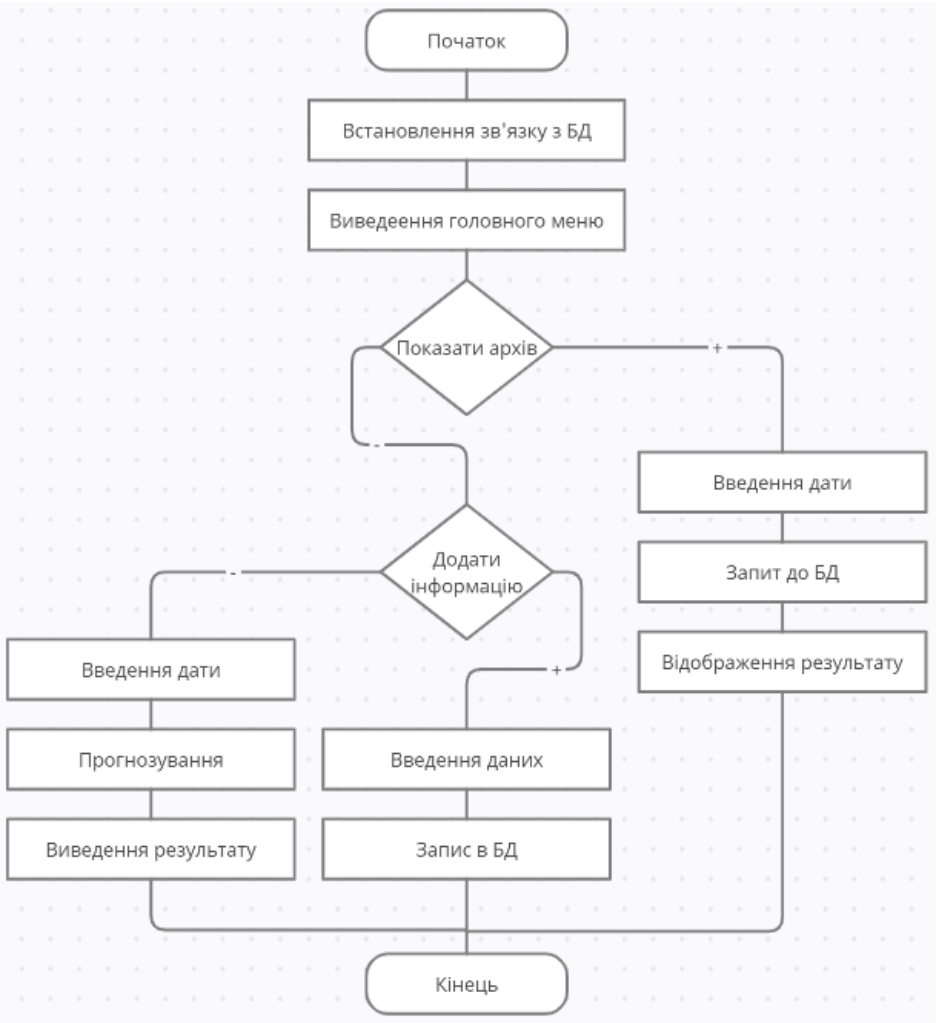



Рисунок Г.5 – Схема загального алгоритму функціонування програмного модуля прогнозування обсягу закупівлі квітів

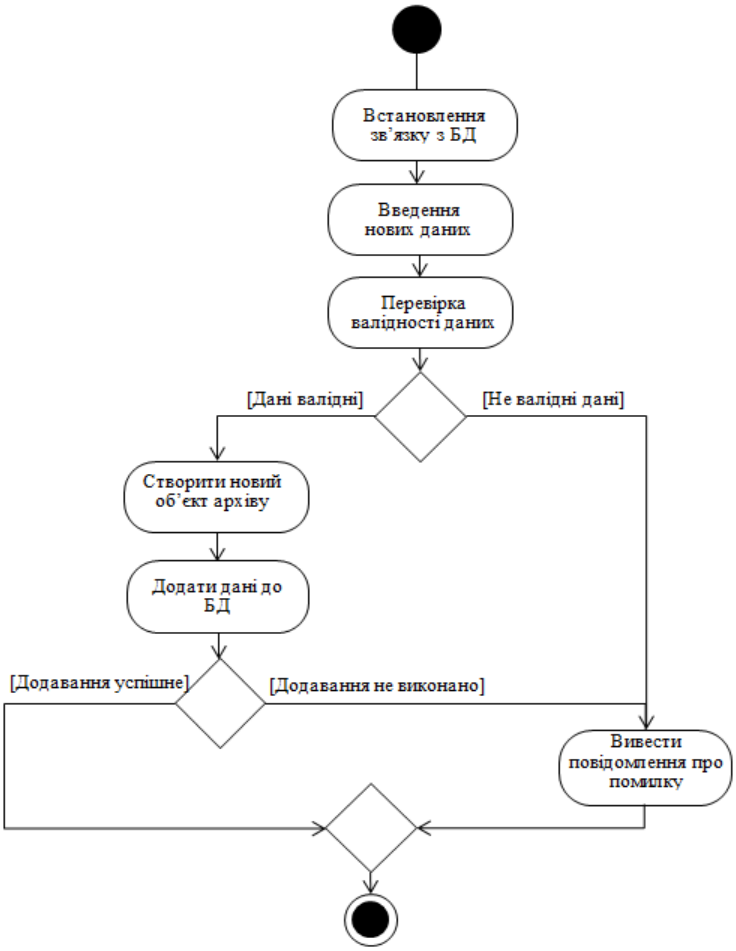



Рисунок Г.6 – Діаграма діяльності системи при введенні нових даних

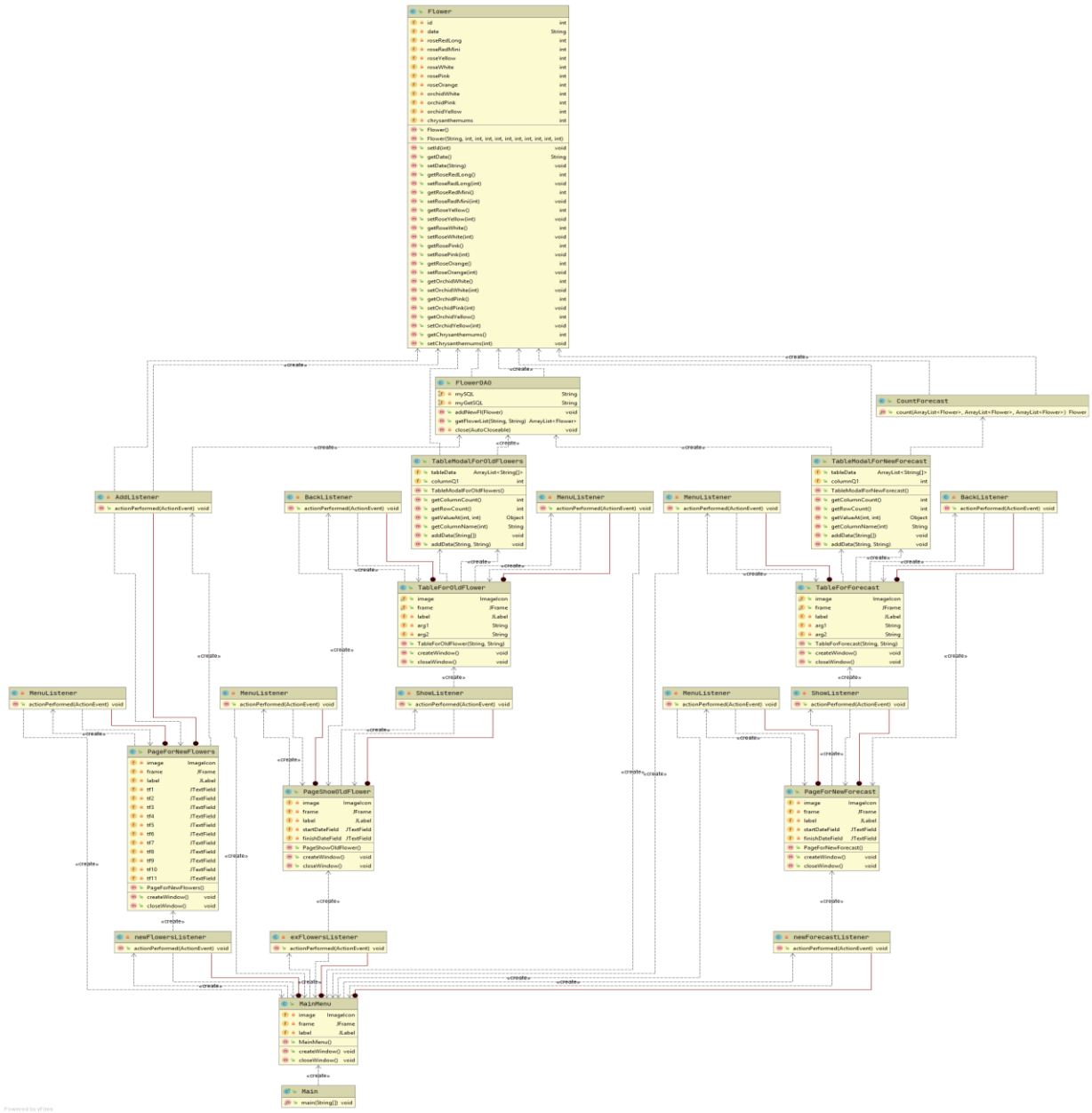
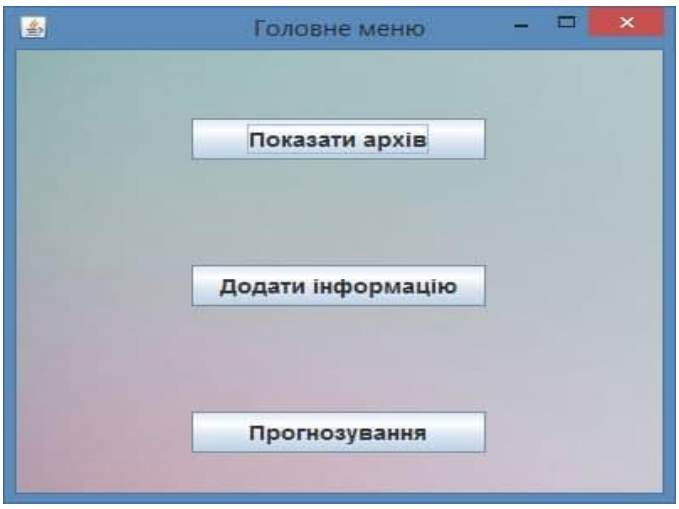
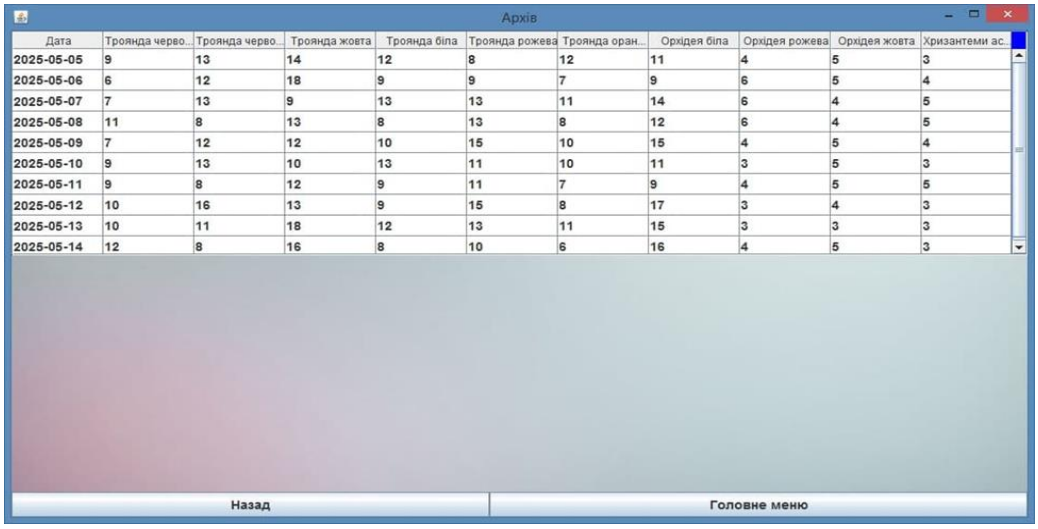



Рисунок Г.7 – Загальна UML-діаграма класів




| Дата | Троянда черво... | Троянда черво... | Троянда жовта | Троянда біла | Троянда рожева | Троянда оран... | Орхідея біла | Орхідея рожева | Орхідея жовта | Хризантеми ас... |
|---|---|---|---|---|---|---|---|---|---|---|
| 2025-05-05 | 9 | 13 | 14 | 12 | 8 | 12 | 11 | 4 | 5 | 3 |
| 2025-05-06 | 6 | 12 | 18 | 9 | 9 | 7 | 9 | 6 | 5 | 4 |
| 2025-05-07 | 7 | 13 | 9 | 13 | 13 | 11 | 14 | 6 | 4 | 5 |
| 2025-05-08 | 11 | 8 | 13 | 8 | 13 | 8 | 12 | 6 | 4 | 5 |
| 2025-05-09 | 7 | 12 | 12 | 10 | 15 | 10 | 15 | 4 | 5 | 4 |
| 2025-05-10 | 9 | 13 | 10 | 13 | 11 | 10 | 11 | 3 | 5 | 3 |
| 2025-05-11 | 9 | 8 | 12 | 9 | 11 | 7 | 9 | 4 | 5 | 5 |
| 2025-05-12 | 10 | 16 | 13 | 9 | 15 | 8 | 17 | 3 | 4 | 3 |
| 2025-05-13 | 10 | 11 | 18 | 12 | 13 | 11 | 15 | 3 | 3 | 3 |
| 2025-05-14 | 12 | 8 | 16 | 8 | 10 | 6 | 16 | 4 | 5 | 3 |

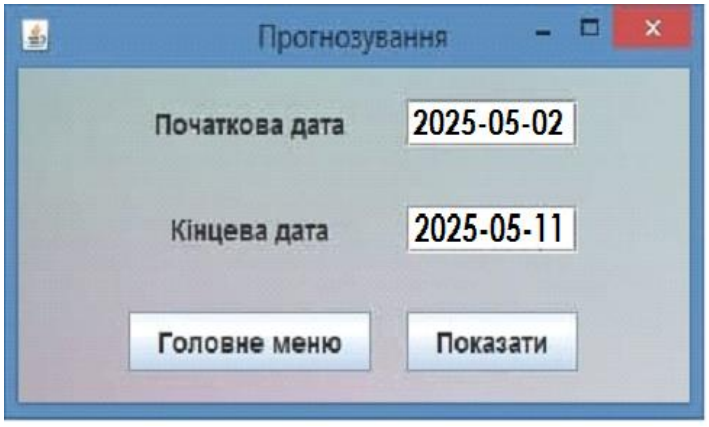



Рисунок Г.8 – Вікна програмного модуля прогнозування обсягу закупівлі квітів

**ДОДАТОК Д (довідниковий)**

**ДОВІДКА ПРО ВПРОВАДЖЕННЯ**

МАЛЕ НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО "ТОВ "ІТІ"
Україна, 21021, м.Вінниця, вул. Келецька, 56

№ 46 від "6" 06 2025р.

# ДОВІДКА

Дана Моклюк Дарії Миколаївні в тому, що результати, одержані нею в процесі виконання бакалаврської кваліфікаційної роботи, а саме алгоритм та програмне забезпечення для прогнозування обсягу закупівлі квітів, планується використати в розробках ТОВ «ІТІ».

Зам. директора ТОВ «ІТІ» Бодяк В.М.