

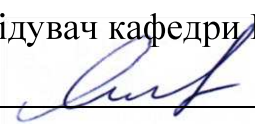
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему: Прогнозування контингенту абітурієнтів для ЗВО

Виконав- студент групи 6145 м

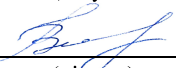

(підпис)

Власенко М.М.

Керівник роботи-

доцент кафедри ІУСТ, к.т.н.,

(посада, науковий ступінь вчене звання)


(підпис)

Ворона М.В.

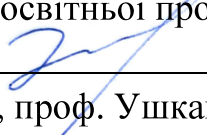
м. Миколаїв – 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 124 "Системний аналіз"
Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми


д.ф-м.н, проф. Ушкац М.В.

" " 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Власенку Максиму Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Прогнозування контингенту абітурієнтів для ЗВО
Керівник роботи Ворона Михайло Владиславович
Затверджені наказом ректора № 1055-уч. від "08" жовтня 2024 року.
2. Термін подання роботи- 15 грудня 2024 р.
3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування
4. Перелік питань, що належать до розробки (найменування розділів)
5. Перелік презентаційних матеріалів,Актуальність роботи,Постановка задачі, Концепція системи, USE CASE,Концептуальна модель,Логічна модель БД, Діаграма послідовності, Математичне забезпечення розробки,Deployment Diagram,Екранні форми розробки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ворона М.В.	05.09.2023	22.12.2023
2	Ворона М.В.	23.12.2023	15.06.2024
3	Ворона М.В.	15.06.2024	25.09.2024
4	Ворона М.В.	25.09.2024	14.11.2024
5	Ворона М.В.	25.09.2024	14.11.2024
6	Ворона М.В.	25.09.2024	14.11.2024

7. Дата видачі завдання 05 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2024	
2	Розробка концепції	15.03.2024	
3	Розділ з розробки проектних рішень по системі (та її частинах)	15.06.2024	
4	Реалізація проекту та розробка робочої документації	25.09.2024	
5	Обчислення економічної ефективності	14.11.2024	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2024	
7	Оформлення пояснювальної записки	14.11.2024	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2024	
9	Подання магістерської роботи рецензенту	20.11.2024	
10	Подання магістерської роботи на державний захист	15.12.2024	

Студент


(підпис)

Власенко М.М.
(ПІБ)

Керівник роботи


(підпис)

Ворона М.В.
(ПІБ)

Анотація

Автором розроблено модель прогнозування контингенту абітурієнтів для закладу вищої освіти (ЗВО)

Використання у закладі вищої освіти методів прогнозування кількості абітурієнтів, вирішує такі важливі завдання, як планування і управління, слабоструктурованими процесами, для підвищення якості прийнятих рішень в складних організаційно-технічних системах.

Об'єкт дослідження: процеси прогнозування кількості абітурієнтів розробки та реалізації

Предмет дослідження: методи прогнозування

Наукова новизна: вибір адекватного методу для закладу вищої освіти

Пояснювальна записка до дипломної роботи складається з 121 сторінок і включає вступ, шість розділів, висновки, список використаних джерел та трьох додатків. Робота містить 27 рисунків і 12 таблиць.

Ключові слова: модель; прогноз; абітурієнти.

Abstract

The author has developed a model for predicting the contingent of applicants for a higher education institution (HEI)

The use of methods for predicting the number of applicants in a higher education institution solves such important tasks as planning and management, weakly structured processes, to improve the quality of decisions made in complex organizational and technical systems.

Object of research: processes of predicting the number of applicants, development and implementation

Subject of research: forecasting methods

Scientific novelty: choosing an adequate method for a higher education institution

The explanatory note to the thesis consists of 121 pages and includes an introduction, five sections, a conclusion, a list of sources and literature used, and three appendices. The work contains 27 figures and 12 tables.

Keywords: model; forecast; applicants.

ЗМІСТ	
ВСТУП	5
РОЗДІЛ 1 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ КІЛЬКОСТІ АБІТУРІЄНТІВ У ВИЩІЙ НАВЧАЛЬНИЙ ЗАКЛАД	7
1.2 Основні поняття і визначення проблеми прогнозування часових рядів	9
1.3 Аналіз методів прогнозування часових рядів	13
1.3.1 Класичні методи аналізу часових рядів	13
1.3.2 Методи згладжування і ковзні середні	19
1.3.3 Методи нечіткого моделювання	22
1.3.4 R/S-аналіз часових рядів	27
1.3.5 Метод клітинних автоматів	33
1.4 Прогнозування часових рядів кількості абітурієнтів	37
1.4.1 Критерії точності прогнозів	37
1.4.2 Модель Хольта та адаптивного згладжування Брауна	38
1.4.3 Модель Бокса-Дженкінса ARIMA	40
1.4.4 Модель клітинних автоматів	45
1.5 Аналіз отриманих результатів та вибір методу прогнозування	46
1.6 Висновки	48
РОЗДІЛ 2 ФОРМУВАННЯ ВИМОГ ДО ПІДСИСТЕМИ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ КІЛЬКОСТІ АБІТУРІЄНТІВ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ	50
2.1 Формування вимог до підсистеми короткострокового прогнозування кількості абітурієнтів	50
2.2.1 Аналіз недоліків існуючих підсистем прогнозування кількості абітурієнтів та обґрунтування необхідності удосконалення	50
2.3 Розробка концепції підсистеми короткострокового прогнозування кількості абітурієнтів	52
2.3.1 Опис результатів вивчення об'єкта автоматизації	52
2.3.2 Аналіз вимог до підсистеми короткострокового прогнозування кількості абітурієнтів	53
2.3.3 Розробка концепції підсистеми	54
2.4 Постановка задачі	61
РОЗДІЛ 3 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ЩОДО ПІДСИСТЕМИ ПРОГНОЗУВАННЯ КІЛЬКОСТІ	64
3.1 Загальносистемні рішення	64

3.1.1 Вибір засобів моделювання.....	64
3.1.2 Місце підсистеми прогнозування в інформаційній системі вищого навчального закладу	65
3.2 Рішення з організаційного забезпечення.....	66
3.3 Рішення з інформаційного забезпечення	67
3.3.1 Розробка концептуальної моделі БД підсистеми	67
3.3.2 Розробка логічної моделі БД підсистеми	69
3.3.3 Розробка фізичної моделі БД підсистеми	71
3.4 Рішення з технічного забезпечення	73
3.5 Рішення з математичного забезпечення.....	74
3.6 Рішення з програмного забезпечення.....	78
3.6.1. Побудова діаграми варіантів використання.....	78
3.6.2 Опис функцій, що автоматизуються.....	83
3.6.3 Побудова діаграми класів	84
3.6.4 Побудова діаграми послідовності.....	88
3.6.5 Методи і засоби розробки ПЗ	91
3.4.6 Побудова діаграми компонентів	93
РОЗДІЛ 4 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПІДСИСТЕМИ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ КІЬКОСТІ СТУДЕНТІВ	96
4.1 Загальні положення	96
4.2. Розрахунок інвестиційних вкладень на впровадження підсистеми короткострокового прогнозування кількості студентів.....	97
4.2.1 Кошторис витрат на розробку підсистеми короткострокового прогнозування кількості студентів	97
4.2.2 Кошторис на технічні засоби, програмне забезпечення, які необхідно придбати для впровадження проекту	99
4.2.3 Розрахунок витрат на монтаж і настройку системи, підготовку кадрів та ін.	100
4.3 Розрахунок витрат на супроводження (експлуатацію) підсистеми.....	100
4.4 Розрахунок показників економічної ефективності від впровадження результатів проектування	100
РОЗДІЛ 5. ОХОРОНА ПРАЦІ ПРИ РОБОТІ З ПК.....	102
5.1 Загальні визначення та актуальність питання охорони праці при роботі з ПК.....	102
5.2 Аналіз небезпечних і шкідливих факторів при роботі з ПК	103

5.3 Розрахунок системи загального освітлення приміщення з ПК	106
5.4 Техніка безпеки при роботі з ПК	107
РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	112
6.1 Вступна частина.....	112
6.2 Забруднення навколишнього середовища біля малого підприємства, розташованого у центрі міста Миколаєва.	113
6.3 Заходи щодо зменшення забруднення.....	116
ВИСНОВКИ	118
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	120
ДОДАТОК А	121
ДОДАТОК Б	122
ДОДАТОК В.....	127
ДОДАТОК Г	130

ВСТУП

Прогнозування контингенту абітурієнтів для ЗВО

Прогнозування контингенту абітурієнтів – це стратегічно важливий процес для будь-якого закладу вищої освіти (ЗВО). Точне передбачення кількості та якості вступників дозволяє університетам ефективно планувати навчальний процес, розподіляти ресурси, розробляти маркетингові стратегії та адаптуватися до змін на ринку освітніх послуг.

Актуальність цього питання зумовлена кількома факторами:

Конкуренція між ЗВО. За останні роки спостерігається зростання конкуренції серед закладів вищої освіти, що вимагає від університетів постійно шукати нові способи залучення абітурієнтів.

Зміни демографічної ситуації, демографічні процеси, такі як зниження народжуваності, міграція населення, впливають на кількість потенційних абітурієнтів.

Зміни в освітній політиці. Реформи в освітній сфері, нові стандарти та вимоги до вищої освіти також впливають на вибір абітурієнтів.

Розвиток технологій, нові технології, такі як онлайн-навчання, відкриті освітні ресурси, змінюють спосіб отримання освіти і впливають на вибір молоді.

Мета цього дослідження полягає в аналізі існуючих методів прогнозування контингенту абітурієнтів, визначенні їх сильних і слабких сторін, а також розробці рекомендацій для підвищення точності прогнозів.

Завдання дослідження-проаналізувати літературу з питання прогнозування контингенту абітурієнтів.

Визначити основні фактори, що впливають на вибір абітурієнтів.

Оглянути існуючі методи прогнозування.

Розробити модель прогнозування, яка враховує специфіку конкретного ЗВО. Оцінити точність розробленої моделі.

Практична значимість дослідження полягає в тому, що отримані результати можуть бути використані університетами для більш ефективного планування своєї діяльності та підвищення конкурентоспроможності.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ КІЛЬКОСТІ АБІТУРІЄНТІВ У ВИЩІЙ НАВЧАЛЬНИЙ ЗАКЛАД

1.1 Прогноз і цілі його використання

У зв'язку з розвитком ІКТ практично у всіх організаціях і підприємствах активно використовують зберігання інформації в базах даних (БД), в тому числі і у вигляді часових рядів. При цьому значно підвищується роль людини, як постачальника і як користувача інформації в БД, який вносить свою частку невизначеності в дані при їх сприйнятті і використанні, при вирішенні професійних, зазвичай нечітко сформульованих завдань. Моделлю сучасних підприємств можуть служити складні організаційно-технічні системи, в яких виділені елементи трьох класів (технічні, інформаційні та гуманістичні) і є безліч зв'язків між ними. Проблема невизначеності, обумовлена внутрішньої і зовнішньої нестационарністю, в тому числі, значною роллю людини (гуманістичного елемента), є невід'ємною приналежністю складних організаційно-технічних систем і на практиці характеризується:

- неможливістю або обмеженою можливістю експериментального дослідження процесів, що не дозволяє отримати достатню і достовірну статистичну інформацію;
- суб'єктивної невизначеністю через багатоцільового, многокритеріального, експертного оцінювання інформації;
- неоднорідністю, неточністю інформації.

У моделюванні, проектуванні та аналізі складних організаційно-технічних систем широко використовуються експертні системи, які оперують експертними оцінками, заснованими на досвіді і знаннях експерта (або групи експертів). Фундаментальною проблемою є проблема прогнозування поведінки слабоструктурованих процесів, що протікають в складних організаційних системах, оскільки результати прогнозування є основою для прийняття управлінських рішень. Традиційно прогнозування розвитку процесів в умовах

невизначеності засноване на аналізі часових рядів. Однак невизначеності слабоструктурованих процесів і відповідних їм ЧР породжені як об'єктивної, так і суб'єктивної невизначеністю, властивою складним організаційно-технічним системам. Стосовно до ЧР ці невизначеності виражені в:

1. Невизначеності значень ЧР - гетерогенном характері значень (кількісному і якісному), неточності значень ЧР;
2. Невизначеності довжини ЧР - недостатності довжини ЧР (7 – 60 значень). Згідно з концепцією інформаційного підходу до вимірювання невизначеності при розгляді ЧР, як повідомлення, невизначеність для ЧР убуває зі збільшенням його довжини;
3. Невизначеності поведінки ЧР - нестационарном поведінці ЧР, нечіткості і нестабільності тенденцій.

Тому важливою науково-технічною проблемою є аналіз, моделювання і прогнозування ЧР, що володіють високим ступенем невизначеності, що виражається в поєднанні властивостей: невеликій довжині (7 - 60 значень), нестационарності поведінки, неточності значень і нечіткості тенденцій. Рішення означеної проблеми дозволить моделювати поведінку слабоструктурованих процесів, що створить можливість для підвищення якості прийнятих рішень в складних організаційно-технічних системах.

Кінцева ефективність будь-якого рішення залежить від послідовності подій, що виникають вже після прийняття рішення. Можливість передбачити некеровані аспекти цих подій перед прийняттям рішення дозволяє зробити найкращий вибір, який, в іншому випадку, міг би бути не таким вдалим. Тому системи планування і управління, зазвичай, реалізують функцію прогнозу.

Прогнозування – це передбачення майбутніх подій. Метою прогнозування є зменшення ризику при прийнятті рішень. Прогноз зазвичай виходить помилковим, але помилка залежить від використовуваної прогнозуючої системи. Надаючи прогнозу більше ресурсів, можна збільшити точність прогнозу і зменшити збитки, пов'язані з невизначеністю при прийнятті рішень.

Оскільки прогнозування ніколи не зможе повністю знищити ризик при прийнятті рішень, необхідно явно визначати неточність прогнозу. Зазвичай, прийняте рішення визначається результатами прогнозу (при цьому передбачається, що прогноз правильний) з урахуванням можливої помилки прогнозування.

Сказане вище припускає, що система, що прогнозує повинна забезпечувати визначення помилки прогнозування, також як і саме прогнозування. Такий підхід значно знижує ризик об'єктивно пов'язаний з процесом прийняття рішень.

Необхідно зазначити, що прогнозування – це не кінцева мета. Система, що прогнозує – це частина великої системи управління і як підсистема, вона взаємодіє з іншими компонентами системи, граючи чималу роль в одержуваному результаті.

Передбачення кількості абітурієнтів представлених часовими рядами – необхідний елемент будь-якого вищого навчального закладу. Відповідно, передбачення часових рядів лежить в основі діяльності всієї індустрії освіти.

1.2 Основні поняття і визначення проблеми прогнозування часових рядів

В сучасній статистичній теорії існує багато різноманітних методів прогнозування економічної, соціальної інформації. Значна їх частина відноситься до прогнозування часових рядів. Особливістю прогнозування часових рядів є те, що аналізуються лише дані спостережень без додаткової інформації, без аналізу впливу зовнішніх сил. Звичайно, такий аналіз виглядає досить неповним, але доволі часто прогнози часових рядів є більш точними [1].

Часовий ряд (або ряд динаміки) – ряд послідовних значень (рівнів, спостережень), які характеризують зміну показника в часі.

Розрізняють неперервні та дискретні часові ряди. Неперервні часові ряди частіше всього зустрічаються в інженерній справі. Формально такий ряд представляється числовою функцією часу. Якщо спостереження проводяться в визначені дискретні моменти часу, то говорять про дискретний ряд.

Економічні або соціальні часові ряди, як правило, представлені або як ряди миттєвих вимірів, або як агреговані ряди. Перший тип спостережень можна розглядати як значення неперервного часового ряду в визначені дискретні моменти часу. Так, наприклад, ціна на певний товар існує в будь-який момент часу. Але на практиці ряди цін звичайно будуються по дням, неділям і т. д., тобто ціни відмічаються в визначені фіксовані моменти часу. Прикладами рядів такого типу є котирування валют, акцій, облігацій, процентних ставок і т. д. Інший тип – це ряди, які представляють агреговані або зведені значення однієї або декілька інших змінних (показників). Типовим прикладом є щоденні обсяги операцій з цінними паперами за певний період. До такого ж типу відносяться щоденна кількість опадів, що випало, щорічний об'єм промислової продукції, відстань, яка пройдена авіалайнерами даної компанії за певний період.

Для того щоб визначити проблему прогнозування, розглянемо її детальніше. Результати прогнозування використовуються для підтримки прийняття рішень. Отже, природа прийнятих рішень визначає більшість бажаних характеристик прогнозуючої системи. Вивчення проблеми, яка вирішується, повинне допомогти відповісти на питання про те, що потрібно прогнозувати, яку форму має прийняти прогноз, які тимчасові елементи включаються і яка бажана точність прогнозу.

При визначенні того, що потрібно прогнозувати, необхідно вказувати змінні, які аналізуються і передвіщаються. Тут дуже важливий необхідний рівень деталізації. На використовуваний рівень деталізації впливає безліч факторів: доступність і точність даних, вартість аналізу і переваги користувачів результатів прогнозування. У ситуаціях, коли найкращий набір змінних неясний, можна спробувати різні альтернативи і вибрати один з варіантів, що дає найкращі результати. Зазвичай так здійснюється вибір при розробці прогнозуючих систем, заснованих на аналізі історичних даних.

Другий важливий етап при побудові прогнозуючої системи – це визначення наступних трьох параметрів: періоду прогнозування, горизонту

прогнозування та інтервалу прогнозування. *Період прогнозування* – це основна одиниця часу, на яку робиться прогноз. Може знадобитися прогноз кількості абітурієнтів на наступний навчальний рік. У цьому випадку період – рік. *Горизонт прогнозування* – це число періодів у майбутньому, які покриває прогноз. Тобто, може знадобитися прогноз на 1 рік вперед. У цьому випадку період – 12 місяців, а горизонт – 1 рік. Нарешті, *інтервал прогнозування* – частота, з якою робиться новий прогноз. Часто інтервал прогнозування збігається з періодом прогнозування. У цьому випадку прогноз переглядається кожний період, використовуючи вимоги за останній період та іншу поточну інформацію в якості базису для прогнозу, що переглядається.

Вибір періоду і горизонту прогнозування зазвичай диктується умовами прийняття рішень в області, для якої проводиться прогноз. Для того щоб прогнозування мало сенс, горизонт прогнозування повинен бути не менше, ніж час, необхідний для реалізації рішення, прийнятого на основі прогнозу. Таким чином, прогнозування дуже сильно залежить від природи прийнятого рішення. У деяких випадках, час, необхідний на реалізацію рішення, не визначено, наприклад, як у разі поставки запасних частин для поповнення запасів ремонтних підприємств. Існують методи роботи в умовах такої невизначеності, але вони підвищують варіацію помилки прогнозування. Оскільки із збільшенням горизонту прогнозування точність прогнозу, зазвичай, знижується, часто можна поліпшити процес ухвалення рішення, зменшивши час, необхідний на реалізацію рішення і, отже, зменшивши горизонт і помилку прогнозування.

Інтервал прогнозування часто визначається операційним режимом системи обробки даних, яка забезпечує інформацію про прогнозовану змінну. У тому випадку, якщо котирування повідомляються щодня, то, можливо, для внутрішньоденного прогнозу цих даних недостатньо і інтервал прогнозування – доба є більш обґрунтованим.

Хоча різниця не велика, особливо хочеться звернути увагу на відмінність між даними за період і точковими даними. Дані за період характеризують певний період часу. Точкові дані представляють значення змінної в конкретний

момент часу, наприклад, останній рік набору. Різниця між цими двома типами даних важливо в основному для вибору використовуваної системи збору даних, процесу вимірювань і визначення помилки прогнозування.

Третім аспектом прогнозування є необхідна форма прогнозу. Зазвичай при прогнозуванні проводиться оцінка очікуваного значення змінної, плюс оцінка варіації помилки прогнозування чи проміжку, на якому зберігається ймовірність вмісту реальних майбутніх значень змінної. Цей проміжок називається передбачуваним інтервалом.

У деяких випадках не так важливо пророкування конкретних значень прогнозованої змінної, як прогноз значних змін у її поведінці. Така задача виникає, наприклад, при прогнозі моменту, коли поточний напрямок руху ринку (тренд) змінить свій напрямок на протилежне.

Точність прогнозу, необхідна для конкретної проблеми, має великий вплив на прогножуючу систему. Найважливішою характеристикою системи управління є її здатність домагатися оптимальності при роботі з невизначеністю.

До цих пір, обговорювалося набір проблем, пов'язаних з процесом прийняття рішення. Існує ряд інших факторів, які також необхідно брати до уваги при розгляді проблеми прогнозування. Один з них пов'язаний з процесом, що генерує змінну. Якщо відомо, що процес стабільний, або існують постійні умови, або зміни в часі відбуваються повільно – система, що прогнозує для такого процесу може досить сильно відрізнитися від системи, яка повинна проводити прогнозування нестійкого процесу з частими фундаментальними змінами.

У першому випадку, необхідно активне використання історичних даних для передбачення майбутнього, в той час як у другому – краще зосередитися на суб'єктивній оцінці і прогнозуванні для визначення змін у процесі. Інший фактор – це доступність даних. Історичні дані необхідні для побудови прогножуючих процедур; майбутні спостереження служать для перевірки

прогнозу. Кількість, точність і достовірність цієї інформації важливі при прогнозуванні.

Необхідно також відзначити обчислювальні обмеження прогнозуючих систем. Якщо для вирішення завдання досить прогнозування невеликої кількості змінних і ці прогнози виконуються рідко, то в системі можливе застосування більш глибоких процедур аналізу. У разі ж необхідності частого прогнозування великої кількості змінних, можна велику увагу приділити розробці ефективного управління даними.

І, нарешті, два важливі чинники проблеми прогнозування – можливості і інтерес людей, які роблять і використовують прогноз. В ідеалі, історична інформація аналізується автоматично, і прогноз надається аналітику для можливої модифікації. Введення експерта в процес прогнозування є бажаним і дуже важливим. Далі прогноз передається аналітикам, які використовують його при прийнятті рішень.

1.3 Аналіз методів прогнозування часових рядів

1.3.1 Класичні методи аналізу часових рядів

Підхід до аналізу часових рядів визначається обчислювальними структурами і припущеннями, що лежать в основі моделювання (Побудова моделі) динамічного процесу, що породжує тимчасовий ряд, і на цій основі об'єднує моделі об'єктів і залежностей між ними. Системний принцип дозволяє розглядати динамічний процес у вигляді динамічної системи. За оцінками зарубіжних і вітчизняних систематиків прогностики налічується понад ста методів прогнозування.

Аналіз часових рядів – сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозу. Метою аналізу часових рядів є побудова математичної моделі ряду, за допомогою якої можна пояснити поведінку ряду і здійснити прогноз на майбутні періоди.

Інструментарій класичної економетрики розроблявся на припущенні, що рівні часового ряду породженні випадковим або стохастичним процесом і для них виконується умова стаціонарності [1]. Тому широкий спектр методів аналізу та прогнозування розроблено для стаціонарних часових рядів.

Часовий ряд є *стаціонарним*, якщо він має постійні та скінченні математичні сподівання та дисперсію для кожного періоду часу, і його автокореляційна функція залежить тільки від різниці моментів часу.

Класичні методи аналізу базуються на *декомпозиції* часового ряду, який досліджується, на компоненти тренду, сезонності, циклічності і залишкової компоненти.

Будь-який часовий ряд можна представити як суму детермінованого та випадкового компонентів:

$$X_t = D_t + R_t, \quad t = \overline{1, n}. \quad (1.1)$$

В свою чергу детермінований компонент складається з трьох частин: трендового, сезонного, циклічного компонентів.

$$D_t = Tr_t + S_t + C_t, \quad t = \overline{1, n}. \quad (1.2)$$

Таким чином, будь-який часовий ряд можна розглядати як суму:

$$X_t = Tr_t + S_t + C_t + R_t, \quad t = \overline{1, n}. \quad (1.3)$$

Такий вигляд часового ряду отримав назву *адитивної моделі*. Якщо ж замість реальних значень компонентів використовувати їх логарифми, то отримаємо *мультиплікативну модель*:

$$\begin{aligned} \log X_t &= \log Tr_t + \log S_t + \log C_t + \log R_t, \quad t = \overline{1, n}, \quad \text{або} \\ X_t &= Tr_t \cdot S_t \cdot C_t \cdot R_t, \quad t = \overline{1, n}. \end{aligned} \quad (1.4)$$

Детермінований компонент змінюється за певними правилами, які можуть бути визначені за допомогою досліджень і відповідного аналізу часового ряду. Як правило, одним з основних параметрів, від яких залежить детермінований компонент, є час.

Аналіз часового ряду починається з виділення *трендового компонента*. Його присутність неважко помітити, проаналізувавши графік часового ряду. Як правило, для економічних даних дуже типовим є повільне зростання чи падіння

на протязі тривалого періоду часу. Звичайно, це не означає, що спостерігається постійне зростання кожного року, але на кожному доволі великому періоді воно є помітним. Наявність *тренду* в економічних часових рядах можна пояснити демографічними змінами, технологічними змінами, змінами в структурі виробництва, попиту, тощо. Дія таких факторів є постійною, тому дослідники мають змогу описувати такі зміни за допомогою кривих, які можна задати в аналітичному вигляді.

Сезонний компонент показує коливання навколо трендового компонента. Він визначає вплив пори року. Кожен період часу протягом року характеризується своїм сезонним індексом, який показує наскільки вище або нижче відповідний показник в даний період часу у порівнянні з іншими періодами. Найбільш характерна властивість цієї компоненти – періодичність.

Циклічний компонент складається з послідовних зростань і спадань, які не повторюються регулярно, наприклад, кожен рік, і тому виключаються із сезонної компоненти. Так як ці зростання і спадання чергуються, то їх не можна вважати достатньо випадковими і розглядати як частину незалежної випадкової помилки (випадкової компоненти).

Випадковий компонент є те, що залишилось від часового ряду після виключення тренду, циклічного та сезонного компонентів. Частина таких ефектів може бути віднесена до непередбачених природних катаклізмів (землетруси, пожежі, тощо), частина – до випадкових дій людей. За наявності випадкової компоненти неможливо прогнозувати значення часового ряду без помилки. Але будь-який реальний економічний процес включає випадковий компонент.

Від значення кожного компонента залежить значення часового ряду у кожний період спостережень. Але не завжди представляється можливим характеризувати кожний компонент окремо. Іноді краще робити прогноз відносно всієї моделі, ніж намагатися виділити кожний компонент окремо [2].

При декомпозиції часового ряду на компоненти принципово важливим залишається той факт, що вибір виду тренда і визначення характеристик

сезонної компоненти в великій мірі залежить від суб'єктивної оцінки дослідника. Останнє означає неминучу втрату інформації про динаміку часового ряду, який досліджується, після його декомпозиції [3]. Одним із підходів до побудови прогнозних моделей, які не використовують декомпозицію, є використання клітинних автоматів (див. розділ 1.4.5).

В роботі [4] зазначено, що в результаті проведення вказаної «хірургічної» операції декомпозиції втрачається або спотворюється в окремих випадках істотна «інтегральна» інформація про динаміку поведінки часового ряду, що негативним чином позначається на точності прогнозу, який одержується в результаті.

Виявлення структури часового ряду часто ґрунтується на аналізі автокореляційної функції (корелограми). Корелограма представляє собою графік коефіцієнтів автокореляції r_k між значеннями часового ряду, які розділені k періодами часу (лагами, запізненнями). Вказані коефіцієнти обчислюються наступним чином:

$$r_k = \frac{\sum_{t=1}^{N-k} (X_t - \bar{X}) \cdot (X_{t+k} - \bar{X})}{\sum_{t=1}^N (X_t - \bar{X})^2}, \quad (1.5)$$

де N – довжина часового ряду, X_t – значення змінної X в момент часу t , $t = \overline{1, N}$, $\bar{X}$ – середнє значення змінної X_t .

Стандартна помилка коефіцієнта автокореляції з запізненням k обчислюється за допомогою формули:

$$SE(r_k) = \sqrt{\frac{1 + 2 \sum_{i=1}^{k-1} r_i^2}{N}}. \quad (1.6)$$

Довірчий інтервал для коефіцієнта автокореляції обчислюється за допомогою -розподілу Стюдента: $[0 \mp t \cdot SE]$.

Аналіз автокореляційної функції дозволяє визначити лаг, при якому автокореляція найбільш висока, а отже, і лаг, при якому зв'язок між попереднім і поточним рівнем ряду найбільш тісний. Коефіцієнт автокореляції характеризує тільки лінійний зв'язок поточного і попереднього рівнів ряду. Тому по коефіцієнту автокореляції можна судити про наявність лінійної (або близької

до лінійної) тенденції. Для деяких часових рядів, які мають сильну нелінійну тенденцію коефіцієнт автокореляції рівнів ряду може наближатися до нуля.

Якщо найбільш високим виявився коефіцієнт автокореляції першого порядку ($k=1$), то ряд має лише тенденцію. Якщо найбільш високим виявився коефіцієнт автокореляції порядку i ($k=i$), то ряд має циклічні коливання з періодом k моментів часу.

За знаком коефіцієнта автокореляції не можна робити висновок про зростаючу або спадну тенденцію в рівнях ряду. Більшість фінансових часових рядів містить позитивну автокореляцію рівнів, але при цьому вони можуть мати спадну тенденцію.

Якщо жоден з коефіцієнтів автокореляції не виявився значимим, то можна зробити наступні припущення:

- ряд не містить тенденції і циклічних коливань і має структуру випадкової компоненти;
- ряд містить сильну нелінійну тенденцію, для виявлення якої необхідно провести додатковий аналіз.

Корисний метод дослідження періодичності полягає в дослідженні *приватної автокореляційної функції* (ПАКФ). У ПАКФ усувається залежність між проміжними спостереженнями. Іншими словами, приватна автокореляція на даному лагу аналогічна звичайній автокореляції, за винятком того, що при обчисленні з неї видаляється вплив автокореляцій з меншими лагами. На лагу 1 (коли немає проміжних елементів усередині лага), приватна автокореляція дорівнює, очевидно, звичайній автокореляції. Насправді, приватна автокореляція дає більш чисту картину періодичних залежностей.

Порядок аналізу часового ряду [2,5]:

1) *Побудова і вивчення графіка.* Аналіз часового ряду починається з побудови і вивчення його графіка. Якщо нестационарність часового ряду очевидна, то першою справою треба виділити нестационарну складову ряду. Процес виділення тренду та інших компонент ряду, що призводять до порушення стаціонарності, може проходити в декілька етапів. На кожному з них

розглядається ряд залишків, отриманий у результаті вирахування з вихідного ряду підбраної моделі тренду, або результат різницевих і інших перетворень ряду. Крім графіків, ознаками нестаціонарності часового ряду можуть служити автокореляційна функція, що прямує не до нуля (за винятком дуже великих значень лагів) і наявність яскраво виражених піків на низьких частотах у періодограмі. За допомогою автокореляційної функції досліджують також внутрішні зв'язки між елементами часових рядів.

У вибіркових дослідженнях найпростіші числові характеристики описової статистики (середнє, медіана, дисперсія, стандартне відхилення, коефіцієнти асиметрії й ексцесу) звичайно дають достатньо інформативне уявлення про вибірку. Графічні методи уявлення й аналізу вибірок при цьому грають лише допоміжну роль, дозволяючи краще зрозуміти локалізацію і концентрацію даних, їхній закон розподілу.

Роль графічних методів при аналізі часових рядів цілком інша. Табличне уявлення тимчасового ряду й описових статистиків часті усього не дозволяє зрозуміти характер процесу, у той час як за графіком тимчасового ряду можна зробити досить багато висновків. Надалі вони можуть бути перевірені й уточнені за допомогою розрахунків.

Людське око досить упевнено визначає за графіком тимчасового ряду:

- 1) наявність тренду і його характер;
- 2) наявність сезонних і циклічних компонент;
- 3) ступінь повільності або переривчастості змін послідовних значень ряду після усунення тренду (по цьому показнику можна судити про характер і розмір кореляції між сусідніми елементами ряду).

Так графічний аналіз ряду звичайно задає напрямок його подальшого аналізу.

2) *Вибір моделі для часового ряду.* Після того, як вихідний процес максимально наближений до стаціонарного, можна приступити до вибору різноманітних моделей отриманого процесу. Мета цього етапу – опис і урахування надалі аналізу кореляційної структури аналізованого процесу.

Модель може вважатися підбраною, якщо залишкова компонента ряду є процесом типу, як правило, "білого шуму". Після підбору залишки аналізуються для перевірки адекватності моделі та побудови надійних інтервалів.

3) *Прогнозування або інтерполяція*. Останнім етапом аналізу часового ряду може бути прогнозування його майбутніх (екстраполяція) або відновлення пропущених (інтерполяція) значень і визначення точності цього прогнозу на базі підбраної моделі. Добре підібрати математичну модель вдається не для всякого часового ряду. Нерідко буває і так, що для опису підходять відразу декілька моделей. Неоднозначність вибору моделі може спостерігатися як на етапі виділення детермінованого компонента ряду, так і при виборі структури ряду залишків. Тому досить часто розробляють декілька прогнозів, зроблених за допомогою різних моделей.

Ідея нечіткого моделювання поведінки складних систем по ЧР заснована на базовій моделі нечіткого динамічного процесу (Song, Chissom, 1993), що отримала назву «нечіткого часового ряду». У нечіткому тимчасовому ряді (НЧР) станів динамічного процесу зіставлені нечіткі значення, що моделюються параметрическими функціями приналежності, залежності між ними будуються за спостереженнями і представляються базою нечітких правил «Якщо-То», чисельно виражаються матрицею нечіткого відносини. Для прогнозування НЧР застосовується алгоритм нечіткого логічного виведення.

1.3.2 Методи згладжування і ковзні середні

Серед статистичних методів короткострокового прогнозування в економіці та фінансах широке розповсюдження отримали адаптивні методи прогнозування [6,7], перевагою яких є побудова таких моделей, які здатні враховувати різну інформаційну цінність рівнів ряду та реагувати на ступінь розбіжності прогнозних оцінок з фактичними значеннями.

Адаптивні моделі можуть будуватися за схемою ковзної середньої. За схемою *ковзної середньої* оцінкою поточного рівня є зважене середнє всіх попередніх рівнів, причому ваги при спостереженнях убують по мірі

віддалення від останнього (поточного) рівня, тобто інформаційна цінність спостережень тим більше, чим ближче вони до кінця періоду спостережень.

Розглянемо методи, основані на схемі ковзної середньої.

1) Метод ковзного середнього

Традиційним методом прогнозування майбутнього значення показника є усереднення k його минулих значень:

$$\hat{X}_{t+1} = \frac{(X_t + X_{t-1} + X_{t-2} + \dots + X_{t-k+1})}{k}, \quad (1.7)$$

Такий метод називається методом ковзного середнього. Всі спостереження мають однакові вагові коефіцієнти.

2) Метод експоненціального згладжування

У порівнянні з методом ковзного середнього, де задіяні тільки найбільш свіжі дані, в методі експоненціального згладжування використовується зважене (експоненціальне) ковзне середнє всіх попередніх спостережень [9]:

$$\hat{X}_{t+1} = \alpha \cdot X_t + (1 - \alpha) \cdot \hat{X}_t, \quad (1.8)$$

де α - постійна згладжування.

Більш пізні спостереження мають більшу вагу. Вага присвоюється наступним чином: для останнього спостереження вагою буде величина α , для передостаннього $(1 - \alpha)$, для того, яке було перед ним – $(1 - \alpha)^2$ і т. д.

Експоненціальне згладжування – це просто старий прогноз $\hat{X}_t$ з уточненням у вигляді добутку α і помилки останнього прогнозу $(X_t - \hat{X}_t)$:

$$\hat{X}_{t+1} = \hat{X}_t + \alpha \cdot (X_t - \hat{X}_t). \quad (1.9)$$

Якщо α близьке до 1, то в прогнозі значно враховується величина помилки останнього прогнозування. І навпаки, при малих значеннях α величина, яка прогнозується, найбільш близька до попереднього прогнозу. Оцінку оптимального значення α можна отримати мінімізацією середньоквадратичної помилки (MSE). Послідовно обчислюються прогнози при α , рівному 0.1, 0.2, ..., 0.9, і розраховується величина середньоквадратичної помилки прогнозування для кожного з них. Те значення α , для якого величина

помилки буде найменшою, і обирається для подальшого використання в прогнозах.

Початкове $\hat{X}_0$ беруть рівним першому спостереженню або середньому значенню перших спостережень.

3) Модель Хольта

В моделі Хольта враховується лінійний тренд, який присутній в часових рядах. Якщо в часових рядах є тенденція до росту, то разом з оцінкою поточного рівня S_t необхідна оцінка і нахилу T_t . В моделі Хольта значення рівня і нахилу згладжуються, при цьому використовуються різні постійні згладжування для кожного з них (α, β) . Ці постійні згладжування дозволяються оцінити поточний рівень і нахил, уточнюючи їх кожен раз, коли з'являються нові спостереження. Нижче приведені три рівняння, які складають модель Хольта:

- 1) Оцінка поточного рівня:

$$S_t = \alpha \cdot X_t + (1 - \alpha) \cdot (S_{t-1} + T_{t-1}). \quad (1.10)$$

- 2) Оцінка тренду:

$$T_t = \beta \cdot (S_t - S_{t-1}) + (1 - \beta) \cdot T_{t-1}. \quad (1.11)$$

- 3) Прогноз на p періодів вперед:

$$\hat{X}_{t+p} = S_t + pT_t. \quad (1.12)$$

4) Модель адаптивного згладжування Брауна

В основу моделі адаптивного згладжування Брауна покладена ідея, що можна задати деякий параметр γ такий, що зважена сума відхилень між значеннями, що спостерігаються, і прогнозованими значеннями стає мінімальною:

$$\sum_{i=1}^{\infty} \gamma^i \cdot (X_{t-i} - \hat{X}_{t-i})^2 = \min. \quad (1.13)$$

Нижче приведені три рівняння, які складають модель адаптивного згладжування Брауна [9]:

- 1) Експоненціально згладжений ряд або оцінка поточного рівня:

$$S_t = S_{t-1} + T_{t-1} + (1 - \gamma) \cdot (X_t - \hat{X}_t). \quad (1.14)$$

- 2) Оцінка тренду:

$$T_t = T_{t-1} + (1 - \gamma)^2 \cdot (X_t - \hat{X}_t). \quad (1.15)$$

- 3) Прогноз на p періодів вперед:

$$\hat{X}_{t+p} = S_t + pT_t \quad (1.16)$$

На відміну від моделі Хольта модель адаптивного згладжування Брауна має одну постійну згладжування γ .

1.3.3 Методи нечіткого моделювання

У середині 90-х років минулого століття був розроблений принципово новий і досить потужний клас алгоритмів для прогнозування часових рядів. Більшу частину роботи з дослідження методології та перевірки моделей була проведена двома статистиками, Г.Е.П.Боксом і Г.М. Дженкінсом. З тих пір побудова подібних моделей і одержання на їх основі прогнозів іноді називається методами Бокса-Дженкінса.

Для прогнозування динаміки нестационарних часових рядів, для яких існує такий порядок d , що послідовні різниці цього порядку утворюють стаціонарний часовий ряд, призначена модель Бокса-Дженкінса ARIMA (*модель авторегресії - проінтегрованого ковзного середнього*) [10].

Модель ARIMA є лише одним із представників цілого класу моделей, який також включає модель авторегресії (AR), модель ковзного середнього (MA) і модель авторегресії-ковзного середнього (ARMA).

1) Модель авторегресії AR (p)

Авторегресійний процес – це процес, в якому значення рівнів знаходяться в лінійній залежності від своїх попередніх значень. Широко розповсюдженими прикладами авто регресійних процесів є процеси першого (марковські процеси)

і другого (процеси Юла) порядків. Порядок AR позначається через p . Модель AR (p) має вигляд:

$$X_t = a_0 + a_1 \cdot X_{t-1} + a_2 \cdot X_{t-2} + \dots + a_p \cdot X_{t-p} + \varepsilon_t, \quad (1.17)$$

де X_t – залежна змінна в момент часу t , $a_0, a_1, a_2, \dots, a_p$ – параметри, які оцінюються, ε_t – помилка, яка описує вплив змінних що не враховуються в моделі.

Параметр a_0 залежить від постійного рівня ряду. Якщо дані змінюються в околиці нуля або виражаються як відхилення від середнього, то параметр a_0 не потрібен. Параметри $a_1, a_2, \dots, a_p$ частіше за все шукають через систему рівнянь Юла-Уолкера:

[illegible]

де r_k – коефіцієнт автокореляції між значеннями часового ряду, які розділені k періодами часу (лагами, запізненнями). Він підраховується за наступною формулою (1.5).

2) Модель ковзного середнього $MA(q)$

Модель ковзного середнього (МА) припускає, що в помилках моделі в попередні періоди зосереджена інформація про всю передісторію ряду. В моделях ковзного середнього (МА) прийнято виділяти період запізнення q . Модель $MA(q)$ задається наступним рівнянням:

$$X_t = \mu - b_1 \cdot \varepsilon_{t-1} - b_2 \cdot \varepsilon_{t-2} - \dots - b_q \cdot \varepsilon_{t-q} + \varepsilon_t, \quad (1.19)$$

де X_t – залежна змінна в момент часу t , $b_1, b_2, \dots, b_q$ – параметри, які оцінюються, μ – постійне середнє процесу, ε_t – помилки в попередні періоди часу, які в момент t включені в X_t .

Рівняння (1.19) схоже на рівняння (1.17) за винятком того, що залежна змінна X_t залежить від попередніх значень помилок, замість самої змінної. Моделі з ковзним середнім (МА) дають проноз значення функції X_t , основуючись на лінійній комбінації обмеженого числа попередніх помилок, тоді

як авторегресійні моделі (AR) дають прогноз X_t на основі лінійної функції апроксимації обмеженого числа попередніх значень X_t .

Використання терміну ковзне середнє для моделі в рівнянні (1.19) є історичним, і його не можна путати з процедурою ковзного середнього, яка описувалася в попередньому підрозділі. Тут вираз ковзне середнє відноситься до того факту, що відхилення $X_t - \mu$ є лінійною комбінацією поточних і минулих помилок, а так як час рухається вперед, то помилки, які включені в цю комбінацію, також будуть зміщуватися вперед:

$$\begin{aligned} X_t - \mu &= \varepsilon_t - b_1 \cdot \varepsilon_{t-1} - b_2 \cdot \varepsilon_{t-2} - \dots - b_q \cdot \varepsilon_{t-q}, \\ X_{t+1} - \mu &= \varepsilon_{t+1} - b_1 \cdot \varepsilon_t - b_2 \cdot \varepsilon_{t-1} - \dots - b_q \cdot \varepsilon_{t-q+1}. \end{aligned}$$

Оцінку параметрів b часто здійснюють на основі апроксимації процесу ковзного середнього моделлю авторегресії високого порядку [11]. Процес ковзного середнього допускає еквівалентне представлення у вигляді авторегресії нескінченного порядку:

$$X_t = \varepsilon_t + a_1 \cdot X_{t-1} + a_2 \cdot X_{t-2} + a_3 \cdot X_{t-3} + \dots,$$

в якій коефіцієнти a_k зв'язані з параметрами b відповідно формулі:

$$a_k + \sum_{i=1}^q a_{k-i} \cdot b_i = 0 \quad a_0 = 1, k = 1, 2, \dots \quad (1.20)$$

Обмежуючись кінцевим числом членів авторегресії, отримаємо апроксимацію процесу ковзного середнього у вигляді:

$$X_t = a_1 \cdot X_{t-1} + a_2 \cdot X_{t-2} + \dots + a_m \cdot X_{t-m} + \varepsilon_t,$$

в якій $m \gg q$ повинне бути достатньо великим (не менше ніж $5q$). Спочатку для обраного порядку m оцінюють коефіцієнти авторегресії ($a_1 \dots a_m$) шляхом рішення системи рівняння (1.18). Далі, для визначення параметрів ($b_1 \dots b_q$) використовується відношення (1.20) для $k = 1, 2, \dots, m$. Для оцінки використовується метод найменших квадратів, тобто мінімізується квадратична функція:

$$f(\mathbf{b}) = \sum_{k=1}^m (a_k + \sum_{i=1}^q b_i a_{k-i})^2 \rightarrow \min \quad (1.21)$$

за вектором $\mathbf{b} = \{b_1, b_2, \dots, b_q\}$.

3) Модель авторегресії - ковзного середнього ARMA(p, q)

ARMA-модель включає в себе як члени авторегресії так і члени ковзного середнього:

$$X_t = a_0 + a_1 \cdot X_{t-1} + a_2 \cdot X_{t-2} + \dots + a_p \cdot X_{t-p} + \varepsilon_t + b_1 \cdot \varepsilon_{t-1} + b_2 \cdot \varepsilon_{t-2} + \dots + b_q \cdot \varepsilon_{t-q}, \quad (1.22)$$

де p – порядок авторегресії, q – порядок ковзного середнього. На практиці порядки авторегресії та ковзного середнього рідко перевищують 2 [12].

4) Модель авторегресії проінтегрованого ковзного середнього ARIMA(p, d, q).

В моделях AR, MA, ARMA моделюють стаціонарні процеси. Для прогнозування динаміки нестаціонарних часових рядів, для яких існує такий порядок d , що послідовні різниці цього порядку утворюють стаціонарний часовий ряд, призначена модель авторегресії проінтегрованого ковзного середнього ARIMA(p, d, q). В цій моделі використовується той факт, що перехід від початкового ряду до його різниць відповідного порядку анулює тренд і в результаті можна отримати стаціонарний ряд, який представляється ARMA-моделлю. При цьому використовується поняття різницевого оператора ∇^d порядку d . Різницевий оператор першого порядку має вигляд: $\nabla X_t = X_t - X_{t-1}$. Послідовне d - кратне застосування оператора записується у вигляді оператора d -го порядку: $\nabla^d = \nabla[\nabla^{d-1} X_t]$. Після застосування до початого ряду різницевого оператора певного порядку d маємо стаціонарний часовий ряд, який в подальшому описується моделлю ARMA(p, q).

ARIMA-модель включає в себе всі розглянуті вище моделі:

- при $d = 1, p = 0, q = 0$ маємо випадкове блукання;
- при $d = 0$ – ARMA-процес;
- при $d = 0, p > 0, q = 0$ – стаціонарний процес авторегресії AR-процес;
- при $d = 0, p = 0, q > 0$ – стаціонарний процес ковзного середнього MA-процес;

З практичної точки зору важливо зрозуміти характерні особливості часових рядів, які можна представити ARIMA-моделями. ARIMA-моделі представляють певну форму нестационарності, яка характеризується властивістю однорідності. Нестационарний часовий ряд називається *однорідним*, якщо його випадкова складова, отримана після вилучення детермінованого тренду, представляє собою стаціонарний (в широкому розумінні) часовий ряд. Однорідні часові ряди характеризуються постійною волатильністю, відсутністю стрибків, імпульсів (короткочасних підвищень і понижень) і розривів, і це один їх недоліків цього класу моделей, так як в рядах цінкових індексів, всі ці явища присутні доволі часто [11].

Ідентифікація моделей ARIMA :

Ціль етапу ідентифікації – виявлення конкретного типу загального сімейства лінійних моделей ARIMA, тобто попередня оцінка початкових значень параметрів p, d, q . Процедура ідентифікації виконується в наступному порядку:

1) обчислюються різниці рівнів ряду до тих пір, поки вони не виявляться стаціонарними відносно математичного сподівання і дисперсії. Ця операція дозволяє отримати оцінку параметра d ;

2) виконується оцінювання параметрів в моделі $ARMA(p, q)$ для стаціонарного ряду $Y_t = \nabla^d X_t$.

Основний інструмент, з допомогою якого вирішується поставлена задача, – автокореляційна і приватна автокореляційна функції (АКФ і ПАКФ). При ідентифікації моделей ARMA, використовуються наступні закономірності, які зв'язують параметри моделі і поведінку АКФ і ПАКФ та полегшують процес вибору моделі [13]:

1) модель $ARMA(1, 0)$: АКФ експоненціально загасає. Оцінка автокореляції на лагу 1 є оцінка параметра авторегресії, тому АКФ експоненціально загасає, при від'ємному значенні параметра змінюючи знак, при додатному – не змінюючи; ПАКФ обривається на лагу 1 (для інших лагів кореляцій немає).

2) модель ARMA(2,0): АКФ має форму загасаючої синусоїдної хвилі або експоненціально загасає (в залежності від знаків параметрів); ПАКФ обривається на лагу 2.

3) модель ARMA(0,1): АКФ обривається на лагу 1 (для інших лагів кореляцій немає); ПАКФ експоненціально загасає або монотонно, або змінюючи знак.

4) модель ARMA(0,2): АКФ обривається на лагу 2 (для інших лагів кореляцій немає); ПАКФ експоненціально загасає або має форму загасаючої синусоїдної хвилі.

5) модель ARMA(1,1): АКФ експоненціально загасає, починаючи з першого лагу, загасання може бути монотонне і коливальне; ПАКФ також загасає.

1.3.4 R/S-аналіз часових рядів

1) Показник Херста та його оцінка для часових рядів кількості абітурієнтів

Засновником R/S-аналізу є британський статистик Х.Е. Херст. Він був гідрологом, який почав працювати над проектом нільської греблі близько 1907 р., і прожив у регіоні Нілу майже сорок років. Весь цей час він займався проблемою контролю за рівнем води в резервуарі. Проблема полягала в тому, який щорічний скид води вибрати, щоб резервуар ніколи не переповнювався і не залишався порожнім. При створенні моделі було висунуто загальне припущення про те, що некерована частина системи – в даному випадку приплив води від дощів – слідує випадковим блуканням. Херст вимірював коливання рівня води в резервуарі щодо середнього з плином часу. Тобто він розраховував розмах R , який є різницею між максимальним і мінімальним рівнями накопиченого відхилення $X_{N,t}$:

$$X_{N,t} = \sum_{i=1}^t (e_i - M_N), \quad (1.23)$$

де $X_{N,t}$ – накопичене відхилення за N періодів, e_i – приріст у році i , M_N – середнє e_i за N періодів; тоді розмах обчислюється:

$$R_N = \max_{1 \leq t \leq N} (X_{N,t}) - \min_{1 \leq t \leq N} (X_{N,t}). \quad (1.24)$$

Можна було очікувати, що діапазон (розмах) коливань рівнів води в резервуарі щодо середнього з плином часу буде змінюватися в залежності від величини часового проміжку вимірювань. Якщо ряд випадковий, розмах буде збільшуватися пропорційно кореню квадратному з часу, так як в процесі броунівського руху випадкова частка проходить відстань R , яка збільшується пропорційно квадратному кореню від часу T спостереження за цією часткою, тобто $R \approx \sqrt{T}$.

Для порівняння різних типів часових рядів Херст поділив розмах R на стандартне відхилення спостережень S і ввів наступне відношення:

$$(R/S)_N = a \cdot N^H, \quad (1.23)$$

де a – константа, N – число спостережень. Значення $(R/S)_N$ називають нормованим розмахом, а показник ступеня H називається показником Херста.

Якщо ряд представляє собою випадкове блукання, то показник H повинен дорівнювати 0,50. Тобто розмах накопичених відхилень буде збільшуватися пропорційно квадратному кореню із часу N . Але Херст виявив, що $H = 0,91$. Останнє значить, що нормований розмах збільшується швидше, ніж квадратний корінь із часу. Значення $H = 0,91$ означало, що зміни в щорічних нільських розливах впливають один на одного, або іншими словами, що ряд, який розглядався, має довготривалу пам'ять. Кожне спостереження несе пам'ять про попередні спостереження. Недавні події мають вплив більший, ніж події віддалені, але залишковий вплив цих останніх завжди відчутний. У довготривалому масштабі система, яка дає статистику Херста, є результат довгого потоку взаємопов'язаних подій. Те, що трапляється сьогодні, впливає на майбутнє. Те, де знаходиться система тепер, визначається тим, де вона була в минулому.

Подальше дослідження Херста та інших вчених привели до відкриття існування пам'яті практично в усіх часових рядах, які відображають еволюцію

явищ природи – випадання опадів, плям на сонці, річним кільцям і т. д. Усвідомлення універсальності цього факту з'явилося через третину століття, коли численними дослідженнями було встановлено, що довготривала пам'ять притаманна багатьом часовим рядам, які відображають динаміку еволюційних процесів в соціально-економічній та інших сферах людської діяльності [14].

Є три різні класифікації для показника Херста:

1) $H = 0,50$. Вказує на випадковий ряд. Функція щільності ймовірності може бути нормальною кривою, але це не обов'язкова умова. R/S-аналіз може класифікувати довільний ряд безвідносно до того, який тип розподілу йому відповідає.

2) $0 \leq H < 0,50$. Даний діапазон відповідає антиперсистентним рядам. Такий тип системи часто називають – «повернення до середнього». Якщо система демонструє ріст в попередній період, то скоріше за все, в наступному періоді почнеться спад. І навпаки, якщо йшло зниження, то вірогідний близький підйом. Стійкість такого антиперсистентної поведінки залежить від того, наскільки H близьке до нуля. Такий ряд є більше мінливим або волатильним, ніж ряд випадковий, тому що складається з частих реверсів спад-підйом. Незважаючи на широке поширення концепції повернення до середнього в економіці і фінансовій літературі, до цих пір було знайдено мало антиперсистентних рядів.

3) $0,50 < H < 1,00$. Маємо персистентні, або трендостійкі ряди. Якщо ряд зростає(спадає) в попередній період, то вірогідно, що він буде берегти цю тенденцію якийсь час в майбутньому. Трендостійкість поведінки, або сила персистентності, збільшується при наближенні H до 1. Чим ближче ряд до 0,50, тим більш зачумлений ряд і тим менш виражений його тренд. Персистентні часові ряди є найбільш цікавим класом, так як виявилось, що вони не лише у достатку виявляються в природі, але й властиві ринкам капіталу.

Херст запропонував формулу для оцінки величини H по значенню R/S :

$$H = \frac{\log(R/S)}{\log(a \cdot N)}, \quad (1.24)$$

де R – розмах накопиченого відхилення Z_u ; S – середньоквадратичне відхилення ряду спостережень X ; N – число спостережень; a – задана константа, додатне число. Автор показника Херст емпірично розрахував константу a для часових рядів природних явищ як 0.5, проте в роботі [15] пропонується брати значення $\pi/2$.

Середньоквадратичне відхилення ряду спостережень S обчислюється:

$$S = \sqrt{\frac{1}{N} \sum_{t=1}^N (X_t - \bar{X})^2}, \quad (1.25)$$

де $\bar{X}$ – середнє арифметичне ряду спостережень X за N періодів:

$$\bar{X} = \frac{1}{N} \sum_{t=1}^N X_t. \quad (1.26)$$

Розмах накопиченого відхилення R є найбільш важливим елементом формули розрахунку показника Херста. В загальному вигляді його обчислюють наступним чином:

$$R = \max_{1 \leq u \leq N} (Z_u) - \min_{1 \leq u \leq N} (Z_u), \quad (1.27)$$

де Z_u – накопичене відхилення ряду X від середнього $\bar{X}$:

$$Z_u = \sum_{t=1}^u (X_t - \bar{X}). \quad (1.28)$$

В роботі [15] був проведений розрахунок показника Херста для часових рядів курсів валют, результати якого наведені в таблиці 1.1.

В наведеній таблиці видно, що поведінка всіх представлених валютних пар на всіх розглянутих періодах (день, година, хвилина) персистентна або трендостійка. Показник Херста вищий за 0.5 підтверджує наявність довготривалої пам'яті ринку – теперішнє залежить від минулого, а майбутнє – від теперішнього.

Оцінити глибину пам'яті можна за допомогою алгоритму послідовного R/S -аналізу, який описано в наступному підпункті.

Таблиця 1.1 Показники Херста рядів кількості студентів

Валютна пара	2018 рік (день)	2018 рік (години)	20 квітня 2018 рік (хвилини)
	N = 260	N = 6180	N = 1440
EUR/USD	0,781	0,855	0,832
GBP/USD	0,788	0,860	0,847
USD/CHF	0,783	0,857	0,772
USD/JPY	0,756	0,838	0,836
EUR/GBP	0,726	0,816	0,848
EUR/CHF	0,766	0,844	0,840
EUR/JPY	0,782	0,855	0,837
GBP/CHF	0,756	0,838	0,848
GBP/JPY	0,780	0,854	0,844
CHF/JPY	0,779	0,853	0,835
USD/CAD	0,781	0,855	0,848
EUR/CAD	0,708	0,808	0,849
AUD/USD	0,790	0,861	0,844
AUD/JPY	0,787	0,859	0,842
NZD/USD	0,792	0,863	0,837
XAU/USD	0,773	0,850	0,784
XAG/USD	0,798	0,866	0,808
Середнє значення	0,772	0,849	0,832

2) Послідовний *R/S*-аналіз для оцінки глибини пам'яті часового ряду

Алгоритм послідовного *R/S*-аналізу часового ряду [16]:

1) У вихідному часовому ряду $Z = \{z_1, z_2, \dots, z_n\}$ послідовно виділяємо його початкові часові відрізки $Z_\tau = \{z_1, z_2, \dots, z_\tau\}$, $\tau = 3, 4, \dots, n$, і обчислюємо поточні середні:

$$\bar{z}_\tau = \frac{1}{\tau} \sum_{i=1}^{\tau} z_i, \quad \tau = \overline{3, n}. \quad (1.29)$$

2) Для кожного Z_τ , $\tau = 3, 4, \dots, n$, визначаємо накопичене відхилення для його відрізків довжини t :

$$X_{\tau, t} = \sum_{i=1}^t (z_i - \bar{z}_\tau), \quad t = \overline{1, \tau}. \quad (1.30)$$

3) Обчислюємо різницю між максимальним і мінімальним накопиченими відхиленнями – розмах:

$$R = R(\tau) = \max_{1 \leq t \leq \tau} (X_{\tau,t}) - \min_{1 \leq t \leq \tau} (X_{\tau,t}). \quad (1.31)$$

4) Нормуємо розмах на стандартне відхилення $S = S(\tau)$ для відрізка ЧР Z_τ , $\tau = 3, 4, \dots, n$, отримуємо величину R/S .

5) Обчислюємо показник Херста:

$$H = H(\tau) = \frac{\log(R(\tau)/S(\tau))}{\log(\tau/2)}. \quad (1.32)$$

На основі формули (1.42) будуємо в декартових логарифмічних координатах дві траєкторії. Одна з них, яка називається H -траєкторією, складається із точок з координатами (x_τ, y_τ) , де $x_\tau = \log(\tau)$, $\tau = 3, 4, \dots, n$, $y_\tau = H(\tau)$. Друга називається R/S -траєкторією і складається із точок з координатами (x_τ, y_τ) , де $x_\tau = \log(\tau)$, $\tau = 3, 4, \dots, n$, $y_\tau = \log(R(\tau)/S(\tau))$.

Основою для твердження, що деякий часовий ряд має пам'ять, є виконання наступних умов [17]:

- H -траєкторія ЧР виявляється в області чорного шуму ($H > 0,5$) через декілька своїх початкових точок;
- для R/S -траєкторії ЧР вказані точки входження до чорного шуму демонструють наявність тренду.

Глибину пам'яті визначає той за порядком (в області чорного шуму) номер $q = \tau$, для якого виконуються наступні умови:

- H -траєкторія в q -й точці отримує від'ємний приріст;
- R/S -траєкторія в цій точці демонструє так називає мий «зрив з тренду», тобто різку зміну тренда попередніх точок траєкторії.

Оцінку глибини пам'яті можна представити у вигляді нечіткої множини (НМ) [14]. Вона складається з наступних етапів:

1) Формування на базі вихідного ЧР $Z = \{z_1, z_2, \dots, z_n\}$ сімейства $S(Z) = \{Z^r\}$, $Z^r = \{z_i^r\}$, $i = 1, 2, \dots, n_r$, $r = 1, 2, \dots, m$, яке складається із m часових рядів, де індексом i занумеровані елементи r -го ряду, який отримується із $(r-1)$ -го ЧР Z^{r-1} шляхом видалення його першого елемента z_1^{r-1} . Тут m визначається як найбільше значення індексу r , таке що ряд $Z^m = \{z_i^m\}$, $i = 1, 2, \dots, n_m$ ще має

точку зміни тренду в його R/S -траєкторії; вихідний ЧР Z також входить до сімейства $S(Z)$, в якому йому присвоєна значення індексу $r = 1$.

2) Реалізація R/S - аналізу для всіх ЧР сімейства $S(Z)$. Для кожного ЧР Z' , $r = 1, 2, \dots, m$, необхідно побудувати R/S -траєкторію і H -траєкторію і визначити номер l_r точки, що знаходиться в області чорного шуму, в якій відбулася зміна тренду, тобто l_r – це номер першої за порядком точки, яка знаходиться в області чорного шуму ($H > 0,5$), в якій H -траєкторія отримала від'ємний приріст, а R/S -траєкторія змінила тренд. Таким чином формуємо множину $L(Z)$ значень номерів точок зміни тренду в рядах із сімейства $S(Z)$.

3) Формування нечіткої множини глибини пам'яті для ЧР. Необхідно обчислити долю таких рядів сімействі $S(Z)$, у кожного з яких номер точки l_r дорівнює l : $d(l) = N(l)/m$, де $N(l)$ – кількість всіх рядів в сімействі $S(Z)$, у кожного з яких $l_r = l$, m – загальна кількість рядів в сімействі $S(Z)$. Далі треба визначити максимальну долю d^* і відповідну їй глибину l^* . Для цієї глибини експертним шляхом встановлюється значення функції належності $\mu^* = \mu(l^*)$. Після чого для інших елементів l , які належать множині $L(Z)$, відповідні їм значення функції належності $\mu(l)$ обчислюються по формулі: $\mu(l) = \mu^* d(l) / d^*$. Нечітка множина глибини пам'яті для ЧР записується у вигляді: $M(Z) = \{(l; \mu(l))\}$.

Числове значення оцінки глибини пам'яті можна отримати, обчисливши центр тяжіння:

$$L_{\text{цт}} = \frac{\sum_{l=l_0}^{L_0} l \cdot \mu_z(l)}{\sum_{l=l_0}^{L_0} \mu_z(l)}, \quad (1.33)$$

де l_0 і L_0 – відповідно мінімальна і максимальна глибина пам'яті.

1.3.5 Метод клітинних автоматів

Областю застосування алгоритму прогнозування на базі клітинного автомату є еволюційні процеси, часові ряди показників яких мають довготривалу пам'ять [18].

Клітинний автомат [19] представляється звичайно у вигляді регулярної системи клітин, в кожній із яких зберігається інформація, яка визначає її стан, а

також кінцевого набору правил, які програмують характер зміни цієї інформації (тобто переходу із одного стану в інший) в кожний заданий момент часу. Правила визначають «взаємодію» клітин зі своїми ближніми сусідами. Клітинні автомати можуть бути одно-, двох- або багатомірними. Робота клітинного автомату складається із тактів, тобто представляє собою багаторазове застосування фіксованих правил. Результатом одного такту визначена конфігурація станів клітин автомату.

Якщо математична модель процесу будується на базі клітинного автомату, то в процесі аналізу динаміки його поведінки можна використовувати два підходи [20]. Перший – статистичний, основною ціллю якого є визначення того, з якою ймовірністю і які конфігурації зустрічались раніше і будуть зустрічатися далі. Другий підхід називають конструктивним. Він пов'язаний з конструюванням і аналізом різних типів структур (конфігурацій) і вивченням їх взаємодії. Клітинно-автоматна прогнозна модель об'єднує ці два підходи. Спочатку складається перелік всіх можливих конфігурацій, які зустрічаються в часовому ряді, що розглядається. Далі здійснюється статистичний аналіз їх переходу в певні стани прогнозованої системи. Завершальний етап реалізує перехід конфігурацій в певний стан – прогноз, причому даний перехід представляється в поняттях і термінах теорії нечітких множин [21].

Алгоритм прогнозування на базі клітинного автомату складається із п'яти етапів [14,18,22]:

Етап 1. *Передпрогнозний аналіз даного часового ряду (ЧР)*. На даному етапі вихідний ЧР досліджується на наявність пам'яті та здійснюється оцінка її глибини L . Інструментарій передпрогнозного аналізу – метод послідовного R/S -аналізу, який описаний вище.

Перетворення числового ЧР в лінгвістичний ЧР (ЛЧР). Перетворення вихідного ЧР $Z = \{z_1, z_2, \dots, z_n\}$ в ЛЧР $U = \{u_1, u_2, \dots, u_n\}$ означає заміну числових елементів лінгвістичними, які називаються термами; сукупність термів прийнято називати терм-множиною, яку позначають $T = \{u^1, u^2, \dots, u^k\}$. Множина T складається із k елементів. Для здійснення перетворення вихідний

ряд ділиться на декілька частин або вікон. Для кожного вікна визначаються максимальне і мінімальне значення елементів часового ряду, які входять до нього. Різниця між ними визначає ширину вікна. Знайдена ширина вікна ділиться на визначену кількість інтервалів, яка дорівнює кількості термів в терм-множині. Кожному інтервалу ставиться у відповідність певний терм. Далі кожному елементу вихідного часового ряду $Z = \{z_1, z_2, \dots, z_n\}$ ставиться у відповідність певне значення терму із множини $T = \{u^1, u^2, \dots, u^k\}$, в залежності від того, в який із інтервалів він потрапив, в результаті чого отримується ЛЧР $U = \{u_1, u_2, \dots, u_n\}$. Мінімальна довжина вікна m зв'язана з кількістю термів k наступною залежністю:

$$m \approx 3 \cdot k. \quad (1.34)$$

Далі на основі ЛЧР будується пам'ять клітинного автомату.

Етап 2. Побудова клітинного автомату для отриманого ЛЧР. Наявність в ЛЧР пам'яті пропонується представляти в термінах і поняттях клітинного автомата, зокрема лінійного клітинного автомата. Терміном « l -конфігурація» називають всяку послідовність $g = u_{i+1}u_{i+2}\dots u_{i+l}$, $i = 0, 1, 2, \dots, n-l$, $l = 1, 2, \dots, L$, де L — глибина пам'яті ряду який розглядається. Розглянемо яку-небудь фіксовану l -конфігурацію, яку позначимо $g^0 = u_1^0 u_2^0 \dots u_j^0 \dots u_l^0$. Якщо в ЛЧР виділена послідовність $u_{i+1}u_{i+2}\dots u_{i+j}\dots u_{i+l}$, $i = 0, 1, 2, \dots, n-l$, яка співпадає з g^0 , тобто $u_{i+j} = u_j^0$, $j = 1, 2, \dots, l$, то по відношенню до наступного елементу $u_{i+l+1} = u^0$, де u^0 належить до терм-множини T , говорять, що l -конфігурація переходить у стан u^0 , тобто в лінгвістичну змінну u_{i+l+1} , яка співпадає з термом u^0 .

При побудові клітинного автомату для даного ЛЧР спочатку формують множину M всіх l -конфігурацій, які містяться в ЛЧР, а потім обчислюються частоти переходів l -конфігурацій M в терми з T , тобто будується пам'ять клітинного автомату.

Частотна статистика переходів l -конфігурацій в певний стан u^r із терм-множини T формується наступним чином. Для кожної l -конфігурації g лінгвістичного часового ряду U підраховуємо кількість її переходів в кожен із станів терм-множини T , після чого сумуємо їх. Частота переходу l -конфігурації

g в певний стан u^r обчислюється шляхом ділення кількості переходів даної конфігурації g в даний стан u^r на загальну кількість переходів конфігурації g :

$$w(g, u^r) = W(g, u^r) / \sum_{r=1}^k W(g, u^r). \quad (1.35)$$

Етап 3. *Прогнозування на базі клітинного автомату*. Прогноз терму u_{n+1} представляється у вигляді нечіткої терм-множини (НТМ): $U_{n+1} = \{(u^1; \mu_1), (u^2; \mu_2), \dots, (u^k; \mu_k)\}$, де значення функції належності задовольняють рівності $\mu_1 + \mu_2 + \dots + \mu_k = 1$. Значення $\mu_1, \mu_2, \dots, \mu_k$ обчислюються через значення частот переходів, які отримуються для різних l - конфігурацій з послідовності $u_{n-L+1}u_{n-L+2}\dots u_{n-1}u_n$, де L – глибина пам'яті ряду. Спочатку обчислюються ненормовані значення:

$$\begin{aligned} \mu'_1 &= w(u_n, u^1) + w(u_{n-1}u_n, u^1) + \dots + w(u_{n-L+1} \dots u_{n-1}u_n, u^1), \\ \mu'_2 &= w(u_n, u^2) + w(u_{n-1}u_n, u^2) + \dots + w(u_{n-L+1} \dots u_{n-1}u_n, u^2), \\ &\dots \\ \mu'_k &= w(u_n, u^k) + w(u_{n-1}u_n, u^k) + \dots + w(u_{n-L+1} \dots u_{n-1}u_n, u^k), \end{aligned}$$

і їх сума: $s = \mu'_1 + \mu'_2 + \dots + \mu'_k$. Після нормування обчислених значень отримуємо:

$$\mu_1 = \frac{\mu'_1}{s}, \mu_2 = \frac{\mu'_2}{s}, \dots, \mu_k = \frac{\mu'_k}{s}. \quad (1.36)$$

Так як ЛЧР U отриманий в результаті застосування алгоритма класифікації до деякого числового ЧР Z , НТМ можна перетворити в нечітку (числову) множину $Z_{n+1} = \{(z^1; \mu_1), (z^2; \mu_2), \dots, (z^k; \mu_k)\}$. Числові значення елементів z^1, z^2, z^k отримуються різним шляхом в залежності від алгоритму перетворення числового ЧР в лінгвістичний ЧР. При використанні методу ковзного вікна використовується наступна формула:

$$z^i = (z_{max} - z_{min}) \cdot \frac{i-1}{k} + z_{min} + \frac{z_{max} - z_{min}}{2}, \quad (1.37)$$

де z_{max} і z_{min} – відповідно максимальне і мінімальне значення ЧР в останньому ковзному вікні. Нечіткий числовий прогноз Z_{n+1} переводиться в чітке прогнозне значення z_{n+1} шляхом застосування до Z_{n+1} операції дефазицізації:

$$z_{n+1} = z^1 \cdot \mu_1 + z^2 \cdot \mu_2 + \dots + z^k \cdot \mu_k. \quad (1.38)$$

Етап 4. *Оцінка точності прогнозу*. Формули для оцінки точності прогнозу наведені в пункті 1.4.1.

1.4 Прогнозування часових рядів кількості абітурієнтів

1.4.1 Критерії точності прогнозів

У стандартній статистиці для міри точності прогнозу частіше використовуються такі характеристики [23]:

Середня абсолютна процента помилка **MAPE** (Mean Absolute Percentage Error):

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|X_t - \hat{X}_t|}{X_t} \cdot 100\%, \quad (1.39)$$

де X_t – реальне значення;

$\hat{X}_t$ – прогнозоване значення;

N – кількість прогнозів.

Середнє абсолютне відхилення **MAD** (Mean Absolute Deviation) :

$$MAD = \frac{1}{N} \sum_{t=1}^N |X_t - \hat{X}_t|. \quad (1.40)$$

Середня процентна помилка **MPE** (Mean Percentage Error) :

$$MPE = \frac{1}{N} \sum_{t=1}^N \frac{X_t - \hat{X}_t}{X_t} \cdot 100\%. \quad (1.41)$$

Середня помилка **ME** (Mean Error) :

$$ME = \frac{1}{N} \sum_{t=1}^N (X_t - \hat{X}_t). \quad (1.42)$$

Середньоквадратична помилка **MSE** (Mean Squared Error) :

$$MSE = \frac{1}{N} \sum_{t=1}^N (X_t - \hat{X}_t)^2. \quad (1.43)$$

Корінь із середньоквадратичної помилки **RMSE** (Root Mean Squared Error)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N (X_t - \hat{X}_t)^2}. \quad (1.44)$$

Так як в прогнозуванні курсу валют велике значення має напрям зміни курсу, то доцільно використовувати також таку міру точності прогнозу як відсоток прогнозів знаку приросту, які справилились P .

$$P = \frac{m}{N} \cdot 100\%, \quad (1.45)$$

де

$$m = \sum_{i=2}^N \begin{cases} 1, & \text{якщо } \frac{|X_i - X_{i-1}|}{X_i - X_{i-1}} = \frac{|\hat{X}_i - X_{i-1}|}{\hat{X}_i - X_{i-1}} \\ 0, & \text{якщо } \frac{|X_i - X_{i-1}|}{X_i - X_{i-1}} \neq \frac{|\hat{X}_i - X_{i-1}|}{\hat{X}_i - X_{i-1}} \end{cases}$$

1.4.2 Модель Хольта та адаптивного згладжування Брауна

Об'єктом дослідження є часові ряди послідовних щоденних даних (спостережень) про курси тої чи іншої валюти по відношенню до американського долару, які записуються як $X_1, X_2, \dots, X_N$, де N – довжина ряду. В якості статистичної бази дослідження взяті дані про курси п'яти валют: канадського долару, євро, британського фунту, японської ієни та швейцарського франку. Довжина кожного часового ряду була довільно прийнята рівною 200 щоденним спостереженням за період з 28 квітня 2009 року по 14 грудня 2009 рік. Дані були отримані за допомогою інформаційної торгової платформи MetaTrader.

Проведемо дослідження моделей Хольта та адаптивного згладжування Брауна, які описані в підрозділі 1.3.2.1. За першими 30 рівнями часових рядів, які досліджуються, знайдемо оптимальні значення коефіцієнтів згладжування і довжини бази згладжування. Для цього будемо перебирати коефіцієнти згладжування α, β, γ від 0 до 1 з кроком 0.01 та значення довжини бази згладжування k від 3 до 29. Для кожної можливої комбінації на основі ряду $X_1, X_2, \dots, X_k$, зробимо прогноз на момент часу $k+1$ і знайдемо помилку прогнозу e_1

як абсолютну різницю між фактичним значенням і прогнозованим. Далі базу згладжування зсунемо на крок вперед, отримаємо ряд $X_2, X_3, \dots, X_{k+1}$; зробимо прогноз на момент часу $k+2$ і знайдемо помилку прогнозу e_2 . Всього таким чином отримаємо $30-k$ помилок прогнозу. Знайдемо середню відносну помилку для всіх можливих комбінацій та оберемо такі значення коефіцієнтів згладжування та довжини бази згладжування, при яких помилка найменша. З обраними вище параметрами зробимо прогноз рівня X_{31} . При цьому в якості початкового значення S_0 приймемо перший член ряду; початкове значення коефіцієнту швидкості зростання/спадання T_0 візьмемо рівним нулю. Потім зробимо зсув на один крок уперед і виконаємо ті ж самі дії, тобто за рівнями від 2-го до 31-го знайдемо оптимальні значення зазначених параметрів і спрогнозуємо рівень X_{32} і т.д. Всього отримаємо 170 прогнозів.

Підрахуємо середню відносну помилку прогнозування MAPE за формулою (1.39) та знайдемо максимальну відносну помилку прогнозування. Також підрахуємо відсоток прогнозів знаку приросту, які справдились P за формулою (1.45).

Результати прогнозування наведені в таблицях 1.2-1.3 та на рисунках 1.1-1.2.



Рисунок 1.1 – Результати прогнозування кількості абітурієнтів за методом S-Модель

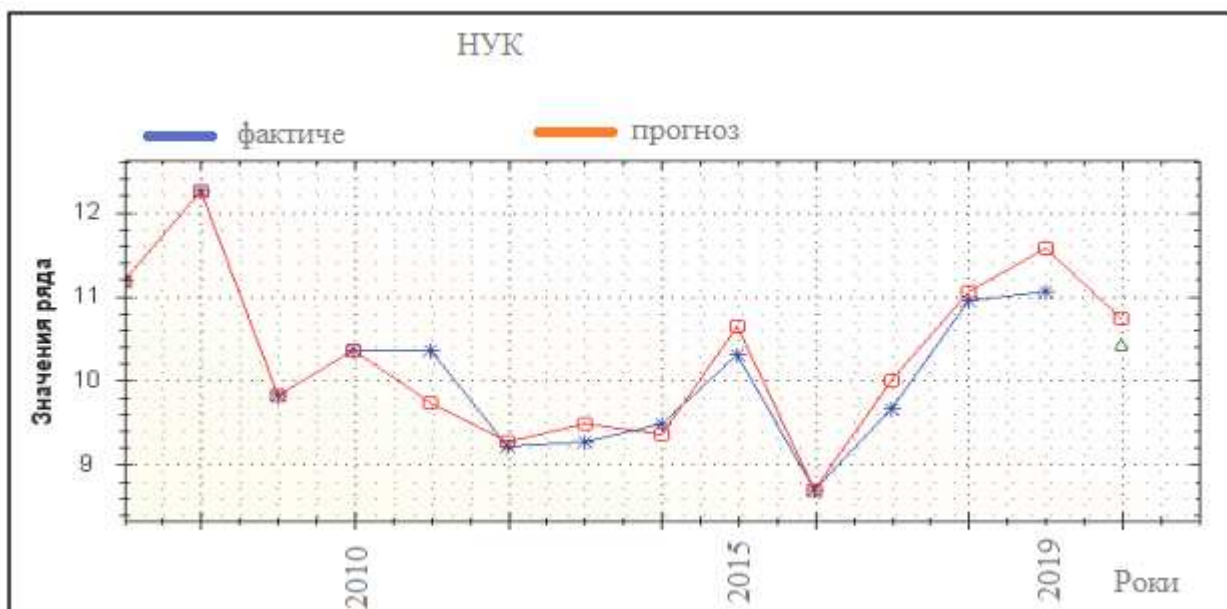


Рисунок 1.2 – Результати прогнозування кількості абітурієнтів за методом Т-Моделі.

Таблиця 1.2 – Результати прогнозування за методами.

Модель	TTend%	MAPE %
Т-модель	2,4	12,8
S-модель	15,6	12,4
D-модель	28,2	18,2
ИНС-модель	27,7	13,4

На основі отриманих результатів дослідження можна зробити висновок, що прогнозування за методом Хольта дає меншу відносну помилку ніж при прогнозуванні за методом адаптивного згладжування Брауна. Проте при застосуванні методу адаптивного згладжування Брауна маємо дещо більший процент прогнозів знаку приросту, які справдились.

1.4.3 Модель Бокса-Дженкінса ARIMA

Об'єктом дослідження є часові ряди послідовних щоденних даних (спостережень) про курси тої чи іншої валюти по відношенню до американського долару, які записуються як $X_1, X_2, \dots, X_N$, де N – довжина ряду. В якості статистичної бази дослідження взяті дані про курси п'яти валют: канадського долару, євро, британського фунту, японської ієни та швейцарського

франку. Довжина кожного часового ряду була довільно прийнята спостереженням за період з 01 вересня 2010 року по 30 червня 2019 року. Дані були отримані за допомогою інформаційної торгової платформи MetaTrader .

Розглянемо часовий ряд кількості абітурієнтів.



Рисунок 1.3 – Графік часового ряду кількості абітурієнтів

Аналізуючи графік часового ряду, зображеного на рисунку 1.3, можна відмітити, що в ньому присутній зростаючий тренд.

Для подальшого аналізу даного часового ряду побудуємо його автокореляційну функцію за допомогою програми Minitab (рисунок 1.4).

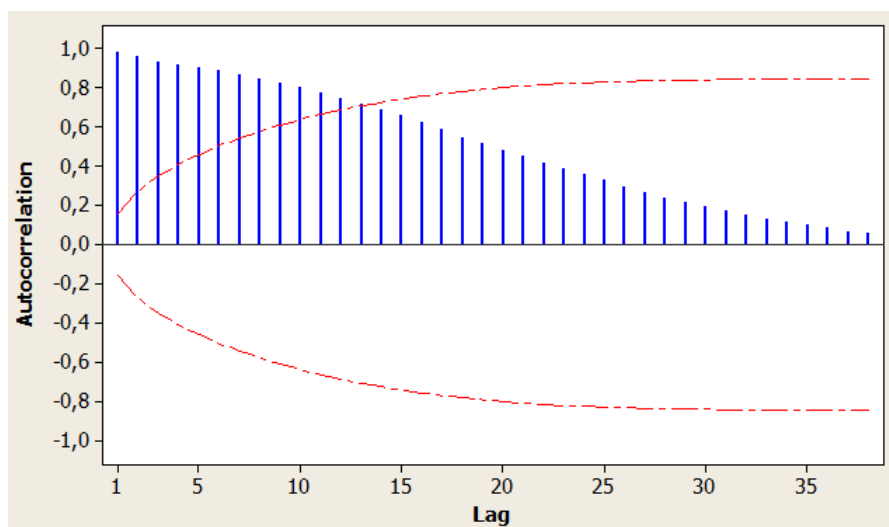


Рисунок 1.4 – Графік автокореляційної функції часового ряду кількості абітурієнтів

Аналізуючи графік автокореляційної функції часового ряду кількості абітурієнтів, зображеного на рисунку 2, можна відмітити, що декілька перших коефіцієнтів автокореляції постійно мають велике значення і прямують до нуля дуже повільно. Тому початкові висновки про присутність тренду були вірними, а також те, що даний часовий ряд є нестационарним.

Для видалення тренду візьмемо різницю з лагом 1 (рисунок 1.5).

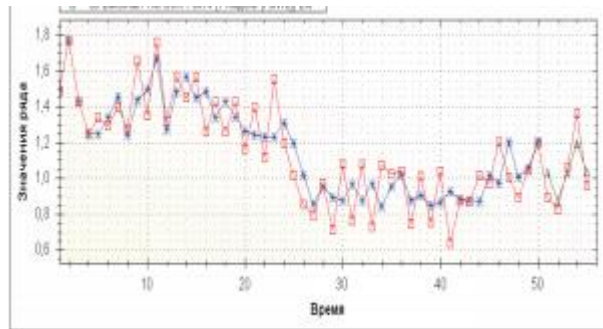


Рисунок 1.5 – Графік перших різниць часового ряду

На графіку перших різниць часового ряду кількості абітурієнтів (рисунок 1.5) всі зміни відбуваються в околиці визначеного фіксованого рівня. Вибірковим середнім є $7 \cdot 10^{-4}$.

Розглянемо графіки автокореляційної та приватної автокореляційної функцій перших різниць часового ряду кількості студентів (рис. 1.6-1.7).

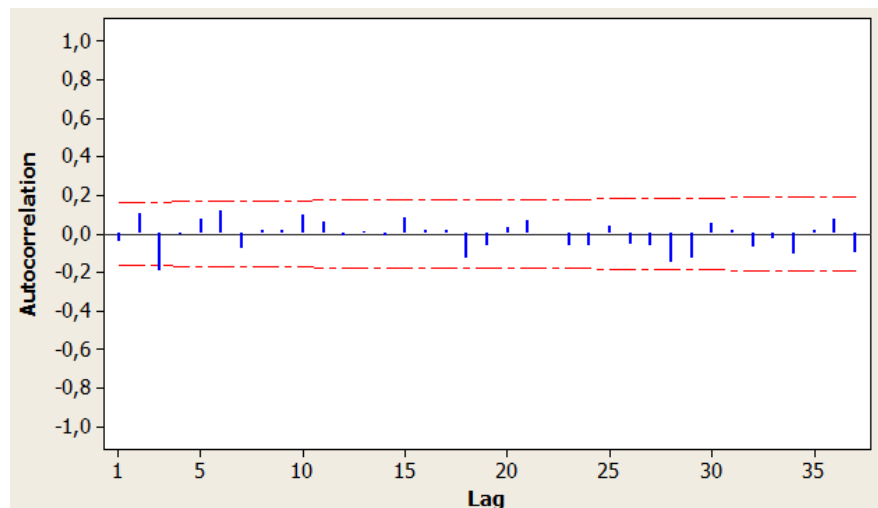


Рисунок 1.6 – Графік автокореляційної функції перших різниць часового ряду кількості студентів

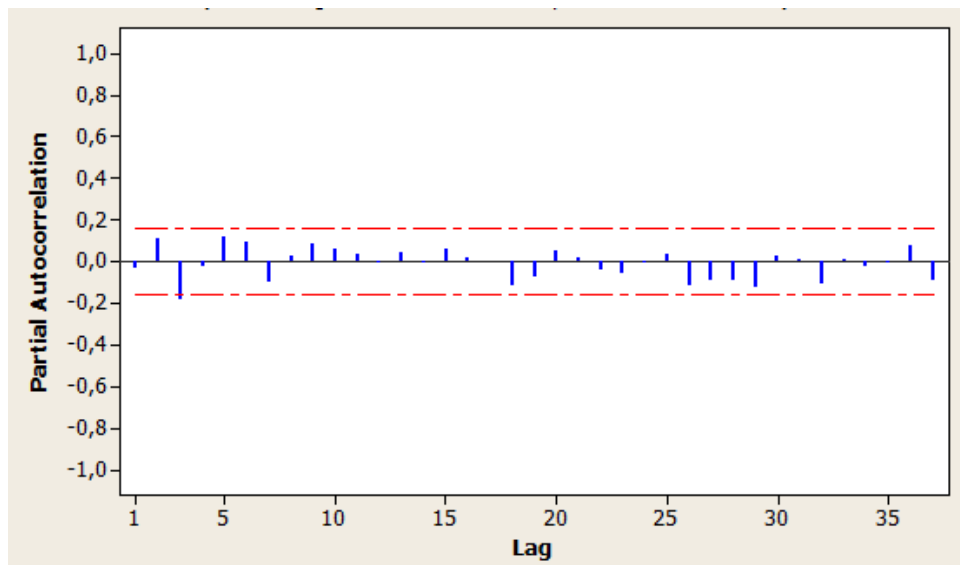


Рисунок 1.7 – Графік приватної автокореляційної функції перших різниць часового ряду кількості студентів

Аналіз вибірових автокореляційної і приватної автокореляційної функцій перших різниць часового ряду кількості студентів дозволяє стверджувати, що взяття перших різниць перетворює початковий ряд в стаціонарний. Тому в моделі ARIMA параметр d дорівнює 1. Також можна стверджувати, що обидва параметри p і q моделі ARIMA мають нульові значення (тобто ряд різниць рівнів часового ряду кількості студентів можна вважати реалізацією процесу «білого шуму»). Відповідно, прогнозним значенням для ряду перших різниць є його вибірове середнє.

Аналогічним чином були дослідженні інші часові ряди. Їх дослідження також показало, що вони задовольняють моделі ARIMA(0,1,0). Таким чином прогноз здійснюється на основі вибірового середнього попередніх приростів курсу валют

$$\hat{X}_{i+1} = X_i + \frac{X_i - X_{i-n}}{n}, \quad (1.46)$$

де n – кількість попередніх приростів курсу валют, на основі яких здійснюється прогноз; $\hat{X}_{i+1}$ – прогнозне значення; X_i – реальне значення.

Таким чином проведемо дослідження аналогічне до дослідження, яке здійснювалось для моделей Хольта та адаптивного згладжування Брауна.

Для дослідження моделі ARIMA(0,1,0) нам знадобиться ще 100 історичних значень. Отримаємо за допомогою інформаційної торгової платформи MetaTrader часові ряди довжиною 300 щоденних спостережень починаючи з 1 вересня 1997 року і закінчуючи 30 червня 2019 року.

Проведемо дослідження наступним чином. За першими 22 рівнями часових рядів, які досліджуються, знайдемо оптимальне значення кількості попередніх приростів кількості студентів n . Для цього будемо перебирати значення n від 10 до 100 з кроком 10. Для кожного можливого значення n на основі ряду $X_1, X_2, \dots, X_k$, де $k = n+1$, зробимо прогноз на момент часу $k+1$ і знайдемо помилку прогнозу e_1 як абсолютну різницю між фактичним значенням і прогнозованим. Далі на основі ряду $X_2, X_3, \dots, X_{k+1}$ зробимо прогноз на момент часу $k+2$ і знайдемо помилку прогнозу e_2 . Всього таким чином отримаємо $130-k$ помилок прогнозу. Знайдемо середню відносну помилку та оберемо таке значення n , при якому помилка найменша. З обраним вище n зробимо прогноз рівня X_{131} . Потім зробимо зсув на один крок уперед і виконаємо ті ж самі дії, тобто за рівнями від 2-го до 131-го знайдемо оптимальне значення n і спрогнозуємо рівень X_{132} і т.д. Всього отримаємо 170 прогнозів.

Результати прогнозування наведені в таблиці 1.4 та на рисунку 1.8.

Таблиця 1.4 – Результати прогнозування за ARIMA

Модель	Класс процесса	Прогноз на 1	Прогноз на 10%	Прогноз на 1	Прогноз на 10%
		TTend%	TTend%	MAPE %	MAPE %
Т-модель	ARIMA	2,2	8,45	6,3	7,05
S-модель	ARIMA	35,68	27,47	19,21	33,7
D-модель	ARIMA	34,95	38,18	10,76	13,15
НС-модель	ARIMA	21,3	27,66	14,34	36,51

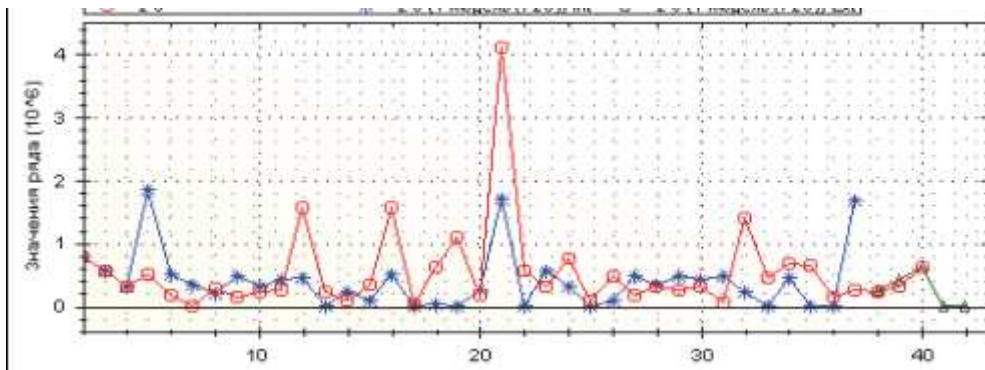


Рисунок 1.8 – Результати прогнозування кількості студентів за методом ARIMA

На основі отриманих результатів дослідження можна зробити висновок, що прогнозування за методом ARIMA дало меншу відносну помилку ніж при прогнозуванні методами адаптивного згладжування Брауна та Хольта. Проте при застосуванні методу ARIMA маємо гірший процент прогнозів знаку приросту, які справдились.

1.4.4 Модель клітинних автоматів

Об'єктом дослідження є часові ряди послідовних щоденних даних (спостережень) про курси тої чи іншої валюти по відношенню до американського долару, які записуються як $X_1, X_2, \dots, X_N$, де N – довжина ряду. В якості статистичної бази дослідження взяті дані про курси п'яти валют: канадського долару, євро, британського фунту, японської ієни та швейцарського франку. Довжина кожного часового ряду була довільно прийнята рівною 300 щоденним спостереженням за період з 31 грудня 2008 року по 14 грудня 2009 рік. Дані були отримані за допомогою інформаційної торгової платформи MetaTrader .

За допомогою послідовного R/S-аналізу, який описано в підрозділі 1.3.3, була оцінена глибина пам'яті даних часових рядів l (див. таблицю 1.5).

Кількість історичних даних i для формування пам'яті клітинного автомату візьмемо рівною 130. Довжина вікна m при перетворенні ЧЧР в ЛЧР підбирається шляхом перебору від l до $130/2$ з кроком $l/2$. Кількість термів k

також підбирається шляхом перебору від 3 до 10 з кроком 1. Блок-схема перебору приведена в 3 розділі.

Отримаємо 170 прогнозів наступним чином. Для ряду $X_1, X_2, \dots, X_i$ знайдемо оптимальні m і k і зробимо прогноз на момент часу $i+1$. Потім базу історичних даних зсунемо на крок вперед і отримаємо ряд $X_2, X_3, \dots, X_{i+1}$, на основі якого зробимо прогноз на момент часу $i+2$. Будемо повторювати ці дії поки не досягнемо кінця ряду X , в результаті чого отримаємо 170 прогнозів.

Результати прогнозування наведені в таблиці 1.5 та на рисунку 1.9.

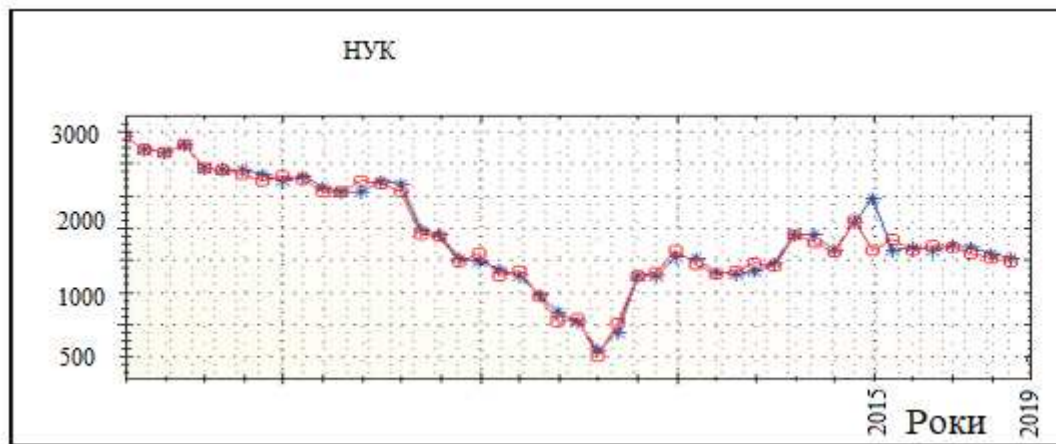


Рисунок 1.9 – Результати прогнозування кількості студентів за методом клітинних автоматів

Таблиця 2.5 – Результати прогнозування за методом клітинних автоматів.

Моделі	Chen	Huarng (2001a)	Huarng (2001b)	Lee и др.	K-based	T-модель
СКО	98,3308	88,6369	73,7399	36,94	61,13	32
MAPE	1,05%	1,03%	0,89%	0,82%	0,72%	0,36%

1.5 Аналіз отриманих результатів та вибір методу прогнозування

Побудуємо гістограми з середньою відносною помилкою та процентом вірно спрогнозованих знаків приросту для порівняння методів прогнозування. Під номером 1 вказується модель Хольта, 2 – модель адаптивного згладжування Брауна, 3 – модель ARIMA(0,1,0), 4 – модель клітинних автоматів.



Рисунок 1.10 – Процент вірно спрогнозованих знаків приросту для кожного з методів



Рисунок 1.11 – Середня відносна помилка, для кожного з методів

Аналізуючи рисунки 1.10 та 1.11 можна зробити висновок, що найкращим методом по обом критеріям (процент вірно спрогнозованих знаків приросту та середня відносна помилка) є метод клітинних автоматів.

Низьку середню відносну помилку дає метод ARIMA, проте він дає незадовільний результат з точки зору прогнозування знаків приросту.

Метод Хольта дає меншу відносну помилку ніж при прогнозуванні за методом адаптивного згладжування Брауна. Проте при застосуванні методу адаптивного згладжування Брауна маємо дещо більший процент прогнозів знаку приросту, які справдились.

Введемо один критерій для вибору найкращого методу прогнозування, який побудуємо на основі зазначених вище критеріїв. Підрахунок критерія F для кожного методу підраховується за наступною формулою:

$$F_i = \left| \frac{\bar{P}_i}{\sum_{n=1}^4 \bar{P}_n} \right| + \left| \frac{MAPE_i}{\sum_{n=1}^4 MAPE_n} \right|,$$

де $\bar{P} = (1 - P)$ – процент невірно спрогнозованих знаків приросту; $MAPE$ – середня відносна помилка.

Найкращим є той метод, який має найменше значення F . Підрахуємо для кожного методу значення критерію F та відобразимо його на гістограмі (рисунок 1.12).

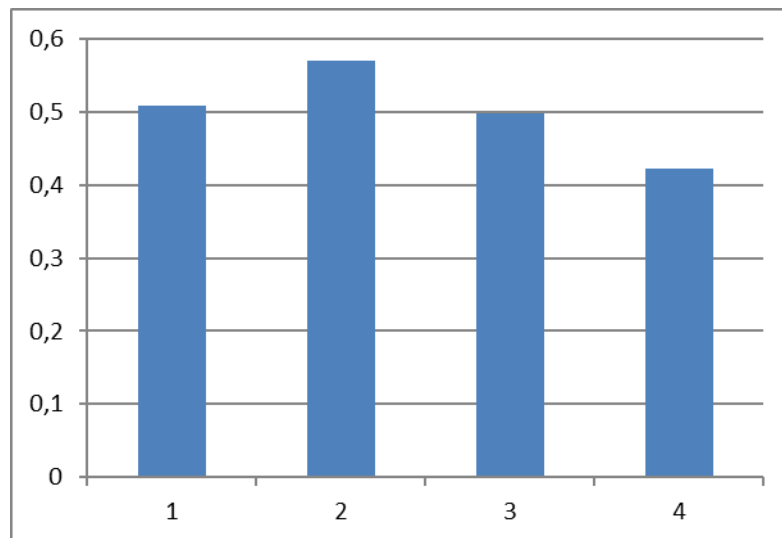


Рисунок 1.12 – Критерій F вибору методу прогнозування

Аналізуючи рисунок 1.12 визначимо порядок методів прогнозування від найкращого до найгіршого: метод клітинних автоматів, метод ARIMA, метод Хольта, метод адаптивного згладжування Брауна.

Таким чином, для розробки підсистеми короткострокового прогнозування курсу валют будемо використовувати метод клітинних автоматів.

1.6 Висновки

В даному розділі було проведено дослідження класичних статистичних методів аналізу та прогнозування часових рядів, таких як метод Хольта, метод адаптивного згладжування Брауна, метод ARIMA, та методу на основі інструментарію клітинних автоматів.

Для вибору найкращого методу використовувався критерій, який підраховується як сума модулів нормованих значень середньої відносної помилки та проценту невірно спрогнозованих знаків приростів. Найкращим виявився метод S-Моделі, так як він має найменше значення критерію.

На другому місці по точності прогнозування стоїть метод ARIMA.

Методи Хольта та адаптивного згладжування Брауна, які засновані на експоненціальному згладжуванні, дали найгіршу точність серед методів, які досліджувалися.

Таким чином при розробці підсистеми короткострокового прогнозування курсу валют будемо використовувати метод S-Моделі, оскільки він виявився найкращим.

РОЗДІЛ 2 ФОРМУВАННЯ ВИМОГ ДО ПІДСИСТЕМИ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ КІЛЬКОСТІ АБІТУРІЄНТІВ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Формування вимог до підсистеми короткострокового прогнозування кількості абітурієнтів

2.2.1 Аналіз недоліків існуючих підсистем прогнозування кількості абітурієнтів та обґрунтування необхідності удосконалення

Робота присвячена моделям нечітких процесів для прогнозування кількості абітурієнтів. Дана проблема актуальна і має важливе значення, так як від її адекватного вирішення залежить обсяг фінансування вузу. Існує багато підходів прогнозування кількості абітурієнтів. Однак, тимчасові ряди, що містять відомості про вступників до ВНЗ, як правило, характеризуються невеликою довжиною, нестационарністю поведінки, що ускладнює побудову статистичних моделей і моделей на основі штучних нейронних мереж для цілей прогнозу. Також, традиційні моделі не можуть бути застосовані, коли історичні дані є лінгвістичними значеннями. Нечіткі тимчасові ряди є ефективним інструментом для вирішення таких проблем. Як додаток нечітких часових рядів наводиться приклад прогнозування абітурієнтів з Університету НУК.

Підсистеми короткострокового прогнозування кількості абітурієнтів надають своєчасну і надійну інформацію, яка дозволяє покращити процес прийняття рішень по формуванню бюджету та витратам.

На даний час існує багато програмних продуктів по прогнозуванню ринка, зокрема кількості абітурієнтів, проте всі вони ґрунтуються на методах технічного аналізу. Як приклад, можна привести такі системи: TRADER, ELLIOTT WAVE ANALYSER PROFESSIONAL 6.0., AINET, PARITY 1.5.

До недоліків існуючих систем короткострокового прогнозування кількості абітурієнтів можна віднести наступні:

- 1) Відсутність кросплатформенності. Більшість систем працює лише в операційній системі Windows.

2) Відсутність розширюваності:

- а) Неможливість використовувати різні джерела даних для прогнозування. Котирування валют можна отримувати з зовнішньої бази даних, web-ресурса або певного файлу.
- б) Неможливість нарощувати функціональність за рахунок додавання зовнішніх програмних модулів, які реалізують додаткові методи прогнозування.

3) Відсутність можливості продивитись історію прогнозів і порівняти реальні та прогнозовані дані.

Таким чином на даному етапі постає необхідність розробки підсистеми короткострокового прогнозування курсів валют, яка буде позбавлена зазначених вище недоліків.

2.2.2 Мета, критерії та обмеження створення підсистеми короткострокового прогнозування кількості абітурієнтів

Підсистема короткострокового прогнозування кількості абітурієнтів призначена для автоматизації процесу прогнозування, скорочення часу обробки даних й одержання більш якісного прогнозу (мінімізація помилки прогнозування).

Метою створення підсистеми короткострокового прогнозування кількості абітурієнтів є підвищення ефективності роботи та підвищення якості надання освітніх послуг. Вона повинна надавати своєчасну і надійну інформацію (прогноз), яка дозволяє покращити процес прийняття рішень витратам.

Підсистема короткострокового прогнозування кількості абітурієнтів може використовуватись керівними установами.

Основні критерії створення підсистеми короткострокового прогнозування кількості абітурієнтів :

- Соціальний критерій. Підсистема розрахована на користувачів з різною комп'ютерною підготовкою. Необхідний зручний та інтуїтивно зрозумілий інтерфейс роботи з нею.

- Економічний критерій. Підсистема буде свободо розповсюджуваною (freeware).
- Критерій безпеки. Підсистема повинна бути безпечною з погляду збереження і обробки даних.

На підсистему короткострокового прогнозування кількості абітурієнтів необхідно накласти деякі обмеження.

Так як підсистема короткострокового прогнозування кількості абітурієнтів може бути впроваджена в певну систему, то слід відзначити, що її впровадження не повинно приводити істотних змін у роботі модулів даної системи. Зміни в технічній базі системи повинні бути такими щоб не знизити якість, швидкість та надійність роботи системи.

2.2.3 Функції та задачі створюваної підсистеми короткострокового прогнозування кількості абітурієнтів

Підсистема повинна виконувати наступні задачі:

- прогнозування певного кількості абітурієнтів на наступну одиницю часу (рік);
- вести історію прогнозів.

Основними функціями підсистеми короткострокового прогнозування кількості абітурієнтів є :

- імпорт даних (з веб-ресурсу, зі зовнішнього файлу, з БД);
- побудова графіків прогнозу й похибки прогнозу кількості абітурієнтів и;
- ведення історії прогнозів кількості абітурієнтів .

2.3 Розробка концепції підсистеми короткострокового прогнозування кількості абітурієнтів

2.3.1 Опис результатів вивчення об'єкта автоматизації

Підсистема короткострокового прогнозування кількості абітурієнтів призначена для автоматизації процесу прогнозування. Вона створюється з метою підвищення ефективності надання освітніх послуг.

Підсистема повинна надавати своєчасну і надійну інформацію (прогноз), яка дозволяє покращити процес прийняття рішень якості надання послуг.

Підсистема короткострокового прогнозування кількості абітурієнтів повинна виконувати наступні задачі:

- прогнозування певної кількості абітурієнтів на наступний проміжок часу (рік);
- вести історію прогнозів.

Для створення підсистеми потрібна розробка бази даних підсистеми, інтерфейсу користувача та моделювання апарату прогнозування на основі Т-Моделі.

2.3.2 Аналіз вимог до підсистеми короткострокового прогнозування кількості абітурієнтів

Крім функціональних вимог до підсистеми короткострокового прогнозування кількості абітурієнтів необхідно врахувати вимоги до підсистеми в цілому:

- зручний інтерфейс користувача;
- висока масштабованість;
- розширюваність;
- надійність;
- висока продуктивність.

Аналізуючи зазначені вимоги, можна зробити висновок, що підсистема повинна мати/підтримувати:

- сучасне програмне забезпечення;
- сучасні рішення в області інформаційних технологій;
- ефективні методи доступу до даних;
- забезпечити простоту модернізації, можливість повторного використання програмного забезпечення (коду).

Враховавши результати вивчення об'єкта автоматизації необхідно провести порівняльний аналіз можливих концепцій підсистеми й оцінити переваги і недоліки кожної з них на підставі критеріїв і вимог користувача.

2.3.3 Розробка концепції підсистеми

Виходячи з вимог наведених у попередніх пунктах, необхідно обрати найбільш ефективну концепцію підсистеми, що розроблюється. На Рисунку 2.1 наведено концептуальну модель підсистеми короткострокового прогнозування кількості абітурієнтів, котра задовольняє вимогам попередніх пунктів.

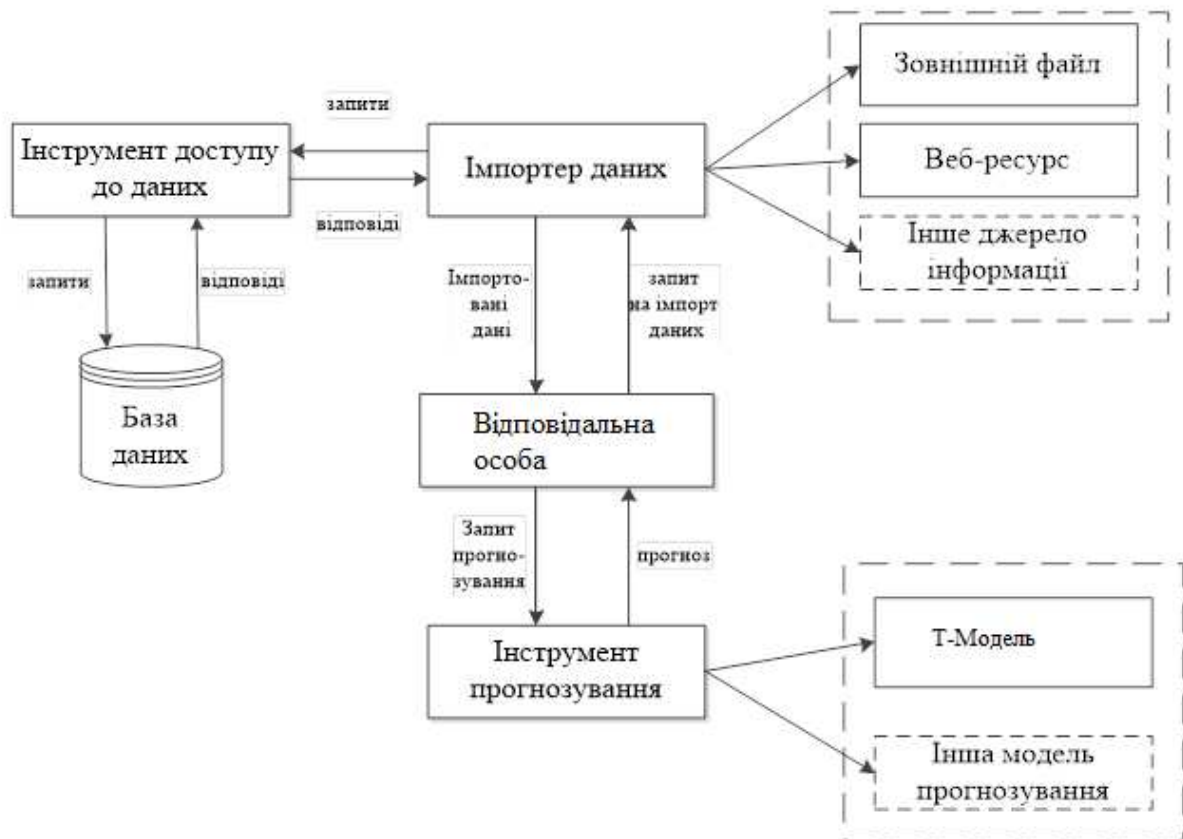


Рисунок 2.1 – Концептуальна модель підсистеми

Інформаційна підсистема короткострокового прогнозування кількості абітурієнтів складається з наступних частин:

Імпортёр даних. Основна функція – здійснювати імпорт даних з обраного джерела інформації.

Інструмент прогнозування. Здійснює прогноз одним із обраних методів, зокрема на основі моделі клітинних автоматів.

Інструмент доступу до даних. Відповідає за збереження та надання за запитом необхідної інформації.

Менеджер. Організує взаємодію вищезазначених компонентів підсистеми.

2.3.4 Вибір варіанту реалізації концепції підсистеми за допомогою функціонально-вартісного аналізу

Для вибору найбільш ефективного варіанту реалізації концепції підсистеми короткострокового прогнозування кількості абітурієнтів використовується метод функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА, Activity Base Costing, ABC) – метод визначення вартості та інших характеристик виробів, послуг і споживачів, що використовують як основу функції та ресурси, що задіяні у виробництві, маркетингу, продажі, доставці, технічній підтримці, наданні послуг, обслуговуванні клієнтів, а також забезпечення якості. В основі методу функціонального аналізу лежить функціональний підхід, відповідно до якого об'єктом аналізу є не сам продукт (система, підсистема, програмне забезпечення), а функції, що він виконує.

ФВА – один з методів, що дозволяє вказати на можливі шляхи поліпшення вартісних показників. Ціль створення ФВА – моделі для удосконалення діяльності підприємств – досягти поліпшення у роботі підприємств по показниках вартості, трудомісткості та продуктивності. Проведення розрахунків по ФВА – моделі дозволяє отримати великий обсяг ФВА-інформації для ухвалення рішення.

Функціонально-вартісний аналіз складається з двох основних етапів:

- функціональний аналіз;
- вартісний аналіз.

1) Функціональний аналіз

Виходячи з вимог до компонентів концепції підсистеми короткострокового прогнозування кількості абітурієнтів та засобів розробки приведемо можливі варіанти реалізації підсистеми (дивись таблиця 2.1).

У таблиці наведено дві апаратні конфігурації, при цьому наводити більш продуктивні конфігурації немає сенсу. Щодо програмного забезпечення то можна зазначити, що усі наведені операційні системи є безкоштовними, операційна система MS Windows 7/XP була отримана за ліцензією MSDN AA.

Таблиця 2.1 – Варіанти засобів розробки

Варіант	Технічне забезпечення	Програмне забезпечення		
		Операційна система	Мова	СУБД
1	CPU: AMD Athlon X2 Dual Core 2ГГц RAM: 1Gb HDD: 160 Gb	MS Windows XP	C++	Embedded Firebird
2				SQLite
3			Java	Apache Derby
4				H2
5			C#	SQLite
6				SQL Server Compact
7		Ubuntu 10.10	C++	Embedded Firebird
8				SQLite
9			Java	Apache Derby
10				H2
11	CPU: Intel Core 2 Duo P8700 2,57 ГГц RAM: 3Gb HDD: 250Gb	MS Windows 7	C++	Embedded Firebird
12				SQLite
13			Java	Apache Derby
14				H2
15			C#	SQLite
16				SQL Server Compact

У Таблиці 2.2 наведено експертну оцінку наведених варіантів за критеріями.

Таблиця 2.2 Експерта оцінка варіантів реалізації

Варіант	Функціональність	Надійність	Адміністрування	Продуктивність	Ефективність
1	8	7	7	8	8
2	9	8	8	9	8
3	8	8	8	8	9
4	10	9	8	10	8
5	7	8	8	7	9
6	7	7	6	8	8
7	9	8	10	10	8
8	8	7	9	8	8
9	9	8	9	8	10
10	9	9	8	10	9
11	8	10	8	8	9
12	9	8	9	7	8
13	8	9	7	8	9
14	10	10	9	10	10
15	7	9	9	8	8
16	7	9	8	10	8

Наступним етапом є визначення вагових коефіцієнтів для кожного з параметрів. У Таблиці 2.3 наведені вагові коефіцієнти для описаних критеріїв, значення коефіцієнтів обираються так щоб сума усіх коефіцієнтів дорівнювала одиниці.

Таблиця 2.3 – Значення вагових коефіцієнтів

Параметр, F_i	Коефіцієнт, φ_i
Функціональність	0,2
Надійність	0,25
Адміністрування	0,15
Продуктивність	0,2
Ефективність	0,2

Обчислимо коефіцієнт технічного рівня для кожного з варіантів за формулою:

$$K[F_i] = \sum_{j=1}^n \varphi_{ij} B_j.$$

У таблиці 2.4 наведено результати обчислення коефіцієнтів по кожному з варіантів.

Таблиця 2.4 – Коефіцієнти технічного рівня для кожного з варіантів

Варіант	Коефіцієнт технічного рівня
1	7,6
2	8,4
3	8,2
4	9,05
5	7,8
6	7,25
7	8,9
8	7,9
9	8,75
10	9,05
11	8,7
12	8,15
13	8,3
14	9,85
15	8,2
16	8,45

Виходячи з отриманих результатів можна судити о ступені технічного рівня кожного з варіантів. Якнайкращим є той варіант, якому відповідає найбільший коефіцієнт технічного рівня.

Таким чином можна сказати, що 14-ий варіант є найкращим з технічної точки зору:

– Апаратна частина:

CPU: Intel Core 2 Duo P8700 2,57 ГГц;

RAM: 3Gb;

HDD: 250Gb.

– Програмна частина:

Операційна система: MS Windows 7;

Мова програмування: Java;

СУБД: H2.

2) Вартісний аналіз

Аналіз функцій, що реалізуються підсистемою, повинен бути доповнений вартісним аналізом. Для цього розглядаються зведені витрати Z_j по кожному j -ому варіанту.

$$Z_j = C_j + E_n K_j,$$

де C_j – поточні витрати по j -ому варіанту;

K_j – капітальні вкладення по j -ому варіанту;

E_n – нормативний коефіцієнт економічної ефективності капітальних вкладень, що дорівнює 0,15.

Зведені витрати Z_j розраховують у сфері виробництва (у розробника) чи в сфері експлуатації (у користувача). У нашому випадку доцільно розраховувати витрати у сфері виробництва.

Поточні витрати при розробці інформаційної системи(підсистеми) C_j – це собівартість створення системи. Для усіх варіантів будемо вважати, що собівартість створення системи однакова. Собівартість системи розрахована у Розділі 4 даної роботи та складає 12 000 грн.

Капітальні вкладення K_j – це витрати, що носять одноразовий характер. У даному випадку внесемо сюди вартість придбання ліцензійного програмного забезпечення та апаратного забезпечення, необхідного для створення системи.

Таким чином, приведемо вартості необхідні для подальшого обчислення:

Вартість апаратних а програмних рішень з необхідними коментарями наведена у Таблиці 2.5.

Відповідно до наведеної вартості можна обчислити капітальні вкладення та відповідно до них розрахувати зведені витрати по кожному варіанту. За результатами функціонального та вартісного аналізу можна розрахувати коефіцієнт техніко-економічного рівня по кожному з варіантів за формулою:

$$K_{TEP_i} = \frac{K_{TP_i}}{Z_i},$$

де K_{TEP} – показник технічного рівня для i -ого варіанту;

Z_i – зведені витрати для i -ого варіанту.

Результати розрахунків наведені у Таблиці 2

Таблиця 2.5 – Вартість апаратних або програмних рішень

Рішення	Вартість	Коментар
CPU: AMD Athlon X2 Dual Core 2ГГц RAM: 1Gb HDD: 160 Gb	2500	Апаратне рішення
CPU: Intel Core 2 Duo P8700 2,57 ГГц RAM: 3Gb HDD: 250Gb	3200	Апаратне рішення
MS Windows XP та MS Windows 7	0	Програмне рішення. Комерційна операційна система, що використовується у освітніх цілях. Ліцензія MSDN AA.
SQLite	400	Програмне рішення. Комерційна СУБД.
Embedded Firebird, Apache Derby, H2, SQL Server Compact	0	Програмне рішення. Безкоштовні СУБД.

Таблиця 2.6 – Коефіцієнти техніко-економічного рівня

Варіант	Вартість, грн	Зведені витрати, Z_i (грн)	Коефіцієнт, K_{TEP_i}
1	2500	12375	0,000614
2	2900	12435	0,000676
3	2500	12375	0,000663
4	2500	12375	0,000731
5	2900	12435	0,000627
6	2500	12375	0,000586
7	2500	12375	0,000719
8	2900	12435	0,000635
9	2500	12375	0,000707
10	2500	12375	0,000731
11	3200	12480	0,000697
12	3600	12540	0,000650
13	3200	12480	0,000665
14	3200	12480	0,000789
15	3600	12540	0,000654

16	3200	12480	0,000677
----	------	-------	----------

Найкращий варіант визначається за максимальним значенням коефіцієнту $K_{ТЕР_i}$. Таким чином, визначимо, що найкращий варіант під номером 14.

Отже підсумковий варіант конфігурації реалізації підсистеми має вигляд:

- Апаратна частина:
CPU: Intel Core 2 Duo P8700 2,57 ГГц;
RAM: 3Gb;
HDD: 250Gb.
- Програмна частина:
Операційна система: MS Windows 7;
Мова програмування: Java;
СУБД: H2.

2.4 Постановка задачі

Необхідно розробити підсистему короткострокового прогнозування кількості абітурієнтів . Підсистема короткострокового прогнозування кількості абітурієнтів призначена для автоматизації процесу прогнозування. Вона створюється з метою надання своєчасної та надійної інформації (прогноз), яка дозволяє покращити процес прийняття рішень по наданню освітніх послуг.

1) Вимоги до підсистеми

Ефективність. Підсистема повинна виконувати поставлену мету в заданих умовах використання і з визначеною якістю.

Масштабованість. Цю адекватність підсистема повинна зберегти при розвитку організаційної структури і зростанні інформаційного навантаження, не вимагаючи при цьому серйозної зміни архітектури системи.

Розширюваність. Підсистема повинна розвиватися, дозволяючи виключати і модифікувати старі, додавати нові компоненти.

Надійність. Збої в роботі критично важливих додатків, простій яких приводить до фінансових втрат, недопустимі і повинні фіксуватися системою для усунення подібних ситуацій.

Кросплатформеність. Можливість працювати ПЗ підсистеми під керівництвом різних операційних систем.

Збереження інвестицій. При модифікації підсистеми слід максимально використовувати раніше придбане і встановлене устаткування і наявні компоненти системи.

Економічна ефективність. Прибуток (безпосередній або непрямий), що одержується за рахунок використання ІС, повинен перевищувати витрати на створення і розвиток ІС.

Безпека. Необхідно забезпечити захист ІС від некоректних дій користувачів.

2) Вимоги до функціональних характеристик підсистеми

Підсистема короткострокового прогнозування кількості абітурієнтів повинна мати наступні функціональні характеристики:

- імпорт даних (із зовнішнього файлу, веб-ресурсу) – імпортуються значення курсів визначених валют за певний період.
- прогнозування певного кількості абітурієнтів на наступний проміжок часу (хвилина, година, день, неділя, місяць);
- побудова графіків прогнозу кількості абітурієнтів ;
- забезпечення зручного інтерфейсу для роботи користувача.

3) Вимоги до надійності підсистеми

Вимоги до даних: у випадку введення символьних даних у поля для чисельних даних та навпаки, ігнорувати спробу введення даних і повідомити про помилку.

4) Вимоги до складу і параметрів технічних засобів

Вимоги до програмного забезпечення:

Підсистема повинна працювати під керівництвом будь-якої операційної системи, яка підтримує JVM. Тестування системи здійснюватиметься в ОС MS Windows 7.

ПЗ підсистеми реалізується на мові Java. Для збереження даних використовується СУБД H2.

Вимоги до апаратного забезпечення (мінімальна конфігурація):

- CPU: Pentium III 1 ГГц;
- RAM: 512 Gb;
- HDD: 40Gb.

При завантаженні даних з веб-ресурсу, необхідне підключення до інтернету.

РОЗДІЛ 3 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ЩОДО ПІДСИСТЕМИ ПРОГНОЗУВАННЯ КІЛЬКОСТІ АБІТУРІЄНТІВ

3.1 Загальносистемні рішення

3.1.1 Вибір засобів моделювання

Для побудови підсистеми необхідним етапом є проектування. В останнє десятиліття в комп'ютерному світі намітилася тенденція проектування систем візуальними (наочними) моделями. Причому в нових методах проектування складних комп'ютерних систем, наприклад об'єктно-орієнтованому проектуванні (ООП) і об'єктно-орієнтованому аналізі (ООА), наочні моделі дуже часто зв'язуються з такими зоровими образами як «погляди», спрямовані на складну систему з різних точок зору. Набір з декількох наочних моделей (модельних поглядів) створює у свідомості фахівців образ складної комп'ютерної системи, яку вони спільно проектують. Разом з тим, наочні моделі є ефективним засобом документування комп'ютерних систем і програмних забезпечень, а також мовою спілкування між програмістами, системними аналітиками й замовниками систем.

На даний момент існує багато методологій моделювання систем, тому необхідно провести їх аналіз та визначити найзручніші засоби для моделювання розроблюваної підсистеми.

Найбільш відомим візуальними моделями, використовуваними для проектування комп'ютерних систем і їхніх програмних забезпечень, є діаграми мови UML (Unified Modeling Language). UML діаграми можуть бути перетворені у вихідний код (пряме перетворення) і навпаки вихідний код може бути перетворений в діаграми (зворотне перетворення). У деяких випадках пряме перетворення може здійснюватися автоматично за допомогою програм конвертерів.

Для побудови реляційних баз даних широко використовуваною є методологія IDEF1X (IDEF1 Extended). Вона відноситься до типу методологій «Сутність-зв'язок» (ER – Entity-Relationship).

Отже, для проектування підсистеми прогнозування кількості студентів доцільно буде використовувати методології UML та IDEF1X.

3.1.2 Місце підсистеми прогнозування в інформаційній системі вищого навчального закладу

Підсистема короткострокового прогнозування кількості студентів може бути впроваджена в інформаційну систему навчального закладу.

Навчальний заклад виконує наступні задачі:

- організовує навчальний процес;
- робить розрахунок витрат на учасників навчання;
- здійснює інформаційне обслуговування учасників .

Місце підсистеми прогнозування серед інших підсистем інформаційної системи навчальний заклад зображено на рисунку 3.1. Підсистема короткострокового прогнозування кількості студентів взаємодіє з підсистемою управління базою даних. Учасники використовують підсистему прогнозування для підтримки прийняття рішень по наданню якості навчання.

Інформаційна система навчальний заклад включає наступні основні підсистеми:

- підсистема, яка здійснює реєстрацію й зіставлення заявок, що надійшли: підготовка угод в автоматичному режимі за результатами їхнього зіставлення відповідно до встановлених правил навчального закладу; реєстрація підготовлених в автоматичному режимі й прямих угод; підготовка інформації про стан ринку у відповідь на запити, що надійшли.
- Підсистема електронного депозитарію, що здійснює зберігання й видачу відомостей про курси валют, проведення розрахунків по курсам валют за підсумками навчального закладу і підготовку інформації у відповідь на запити учасників навчального закладу за цими даними.
- Адміністративна підсистема, яка забезпечує визначення й зміну повноважень абонентів, керування процесом навчального закладу і настроювання підсистеми.

- Підсистема управління базою даних, яка забезпечує зберігання і пошук інформації, що використовується в ході роботи закладу.
- Підсистема прогнозування, яка використовується учасниками навчального закладу для підтримки прийняття рішень по наданню якісних освітніх послуг навчання.



Рисунок 3.1 – Інформаційна система навчальний заклад

3.2 Рішення з організаційного забезпечення

Підсистема короткострокового прогнозування кількості студентів розробляється для інтеграції до інформаційної системи навчальний заклад. Та при необхідності підсистема, що розробляється, може виступати у якості незалежної інформаційної підсистеми, що дозволяє використовувати підсистему у інших рішеннях.

Користувачами підсистеми, що розробляється будуть учасники навчального закладу. Для нормального функціонування підсистеми всі операції повинні відповідати керівництву користувача, який наведено у Додатку Б.

Програмний продукт розгортається на комп'ютері користувача, на якому необхідно встановити віртуальну машину Java. Слід зазначити, якщо підсистема входить до складу системи навчальний заклад, користувач повинен бути авторизований до роботи у системі. Ця авторизація виконується при старті системи або підключенні до системи нового користувача, тобто підсистема не виконує аутентифікацію користувачів.

Користувач або заклад, що використовуює програмний продукт на базі мови програмування Java повинен бути ознайомлений та згоджуватися з ліцензіями GNU General Public License та Java Community Process, що відповідають ліцензуванню JRE.

Користувач має можливість виконувати прогнозування відповідним методом прогнозування, додавати (імпортувати) нові часові ряди, переглядати історію прогнозування.

Адміністрування підсистеми проводиться адміністратором, що обслуговує операційну систему та ПК користувача. Адміністратор має можливість змінювати сукупність методів прогнозування та джерел імпортування, що надаються користувачеві; конфігурувати систему відповідно до операційної системи.

До розробки зовнішніх модулів прогнозування та імпортування, що не входять до початкового набору модулів залучається програміст. Програміст має право відповідно до встановленого програмного інтерфейсу розробити окремий вузол(модуль) підсистеми, та розробити відповідну документацію для впровадження окремого модуля до складу підсистеми, за котрою буде діяти адміністратор.

Слід зазначити, що усі дії адміністратора та програміста не повинні суперечити умовам ліцензування складових підсистеми.

3.3 Рішення з інформаційного забезпечення

3.3.1 Розробка концептуальної моделі БД підсистеми

Для побудови концептуальної моделі використаємо ER-діаграму. На використанні ER-моделі базується більшість сучасних підходів до проектування баз даних (головним чином, реляційних). Дана модель була запропонована Ченом в 1976 році.

Основними елементами ER-діаграм є сутності, зв'язки між ними та їх ознаки (атрибути).

Сутність – будь-який помітний серед інших об'єкт, інформацію про який необхідно зберігати в базі даних. Необхідно розрізняти такі поняття як тип сутності і екземпляр сутності. Поняття тип сутності відноситься к набору однорідних осіб, предметів, подій чи ідей, виступаючих як єдине ціле. Екземпляр сутності відноситься до конкретного предмету в наборі.

Атрибут – поименована характеристика сутності. Його найменування повинно бути унікальним для конкретного типу сутності.

Ключ – мінімальний набір атрибутів, по значенням яких можна однозначно знайти потрібний екземпляр сутності.

Одна з основних вимог до організації бази є забезпечення можливості знаходження одних сутностей по значенням інших, для чого необхідно встановлювати між ними визначені зв'язки. Зв'язок – асоціювання двох або більше сутностей. Зв'язки можуть бути декількох типів:

- один до одного (один екземпляр однієї сутності пов'язаний з одним екземпляром іншої сутності);
- один до багатьох (один екземпляр однієї сутності пов'язаний з множиною екземплярів іншої сутності);
- багато до багатьох (множина екземплярів однієї сутності пов'язана з множиною екземплярів іншої сутності).

При побудові концептуальних моделей використовують ER-діаграми. В них сутності зображуються поміченими прямокутниками, асоціації – поміченими ромбами, атрибути – поміченими еліпсами, а зв'язки між ними – ненаправленими ребрами, над якими може проставлятися степінь зв'язку (1 або літера, яка заміняє слово «багато»). Атрибути, які можуть набувати багатьох

значень називаються багатозначними і позначаються подвійною лінією. Сутності, які не можуть існувати окремо без іншої сутності називаються слабкими і позначаються подвійною лінією. Сильний зв'язок (ключ однієї сутності пов'язаний з ключем іншої сутності) також позначають подвійною лінією. Атрибут, який входить до складу ключа підкреслюється подвійною лінією.

В БД розроблюваної підсистеми сутності з їх атрибутами та відношеннями між ними зображені на рисунку 3.1.

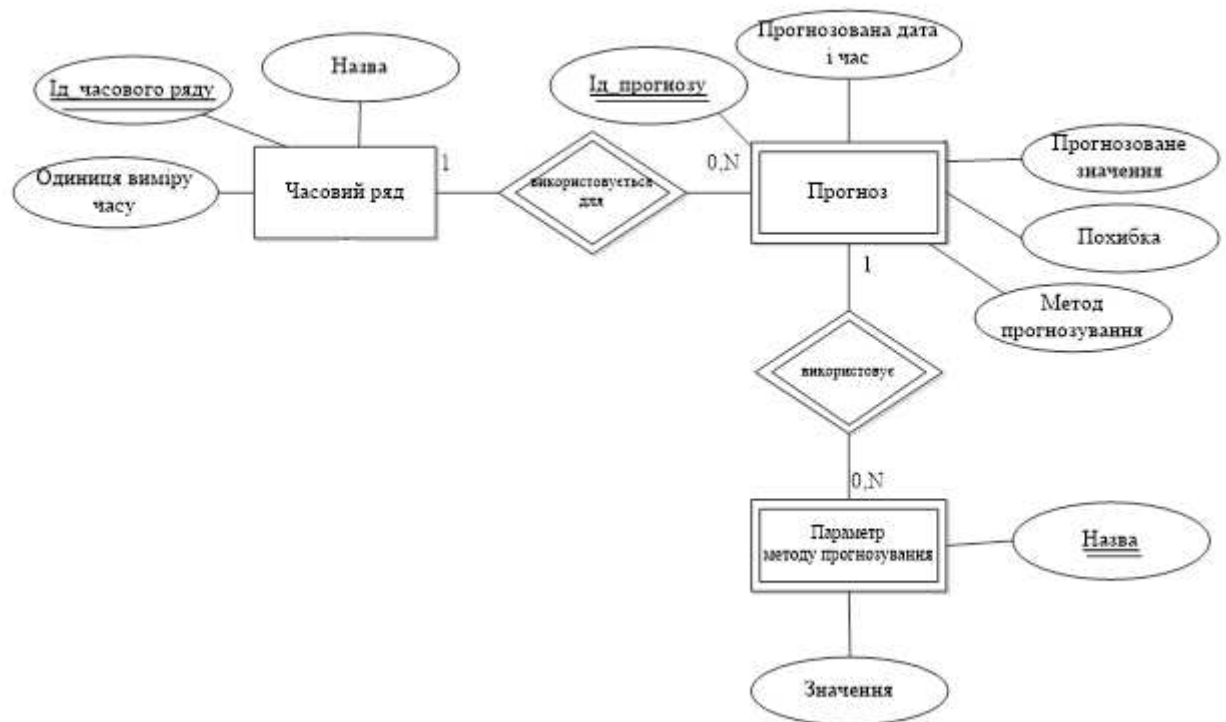


Рисунок 3.1 – Концептуальна модель БД підсистеми

3.3.2 Розробка логічної моделі БД підсистеми

Логічна модель даних є візуальним поданням структур даних, їхніх атрибутів і бізнес-правил. Логічна модель представляє дані таким чином, щоб вони легко приймалися бізнес-користувачами. Проектування логічної моделі повинно бути вільно від вимог платформи й мови реалізації або способу подальшого використання даних. Таким чином, логічна модель формується в термінах бази даних, але не залежить від конкретної СУБД.

Перехід від концептуальної моделі до логічної моделі даних має певні формальні правила, які наведені в таблиці 3.1. Таблиця має наступні скорочення: КА і КВ – ключові атрибути відповідно сутностей А і В; НА і НВ – неключові атрибути відповідно сутностей А і В; R – атрибути зв'язку.

Таблиця 3.1 – Перехід від концептуальної моделі до логічної:

Концептуальна модель	Логічна модель

Для побудови логічної моделі використаємо нотацію IDEF1X.

Згідно нотації IDEF1X таблиці зображуються прямокутником, який складається з двох частин: в верхній частині зазначаються атрибути, які входять до первинного ключа, а в нижній всі інші атрибути та зовнішні, альтернативні ключі. Напроти атрибуту, який входить до альтернативного ключа ставлять позначення АК, до зовнішнього ключа – FK. Напроти кожного атрибуту вказують його тип. Типи даних позначають наступним чином: числовий – N, текстовий – S, дата – D, відмітка часу – T, інші – O.

Сутність, унікальність якої залежить від значень атрибута, який входить до зовнішнього ключа, є залежною сутністю. Вона позначається закругленим прямокутником. При цьому зв'язок між вказаною сутністю та сутністю, на який вказує зовнішній ключ, є сильним і позначається суцільною лінією.

Логічна модель БД підсистеми в нотації IDEF1X зображена на рисунку 3.2.

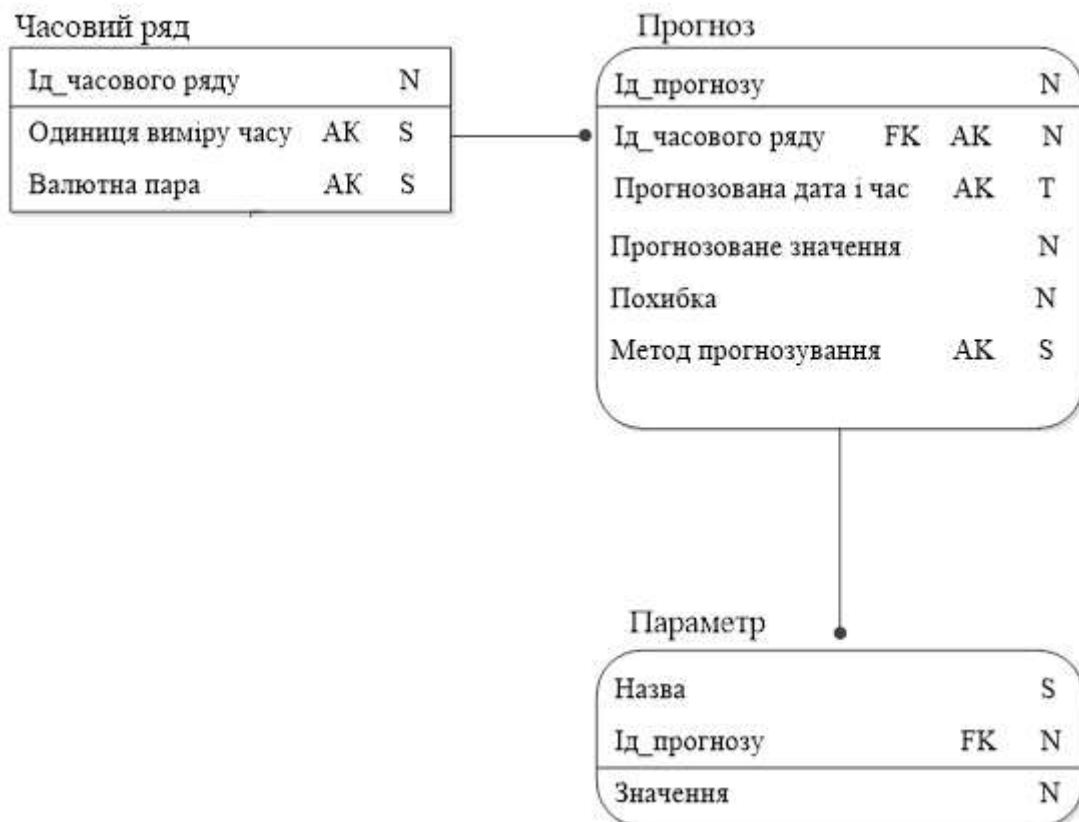


Рисунок 3.2 – Логічна модель БД підсистеми в нотації IDEF1X

Наступним етапом проектування є побудова фізичної моделі БД.

3.3.3 Розробка фізичної моделі БД підсистеми

Наступним після створення логічної моделі кроком є побудова фізичної моделі. Ця модель відповідає практичній реалізації бази даних і визначає ті фізичні об'єкти, які повинні бути реалізовані. Фізична модель даних описує структури зберігання даних з використанням всіх особливостей конкретної СУБД. Вона безпосередньо враховує такі аспекти, як архітектуру, безпеку, ефективність доступу.

Фізична модель даних будується методом нормалізації та типізації логічної моделі даних. База даних називається нормалізованою, коли кожне поле (атрибут) містить лише одне значення і кожна таблиця (сутність) містить вірний набір полів. Існує шість нормальних форм: 1НФ, 2НФ, 3НФ, БКНФ (нормальна форма Бойса-Кода), 4НФ, 5НФ. Кожна нормальна форма об'єднує в собі всі попередні нормальні форми. Тобто якщо таблиця задовольняє 3НФ, то

вона задовольняє і попередні 1НФ та 2НФ. Ціллю нормалізації є зменшення потенціального протиріччя інформації, яка зберігається в БД.

Нормальні форми можна визначити наступним чином:

- 1НФ – всі поля повинні містити тільки одне значення.
- 2НФ – будь-яке поле, яке не входить до складу потенціального ключа, залежить від всього складового потенціального ключа і не залежить від його частини.
- 3НФ – жодне з неключових полів не визначає інше неключове поле.
- БКНФ – кожен визначник є ключовим полем. Була розроблена для ситуацій, коли неключове поле визначає частину первинного ключа.
- 4НФ – в таблиці, яка містить тільки ключові поля, частина ключа може визначати множину значень тільки одного іншого поля.

Проаналізувавши логічну модель БД можна зробити висновок, що зв'язки на фізичній моделі БД будуть відповідати зв'язкам на логічній моделі. Розробимо фізичну структуру кожної таблиці, що входить до бази даних.

Для розробки БД підсистеми короткострокового прогнозування курсів валют будемо використовувати СУБД Н2. Н2 є відкритою кросплатформенною СУБД, яка написана на мові Java і характеризується легкою інтеграцією з Java-програмами.

Фізична структура таблиць, які входять до БД підсистеми наведена в таблицях 3.2 – 3.5.

Таблиця 3.2 – Структура таблиці «Часовий ряд»

Назва поля	Тип даних	Ключі
idTimeSeries	INTEGER	PK
timeUnit	VARCHAR(15)	AK
currencyPair	VARCHAR(6)	AK

Таблиця 3.3 – Структура таблиці «Прогноз»

Назва поля	Тип даних	Ключі
idPrognose	INTEGER	PK

idTimeSeries	INTEGER	AK, FK
dateTimeProg	TIMESTAMP	AK
typePriceProg	VARCHAR(4)	AK
valueProg	NUMERIC(10,7)	
error	NUMERIC(10,7)	
method	VARCHAR(20)	AK

Таблиця 3.4 – Структура таблиці «Параметр»

Назва поля	Тип даних	Ключі
name	VARCHAR(15)	PK
idPrognose	INTEGER	PK, FK
value	DOUBLE	

Проаналізувавши задані таблиці, можна зробити, висновок, що вони знаходяться в нормальній формі Бойса-Кодда, тобто вони задовольняють 1НФ, 2НФ, 3НФ і будь-яка функціональна залежність між їх полями зводиться до повної функціональної залежності від можливого ключа.

Розроблена база даних є локального масштабу. Для зберігання даних використовується СУБД H2, яка може працювати у вбудованому режимі (embedded mode). У вбудованому режимі, додаток відкриває базу даних в межах однієї JVM з використанням JDBC. Це найшвидший та найпростіший спосіб підключення.

3.4 Рішення з технічного забезпечення

Так як дана підсистема є достатньо простою з точки зору реалізації збереження даних, то вона може бути розгорнута на одному комп'ютері, тому розробка мережевої інфраструктури не є необхідною.

За зведеними даними характеристик обраного прогнозного забезпечення і пропозицій на ринку комп'ютерної техніки визначимо конфігурацію технічних засобів.

Вимоги до технічного забезпечення клієнтських машин (мінімальна конфігурація):

- Процесор – Pentium IV 1,6 ГГц;
- Оперативна пам'ять – 512 Мб;
- Жорсткий диск – 40 Гб.

Для імпорту даних з веб-ресурсу необхідне підключення до Internet.

3.5 Рішення з математичного забезпечення

Для короткострокового прогнозування кількості студентів використовується модель клітинних автоматів.

Алгоритм прогнозування на базі клітинного автомату складається із п'яти етапів:

Етап 5. *Передпрогнозний аналіз даного часового ряду(ЧР)*. На даному етапі вихідний ЧР досліджується на наявність пам'яті та здійснюється оцінка її глибини L . Інструментарій передпрогнозного аналізу – метод послідовного R/S -аналізу, який описаний в підрозділі 1.3.4.

Перетворення числового ЧР в лінгвістичний ЧР (ЛЧР). Перетворення вихідного ЧР $Z = \{z_1, z_2, \dots, z_n\}$ в ЛЧР $U = \{u_1, u_2, \dots, u_n\}$ означає заміну числових елементів лінгвістичними, які називаються термами; сукупність термів прийнято називати терм-множиною, яку позначають $T = \{u^1, u^2, \dots, u^k\}$. Множина T складається із k елементів. Для здійснення перетворення вихідний ряд ділиться на декілька частин або вікон. Для кожного вікна визначаються максимальне і мінімальне значення елементів часового ряду, які входять до нього. Різниця між ними визначає ширину вікна. Знайдена ширина вікна ділиться на визначену кількість інтервалів, яка дорівнює кількості термів в терм-множині. Кожному інтервалу ставиться у відповідність певний терм. Далі кожному елементу вихідного часового ряду Z ставиться у відповідність певне значення терму, в залежності від того, в який із інтервалів він потрапив, в результаті чого отримується ЛЧР $U = \{u_{m+1}, u_{m+2}, \dots, u_n\}$. Мінімальна довжина вікна m зв'язана з кількістю термів k наступною залежністю:

$$m \approx 3 \cdot k. \quad (1.34)$$

Етап 6. Побудова клітинного автомату для отриманого ЛЧР. Наявність в ЛЧР пам'яті пропонується представляти в термінах і поняттях клітинного автомата, зокрема лінійного клітинного автомата. Терміном « l -конфігурація» називають всяку послідовність $g = u_{i+1}u_{i+2}\dots u_{i+l}$, $i = 0, 1, 2, \dots, n-l$, $l=1, 2, \dots, L$, де L – глибина пам'яті ряду який розглядається. Розглянемо яку-небудь фіксовану l -конфігурацію, яку позначимо $g^0 = u_1^0 u_2^0 \dots u_j^0 \dots u_l^0$. Якщо в ЛЧР виділена послідовність $u_{i+1}u_{i+2}\dots u_{i+j}\dots u_{i+l}$, $i = 0, 1, 2, \dots, n-l$, яка співпадає з g^0 , тобто $u_{i+j} = u_j^0$, $j = 1, 2, \dots, l$, то по відношенню до наступного елементу $u_{i+l+1} = u^0$, де u^0 належить до терм-множини T , говорять, що l - конфігурація переходить у стан u^0 , тобто в лінгвістичну змінну u_{i+l+1} , яка співпадає з термом u^0 .

При побудові клітинного автомату для даного ЛЧР спочатку формують множину M всіх l - конфігурацій, які містяться в ЛЧР, а потім обчислюються частоти переходів l - конфігурацій M в терми з T , тобто будується пам'ять клітинного автомату.

Частотна статистика переходів l - конфігурацій в певний стан u^r із терм-множини T формується наступним чином. Для кожної l - конфігурації g лінгвістичного часового ряду U підраховуємо кількість її переходів в кожен із станів терм-множини T , після чого сумуємо їх. Частота переходу l - конфігурації g в певний стан u^r обчислюється шляхом ділення кількості переходів даної конфігурації g в даний стан u^r на загальну кількість переходів конфігурації g :

$$w(g, u^r) = W(g, u^r) / \sum_{r=1}^k W(g, u^r).$$

Етап 7. Прогнозування на базі клітинного автомату. Прогноз терму u_{n+1} представляється у вигляді нечіткої терм-множини (НТМ): $U_{n+1} = \{(u^1; \mu_1), (u^2; \mu_2), \dots, (u^k; \mu_k)\}$, де значення функції належності задовольняють рівності $\mu_1 + \mu_2 + \dots + \mu_k = 1$. Значення $\mu_1, \mu_2, \dots, \mu_k$ обчислюються через значення частот переходів, які отримуються для різних l - конфігурацій з послідовності $u_{n-L+1}u_{n-L+2}\dots u_{n-1}u_n$, де L – глибина пам'яті ряду. Спочатку обчислюються ненормовані значення:

$$\mu'_1 = w(u_n, u^1) + w(u_{n-1}u_n, u^1) + \dots + w(u_{n-L+1} \dots u_{n-1}u_n, u^1),$$

$$\mu'_2 = w(u_n, u^2) + w(u_{n-1}u_n, u^2) + \dots + w(u_{n-L+1} \dots u_{n-1}u_n, u^2),$$

$$\mu'_k = w(u_n, u^k) + w(u_{n-1} u_n, u^k) + \dots + w(u_{n-L+1} \dots u_{n-1} u_n, u^k),$$

і їх сума: $s = \mu'_1 + \mu'_2 + \dots + \mu'_k$. Після нормування обчислених значень отримуємо:

$$\mu_1 = \frac{\mu'_1}{s}, \mu_2 = \frac{\mu'_2}{s}, \dots, \mu_k = \frac{\mu'_k}{s}.$$

Так як ЛЧР U отриманий в результаті застосування алгоритма класифікації до деякого числового ЧР Z , НТМ можна перетворити в нечітку (числову) множину $Z_{n+1} = \{(z^1; \mu_1), (z^2; \mu_2), \dots, (z^k; \mu_k)\}$. Числові значення елементів z^1, z^2, z^k отримуються наступним чином:

$$z^i = (z_{max} - z_{min}) \cdot \frac{i-1}{k} + z_{min} + \frac{z_{max} - z_{min}}{2},$$

де z_{max} і z_{min} — відповідно максимальне і мінімальне значення ЧР в останньому ковзному вікні. Нечіткий числовий прогноз Z_{n+1} переводиться в чітке прогнозне значення z_{n+1} шляхом застосування до Z_{n+1} операції дефазифікації:

$$z_{n+1} = z^1 \cdot \mu_1 + z^2 \cdot \mu_2 + \dots + z^k \cdot \mu_k.$$

Етап 8. *Оцінка точності прогнозу*. Рівень точності прогнозу обчислюється наступним чином: починаючи з довжини конфігурації l послідовно прогнозуються $(l+1)$ -й, $(l+2)$ -й, і т.д. відомий елемент часового ряду. Потім, використовуючи формули наведені в підрозділі 1.4.1, розраховуємо оцінку точності прогнозу.

На рисунку 3.3 зображений алгоритм здійснення прогнозування.

На рисунку наступні позначення:

- n — довжина часового ряду на основі якого робиться прогноз;
- p — максимальна довжина конфігурацій автомата;
- m — довжина ковзного вікна;
- k — кількість термів;
- k_{opt} — оптимальна кількість термів;
- m_{opt} — оптимальна довжина ковзного вікна;
- e_{min} — мінімальна помилка прогнозування;
- X_{prog} — Прогнозне значення;

- X_{fact} – реальне значення.

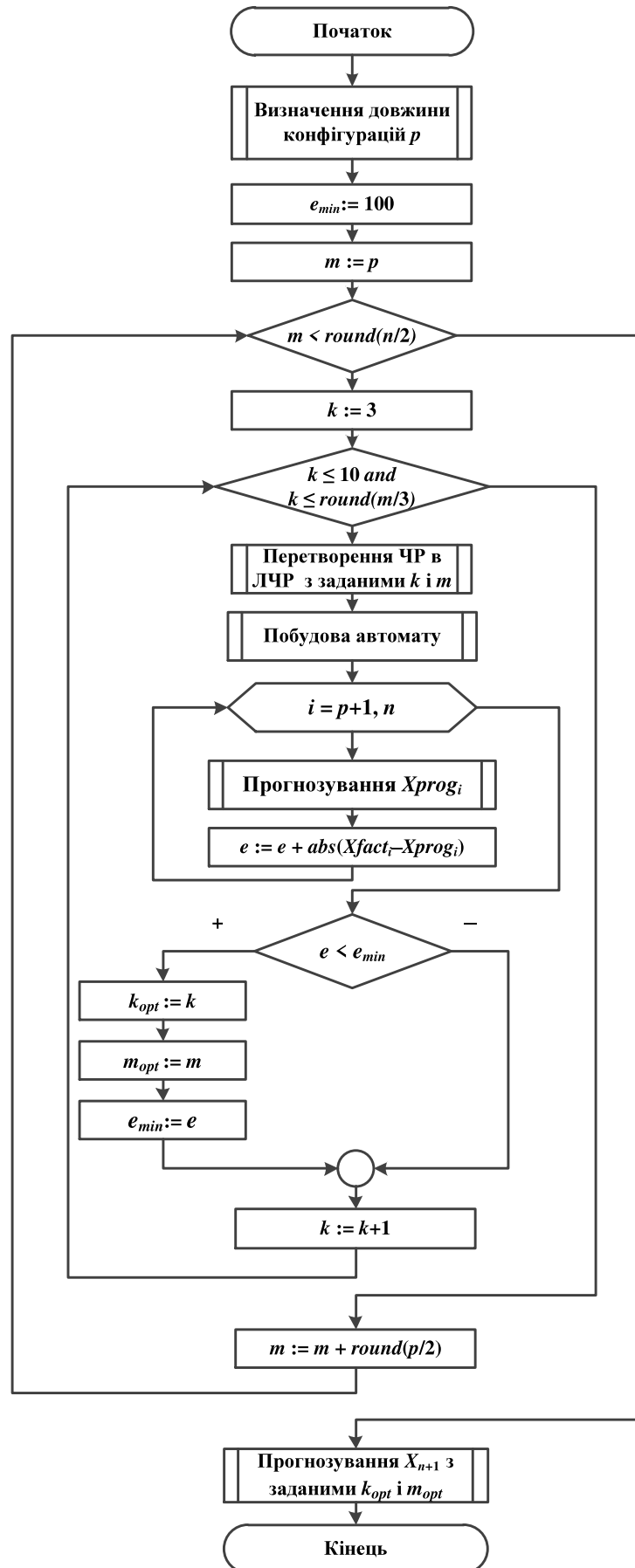


Рисунок 3.3 – Алгоритм прогнозування

3.6 Рішення з програмного забезпечення

3.6.1. Побудова діаграми варіантів використання

Для виявлення основних функцій, що входять до складу підсистеми і вивчення динаміки її функціонування в цілому необхідно побудувати функціональну модель. Вона демонструє, яким чином вхідні дані перетворюються у вихідні дані, не розглядаючи порядок або спосіб реалізації перетворень. Ця модель складається з набору діаграм прецедентів, які показують потоки значень від зовнішніх входів через операції і внутрішні сховища даних до зовнішніх виходів.

Традиційно проектування програмного забезпечення починається з представлення у зручному вигляді інформації – з якими зовнішніми сутностями (акторами) і даними буде обмінюватись система. Для цього зручно використовувати діаграму прецедентів Use Case UML. Цей засіб використовується для опису документообігу та обробки інформації. Він є основним засобом моделювання функціональних вимог системи при проектуванні. З їх допомогою вони розбиваються на функціональні компоненти (процеси) і представляються у вигляді мережі, зв'язаної потоками даних. Головна мета таких засобів – продемонструвати, як кожен процес перетворить свої вхідні дані у вихідні, а також, виявити відносини між цими процесами.

Джерела інформації (актори) породжують інформаційні потоки (потоки даних), що переносять інформацію до підсистем чи до процесів. Ті у свою чергу перетворюють інформацію і породжують нові потоки, що переносять інформацію до інших процесів чи підсистем, накопичувачів даних або зовнішніх сутностей – споживачів інформації.

Таким чином, основними компонентами діаграм потоків даних є:

- актори;
- прецеденти;
- накопичувачі даних
- потоки даних.

Зовнішня сутність (актор) являє собою матеріальний предмет чи фізичну особу, що представляє собою джерело чи приймач інформації. Для розроблюваної підсистеми прогнозування кількості студентів, зовнішньою сутністю є користувач (трейдер).

Прецеденти – це вимоги, в основному це функціональні вимоги, які вказують на те, що має робити підсистема. За допомогою прецедентів моделюється діалог між актором і системою. Набір усіх прецедентів підсистеми визначає способи її використання.

Виділимо наступні прецеденти:

- імпорт даних про освітній процес;
- формування прогнозу;
- передпрогнозний аналіз даних ;
- перетворення ЧР в ЛЧР;
- побудова клітинного автомату;
- отримання історії прогнозів;
- побудова графіків прогнозу та похибки.

Для розроблюваної підсистеми короткострокового прогнозування кількості студентів основні процеси на Use Case пов'язані з імпортом даних про освітній процес та прогнозуванням кількості студентів .

Накопичувач даних являє собою абстрактний пристрій для збереження інформації, яку можна в будь-який момент помістити в накопичувач і через якийсь час знову взяти, причому способи запису і зчитування можуть бути будь-якими.

Важливу специфічну роль у моделі відіграє спеціальний вид діаграми потоків – діаграма варіантів використання. Діаграма варіантів використання – це погляд на систему, що не залежить від її реалізації. Вона відображає, що система робить, а не як вона це робить. Діаграма показує взаємодію об'єктів, а не класів. У проєктованій підсистемі є два об'єкта – користувач, який взаємодіє з підсистемою, та база даних підсистеми, де зберігаються дані для прогнозування. У випадку інтегрування підсистеми до

системи, вона ще буде взаємодіяти з БД системи. Діаграма варіантів використання для підсистеми короткострокового прогнозування кількості студентів зображена на рисунку 3.4.

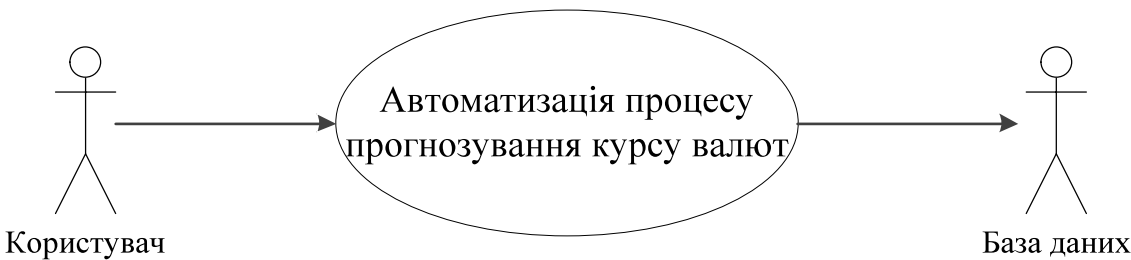


Рисунок 3.4 – Діаграма варіантів використання нульового рівня

Декомпозиція процесів діаграми варіантів використання нульового рівня наведена у таблиці 3.6.

Таблиця 3.6 – Декомпозиція процесів для діаграми використання

Процес нульового рівня	Процес першого рівня
Автоматизація процесу прогнозування кількості студентів	Імпорт даних про освітній процес. Формування прогнозу. Отримання історії прогнозів. Побудова графіків прогнозу та похибки.

Для продовження аналізу функціональної моделі програмного забезпечення будуються діаграми наступних рівнів. Use Case першого рівня будується як декомпозиція процесів, що присутні на діаграмі варіантів використання нульового рівня. Use Case першого рівня представлення на рисунку 3.5.

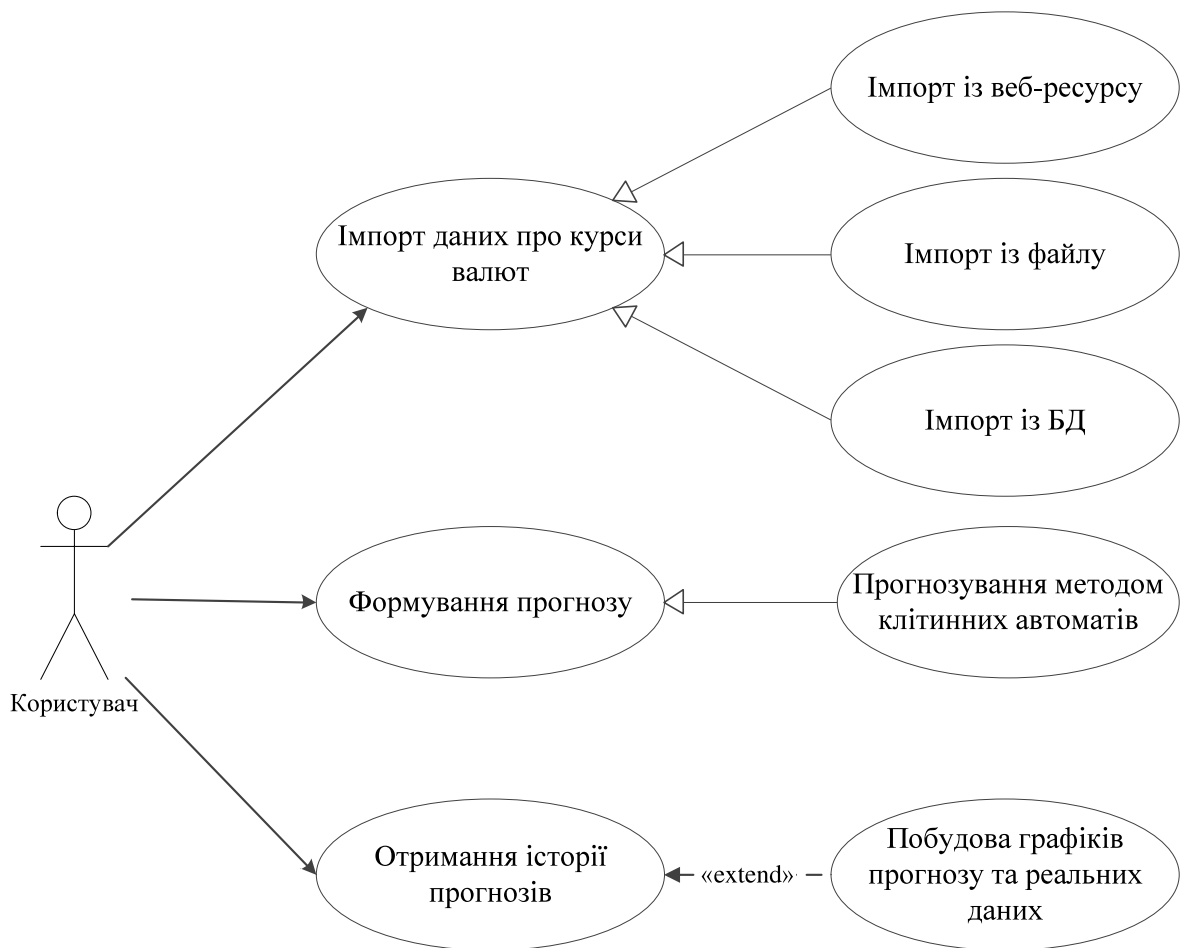


Рисунок 3.5 – Діаграма варіантів використання першого рівня

Головними варіантами використання підсистеми є «Імпорт даних про освітній процес», «Формування прогнозу» та «Отримання історії прогнозів». Спеціальними випадками варіанту використання «Імпорт даних про освітній процес» є такі варіанти використання як «Імпорт із веб-ресурсу», «Імпорт із файлу». Варіант використання «Прогнозування методом клітинних автоматів» є спеціальним випадком варіанту використання «Формувати прогноз». Варіант використання «Отримання історії прогнозів» може бути розширений побудовою графіку прогнозів.

Словник даних

У таблиці 3.7 наведено опис потоків даних декомпозиції діаграми варіантів, зображеної на рисунку 3.5.

Таблиця 3.7 – Декомпозиція потоків даних для діаграми Use Case

Потоки даних на діаграмі нульового рівня	Потоки даних на діаграмі першого рівня	Пояснення
Інформація від користувача	Запит на імпорт даних про освітній процес	Назва валюти або назва зовнішнього файлу, який містить статистику курсів валюти, вибір джерела імпорту.
	Запит на формування прогнозу	Запит користувача на формування прогнозу певної валюти.
	Запит на перегляд історії прогнозів	Запит користувача на перегляд історії прогнозів (прогнозу певної валюти за певний період).
	Запит на вивід «Довідки»	Запит користувача на вивід «Довідки» на екран монітору.
Інформація для користувача	Історія прогнозів курсу обраної валюти	Вивід на екран монітору історії прогнозів курсу обраної валюти з БД.
	Прогнозні дані розглянутої валюти	Вивід на екран монітору прогнозних даних обраної валюти
	Графік прогнозів	Вивід на екран монітору графіку прогнозів певної валюти

Декомпозиція варіанту використання «Прогнозування методом клітинних автоматів» зображена на рисунку 3.6.

Варіант використання «Прогнозування Т-модель» дозволяє побудувати прогноз певної валюти на визначений період на основі моделі клітинних автоматів. Для побудови прогнозу необхідно надати інформацію щодо здійснення прогнозу (розмір часового ряду для аналізу, метод прогнозування, дату, яку необхідно спрогнозувати). В результаті користувач отримує результати прогнозування (прогноз і його помилку).

Варіант використання «Прогнозування Т-модель» включає побудову клітинного автомату і власне здійснення прогнозу. Побудова клітинного автомату включає передпрогнозний аналіз даних та перетворення ЧР в ЛЧР.

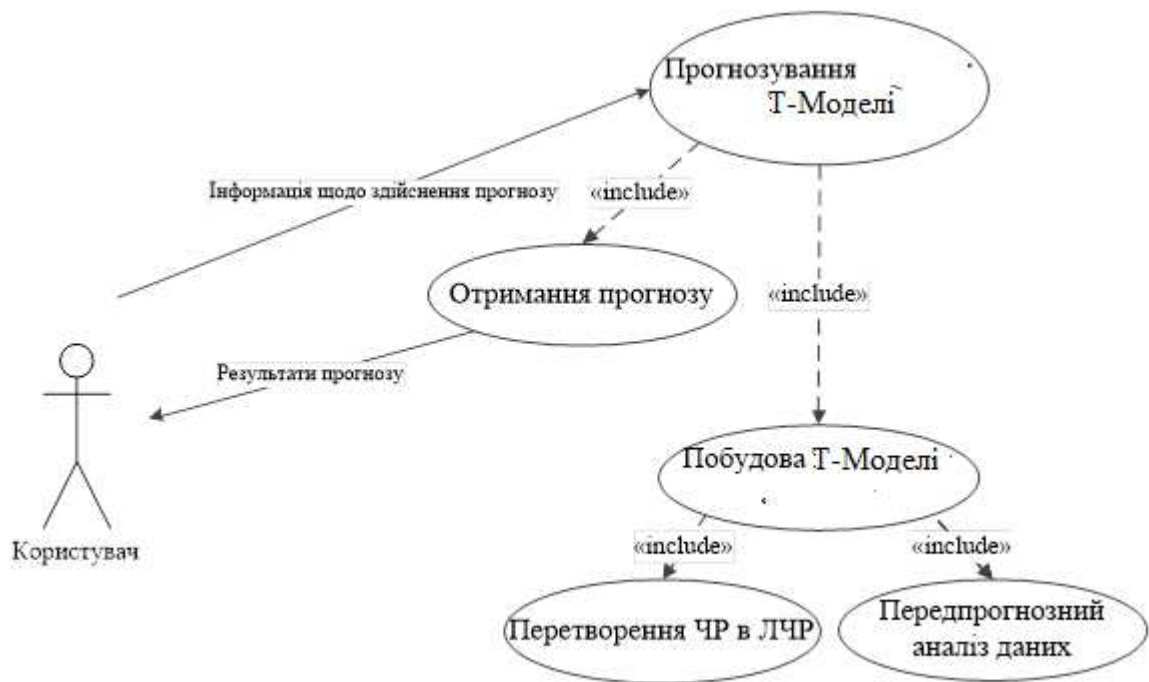


Рисунок 3.6 – Декомпозиція варіанту використання «Прогнозування Т-модель»

3.6.2 Опис функцій, що автоматизуються

а) Мета створення підсистеми короткострокового прогнозування курсів валют та функції, що автоматизуються.

Мета розробки – автоматизації процесу прогнозування, а саме побудова прогнозу на базі моделі клітинного автомату, з метою скорочення часу обробки даних й одержання найбільш якісного прогнозу (мінімізація помилки прогнозування).

Функції, що автоматизуються:

- імпорт даних (із зовнішнього файлу, із веб-ресурсу або БД зовнішньої системи) – імпортуються значення курсів визначених валют за певний період.
- формування вибірки даних;
- передпрогнозний аналіз вхідних даних – R/S-аналіз, оцінка глибини пам'яті часового ряду;
- перетворення числового часового ряду в лінгвістичний часовий ряд;

- побудова клітинного автомату;
- прогнозування на базі клітинного автомату;
- побудова графіку прогнозів кількості студентів ;
- ведення історії прогнозів курсів валют.

б) Вхідні дані:

- назва валюти;
- одиниця виміру часу;
- історичні дані про освітній процес;
- дата, на яку потрібно зробити прогноз;
метод прогнозування;
- розмір часового ряду для аналізу;

в) Вихідні дані:

- історія прогнозів курсів валют;
- прогнозовані дані розглянутої валюти;
- графіки прогнозів.

3.6.3 Побудова діаграми класів

Клас (class) в мові UML служить для позначення безлічі об'єктів, які володіють одноковою структурою, поведінкою і відносинами з об'єктами інших класів. Діаграма класів служить для представлення статичної моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класів може відображати, зокрема, різні взаємозв'язки між окремою суттю наочної області, такими як об'єкти під системи, а також описує їх внутрішню структуру і типи відносин. Діаграма класів, які описують типи даних в підсистемі наведена на рисунку 3.7.

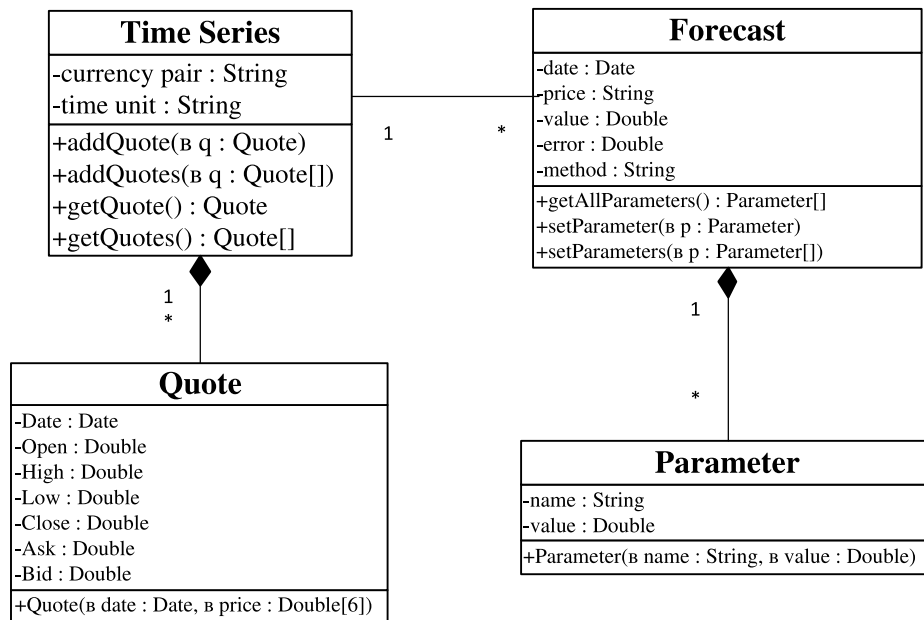


Рисунок 3.7 – Діаграма класів типів даних в підсистемі

Розглянемо наведені класи.

Клас **TimeSeries** (Часовий ряд) – містить інформацію про котирування валютної пари (назва валютної пари, одиниця виміру часу, масив котирувань), повертає масив котирувань.

Клас **Quote** (Котирування) – складова частина класу **TimeSeries**, містить наступну інформацію:

- **Date** – дата;
- **Open** (ціна відкриття) – ціна першої укладеної угоди протягом певного часового періоду;
- **Max** і **min** (максимальна ціна) – показують відповідно найбільшу і найменшу ціну угоди протягом певного часового періоду.
- **Close** (ціна закриття) – ціна останньої угоди протягом торгового дня. Звичайно вона вважається найбільш інформативною, так як включає всю біржову інформацію по підсумкам певного часового періоду. На її основі звичайно будуються графіки.
- **Bid** – ціна пропозиції, найкраще котирування на купівлю.
- **Ask** – ціна попиту, найкраще котирування на продаж.

Клас **Forecast** (Прогноз) – містить дані про прогноз (прогнозовану дату, прогнозовану ціну, прогнозоване значення, похибку, назва методу яким

здійснювався прогноз та масив параметрів, які використовувались для здійснення прогнозу).

Клас Parameter (Параметр) – є складовою частиною класу Forecast, містить назву параметру та значення.

Діаграма класів для всієї підсистеми наведена на рисунку 3.8.

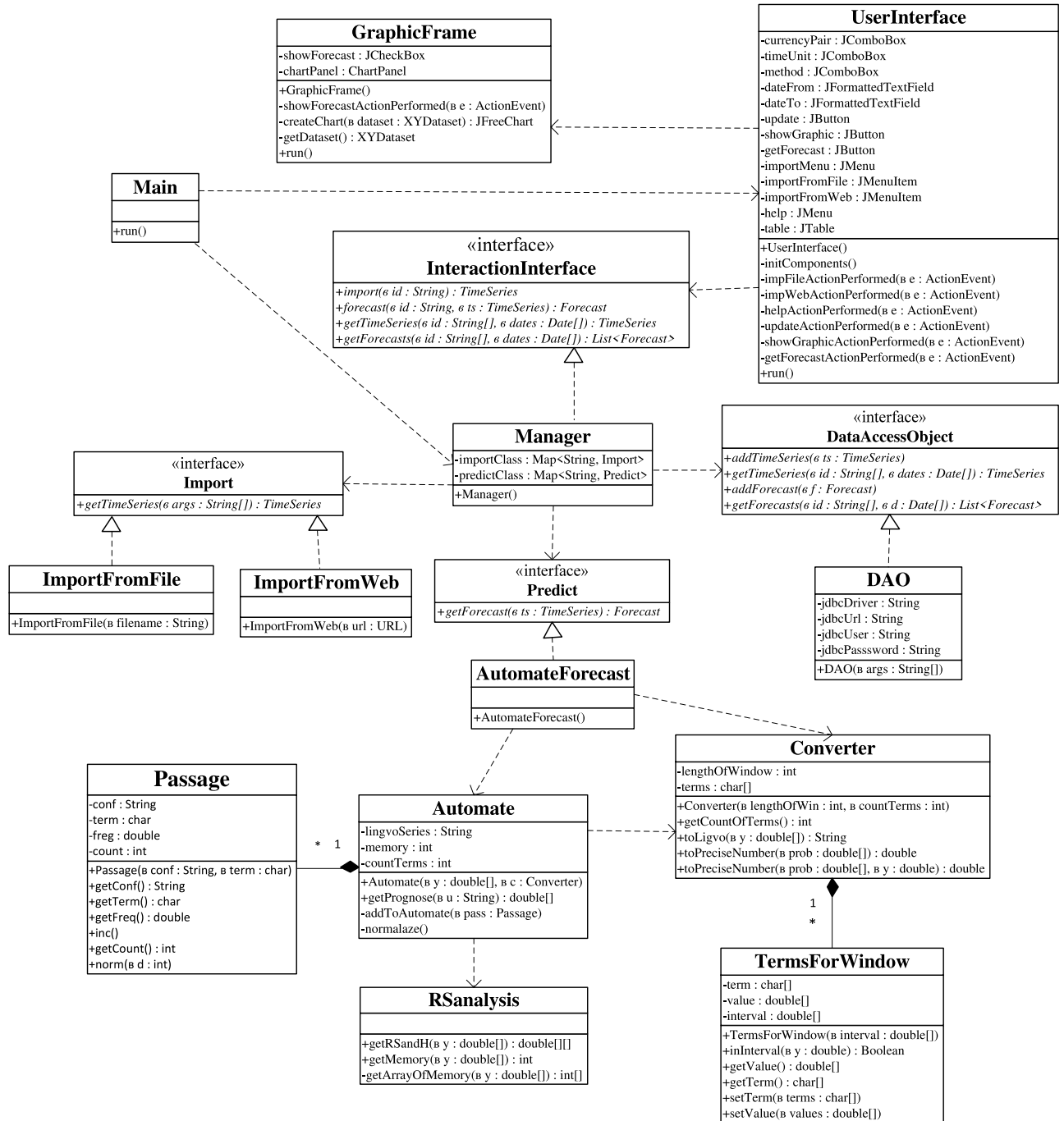


Рисунок 3.7 – Діаграма класів ПЗ підсистеми

Розглянемо кожен з приведених на діаграмі класів.

Клас Main – головний клас, який містить точку входу в програму (функцію main). Він створює об'єкти класів Manager та UserInterface.

Клас Manager – організує взаємодію між класами, які відповідають за доступ до БД, імпорт даних в підсистему та здійснення прогнозування. Також цей клас реалізує інтерфейс InteractionInterface, за допомогою якого здійснюється взаємодія з користувачем підсистеми. Функції класу Manager:

- зчитування конфігураційного файлу та загрузка необхідних модулів;
- користуючись послугами інших класів здійснювати імпорт даних в програму, збереження та видачу необхідних даних з бази даних програми, прогноз певного часового ряду.

Клас UserInterface – відображає інтерфейс користувача. Функція цього компоненту – здійснювати взаємодію з користувачем. Для цього він використовує інтерфейс Interaction. Цей клас також здійснює побудову графіків, для чого використовує клас GraphicFrame.

Клас DAO реалізовує інтерфейс DataAccessObject, за допомогою якого здійснюється доступ до даних підсистеми. Його основна функція – зберігати та видавати необхідні дані із БД. Для чого він здійснює наступні функції:

- зберегти часовий ряд в БД;
- отримати необхідний часовий ряд з БД;
- зберегти прогноз в БД;
- отримати прогнози з БД.

Клас AutomateForecast реалізовує інтерфейс Predict через який в підсистемі здійснюється прогнозування. Функція класу AutomateForecast – здійснювати прогноз. В якості вхідних даних він отримує об'єкт TimeSeries, який містить дані про часовий ряд, а повертає об'єкт Forecast, який містить дані про прогноз. Для здійснення прогнозу він використовує класи Automate і Converter. Клас Automate відповідає за побудову автомату та здійснення прогнозу у вигляді нечіткої терм-множини. Для зберігання частот переходів із конфігурацій в терми клас Automate містить об'єкти класу Passage. Клас

Converter відповідає за перетворення числового часового ряду в лінгвістичний часовий ряд та за перетворення прогнозу у вигляді нечіткої терм-множини в числовий вид. Для здійснення перетворення прогнозу в числовий вид клас Converter зберігає для кожного інтервалу, на якому здійснювалося перетворення значення термів, які в нього входять, та їх числове представлення. Для цього використовується клас TermsForWindow.

Клас ImportFromWeb здійснює імпорт даних з веб-ресурсу і повертає їх у вигляді об'єкту TimeSeries. В якості вхідних даних отримує дані про часовий ряд, який необхідно загрузити.

Клас ImportFromFile здійснює імпорт даних з файлу і повертає їх у вигляді об'єкту TimeSeries. В якості вхідних даних отримує назву файлу та дані про часовий ряд, який необхідно загрузити.

Класи ImportFromWeb та ImportFromFile реалізують інтерфейс Import.

Створення інтерфейсів Import та Predict дозволяє забезпечити розширюваність програми. Сторонній розробник може написати свій клас, який буде реалізовувати вказані інтерфейси та підключити його до програми. Таким чином, можлива реалізація у майбутньому імпорту даних з якоїсь зовнішньої бази даних, додавання нового методу для прогнозування.

3.6.4 Побудова діаграми послідовності

Однією з характерних особливостей систем різної природи і призначення є взаємодія між собою окремих елементів, з яких утворені ці системи. Йдеться про те, що різні складові елементи систем не існують ізольовано, а роблять певний вплив один на одного, що і відрізняє систему як цілісне утворення від простої сукупності елементів.

У мові UML взаємодія елементів розглядається в інформаційному аспекті їх комунікації, тобто взаємодіючі об'єкти обмінюються між собою деякою інформацією. При цьому інформація приймає форму закінчених повідомлень. Іншими словами, хоча повідомлення і має інформаційний зміст, воно має додаткову властивість робити направлений вплив на свого одержувача. Це

повністю узгоджується з принципами ООП, коли будь-які види інформаційної взаємодії між елементами системи повинні бути зведені до відправки і прийому повідомлень між ними.

Взаємодії об'єктів можна розглядати в часі, і тоді для представлення тимчасових особливостей передачі і прийому повідомлень між об'єктами використовується діаграма послідовності.

На діаграмі послідовності зображуються виключно ті об'єкти, які безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами. Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об'єктів в часі. При цьому діаграма послідовності має як би два вимірювання. Одне — зліва направо у вигляді вертикальних ліній, кожна з яких зображає лінію життя окремого об'єкту, що бере участь у взаємодії. Графічно кожен об'єкт зображується прямокутником і розташовується у верхній частині своєї лінії життя.

Друге вимірювання діаграми послідовності — вертикальна тимчасова вісь, направлена зверху вниз. Початковому моменту часу відповідає сама верхня частина діаграми. При цьому взаємодії об'єктів реалізуються за допомогою повідомлень, які посилаються одними об'єктами іншим. Повідомлення зображуються у вигляді горизонтальних стрілок з ім'ям повідомлення і також утворюють порядок за часом свого виникнення. Іншими словами, повідомлення, розташовані на діаграмі послідовності вище, ініціюються раніше тих, які розташовані нижче. При цьому масштаб на осі часу не вказується, оскільки діаграма послідовності моделює лише тимчасову впорядкованість взаємодій типу "раніше-пізніше".

Діаграма послідовності підсистеми приведена нижче.

При запуску програми створюється головний об'єкт `main`. Він створює об'єкти менеджер (`manage`) та інтерфейс користувача (`interface`). При створенні об'єкту `manage` зчитуються конфігураційні дані та створюються інші об'єкти: `imp1`, який імпортує дані з веб-ресурсу, `imp2`, який імпортує дані з файлу, `forecast`, який здійснює прогнози, `dataAccess`, який взаємодіє з БД.

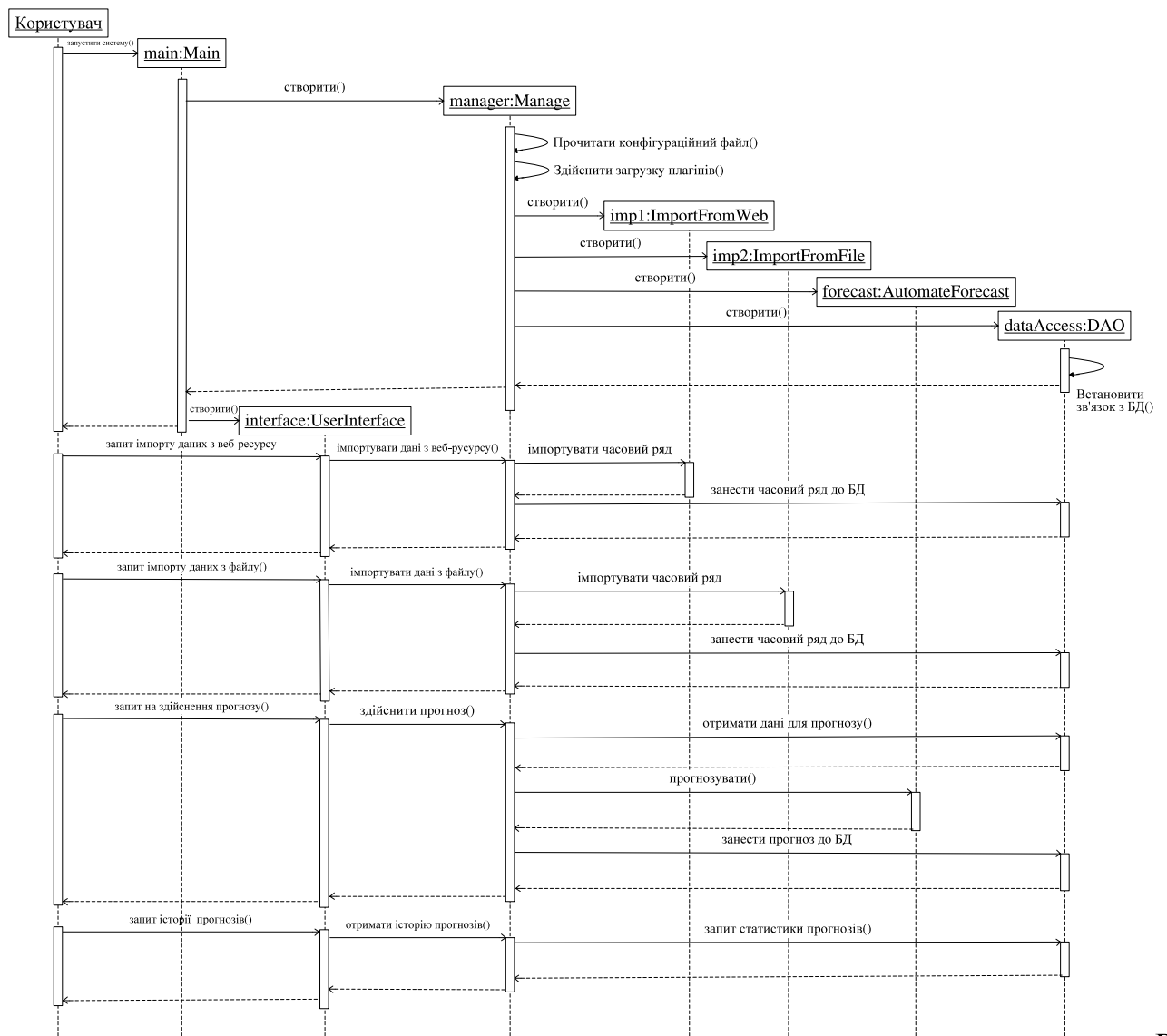
Далі зображуються обміни повідомленнями між об'єктами при виконанні запитів, які надходять від користувача:

- запит імпорту даних з інтернету;
- запит імпорту даних з файлу;
- запит на здійснення прогнозу;
- запит на отримання історії прогнозів.

При запиті користувача активується об'єкт `interface`, який для виконання запиту користувача користується послугами об'єкта `manage`.

Якщо виконується запит на імпорт даних, то об'єкт `manage` звертається до об'єкта `imp1` або `imp2` (в залежності від вказаного джерела імпорту), який власне і здійснює імпорт даних. Далі, отримавши від об'єкта `imp1` або `imp2` визначений часовий ряд, об'єкт `manage` звертається до об'єкту `dataAccess`, щоб зберегти тільки що отримані дані в базі даних.

Якщо виконується запит на здійснення прогнозу, то об'єкт `manage` звертається до об'єкту `dataAccess`, щоб отримати вказаний часовий ряд для здійснення прогнозу, потім він визиває метод прогнозування об'єкту `forecast`, отримує від нього прогноз і знову звертається до об'єкту `dataAccess`, щоб зберегти отриманий прогноз.



Р

исунок 3.8 – Діаграма послідовності ПЗ підсистеми

3.6.5 Методи і засоби розробки ПЗ

Так як однією з вимог до розроблюваної підсистеми є кросплатформеність, то в якості мови програмування доцільно вибрати мову Java. Java програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи. Основна перевага використання байт-коду це портативність. Тим не менш, додаткові витрати на інтерпретацію означають, що інтерпретовані програми будуть майже завжди працювати повільніше, ніж скомпільовані у машинний код, і саме тому Java одержала репутацію «повільної» мови. Проте, цей розрив суттєво

скоротився після введення декількох методів оптимізації у сучасних реалізаціях JVM. Основні можливості Java:

- автоматичне управління пам'яттю;
- розширені можливості обробки виключних ситуацій;
- багатий набір засобів фільтрації вводу/виводу;
- набір стандартних колекцій, таких як масив, список, стек і т.п.;
- присутність класів, які дозволяють виконувати HTTP-запити і оброблювати відповіді;
- вбудовані в мову засоби створення багато поточних додатків;
- уніфікований доступ до баз даних:
 - на рівні окремих SQL-запитів – на основі JDBC, SQLJ;
 - на рівні концепції об'єктів, які мають здатність до збереження в базі даних – на основі Java Data Objects (англ.) і Java Persistence API (англ.);
- підтримка шаблонів;
- паралельне виконання програм;

В якості середовища розробки оберемо вільне інтегроване середовище розробки додатків NetBeans IDE. За якістю та можливостями останні версії NetBeans IDE не уступають найкращим комерційним (платним) інтегрованим середовищам розробки мови Java, таким як, наприклад, IntelliJ IDEA.

Для розробки користувацького інтерфейсу будемо використовувати бібліотеку Swing, яка є частиною бібліотеки базових класів Java. Для побудови графіків будемо використовувати бібліотеку JFreeChart.

Для побудови та управління базою даних будемо використовувати СУБД H2, яка може працювати у вбудованому режимі (embedded mode) та легко вбудовується в Java додатки.

Програмне забезпечення буде розроблюватися та тестуватися в операційній системі Window 7, проте розроблювана підсистема є

кросплатформенною, її можна розгорнути на всіх платформах, які підтримують JVM.

3.4.6 Побудова діаграми компонентів

Для зображення структури програмного забезпечення використаємо таку діаграму UML як діаграма компонент. Діаграма компонент – діаграма, на якій відображаються компоненти, залежності та зв'язки між ними. Вона описує особливості фізичного представлення системи. Діаграма компонентів дозволяє визначити архітектуру системи, що розробляється, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний, бінарний і виконуваний код.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компоновки, а деякі під час роботи програми. У багатьох середовищах розробки модуль чи компонент відповідає файлу. Пунктирні стрілки, що з'єднують модулі, показують відношення взаємозалежності, аналогічні тим, які мають місце при компіляції вихідних текстів програм.

Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними. Компонент реалізує деякий набір інтерфейсів і служить для загального позначення елементів фізичного представлення моделі. Для графічного подання компонента може використовуватися спеціальний символ - прямокутник зі вставленими зліва двома дрібнішими прямокутниками. Залежність служить для подання факту наявності такого зв'язку, коли зміна одного елемента моделі впливає або призводить до зміни іншого елемента моделі. Відношення залежності на діаграмі компонентів зображується пунктирною лінією зі стрілкою, спрямованою від клієнта (залежного елемента) до джерела (незалежного елемента). Інтерфейс зображується колом, яка з'єднується з компонентом відрізком лінії без стрілок. Для представлення внутрішньої структури інтерфейсу його зображають у вигляді прямокутника класу зі стереотипом «інтерфейс» і можливими секціями атрибутів і операцій.

На рисунку 3.9 зображена діаграма компонентів ПЗ підсистеми. На цій діаграмі зображені взаємозв'язки між модулями ПЗ, які представлені файлами з байт-кодом (в Java кожен клас представляється в окремому файлі). Головний модуль, який запускає програму – це Main. Він створює об'єкти Manager та UseInterface. При створенні Manager використовує конфігураційний файл config.xml для налаштувань запуску програми. В цьому файлі міститься інформація про плагіни, які необхідно загрузити в програму (назва методу, шлях до плагіну, головний клас). Manager зчитує конфігураційний файл, після чого здійснює загрузку необхідних класів (тобто класів, про які вказано в конфігураційному файлі). Таким чином здійснюється динамічна загрузка коду, що надає системі можливості розширюваності, тобто користувач (сторонній розробник) може розробити плагін, який буде реалізовувати його власний метод прогнозування, або варіант імпорту з іншого джерела та прописати його в конфігураційному файлі, після чого можна буде його використовувати в програмі. При цьому класи, які розробляються стороннім розробником, повинні реалізовувати інтерфейси Import та Predict. Метод прогнозування на основі моделі клітинних автоматів та методи імпорту з файлу та з веб-ресурсу знаходяться в проекті і не потребують загрузки. Опис прикладу конфігураційного файлу знаходиться в додатку Б.

На діаграмі зображені класи ImportFromWeb та ImportFromFile, які реалізують інтерфейс Import, та використовуються для імпорту даних відповідно з веб-ресурсу та файлу. Також на діаграмі зображений клас AutomateFocast, який реалізує інтерфейс Predict та здійснює прогнозування на основі моделі клітинних автоматів. Клас AutomateFocast залежить від класів Converter, Automate, RSanalysis.

Клас DAO реалізує інтерфейс DataAccessObject і використовується для взаємодії з базою даних. Він взаємодіє з СУБД H2 через інтерфейс JDBC (Java DataBase Connectivity). JDBC-драйвер, який транслює JDBC-запити в протокол бази даних H2 та сама СУБД H2 знаходяться в бібліотеці h2.jar.

Клас Manager взаємодіє з класом DAO через інтерфейс DataAccessObject.

Клас Manager реалізує інтерфейс взаємодії з користувачем, методи якого використовуються класом UseInterface, який призначений для виводу екранної форми програми. Для побудови графіків клас UseInterface використовує бібліотеку jfreechart.jar.

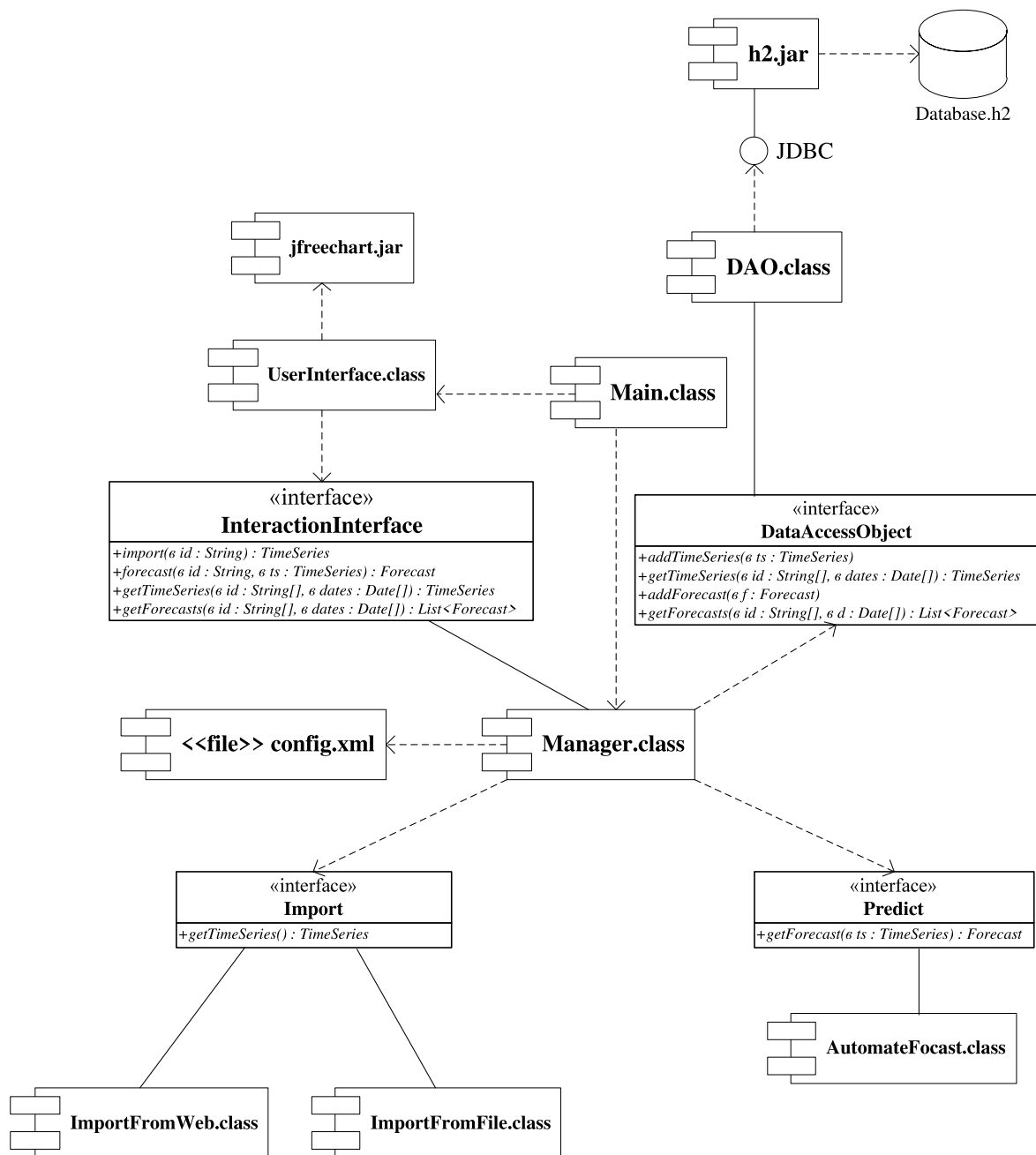


Рисунок 3.9 – Діаграма компонентів ПЗ підсистеми.

РОЗДІЛ 4 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПІДСИСТЕМИ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ КІЛЬКОСТІ СТУДЕНТІВ

4.1 Загальні положення

Підсистема короткострокового прогнозування кількості студентів призначена для використання в вищих навчальних закладах. Обґрунтувати доцільність розробки з економічної точки зору можна, розрахувавши собівартість підсистеми короткострокового прогнозування кількості студентів й додатковий прибуток, який буде отримано в результаті її впровадження. За цими показниками можна розрахувати річний економічний ефект, ефективність витрат на створення підсистеми короткострокового прогнозування й строк її окупності.

На даний момент сучасні обчислювальні засоби широко використовуються в повсякденній діяльності людини. Сьогодні обчислювальна техніка дозволяє вирішувати найрізноманітніші задачі: від звільнення людини від монотонної ручної праці до складних технічних розрахунків, що не під силу виконати одній людині. Сучасні засоби програмування дозволяють вирішувати задачі, що стояли в недавньому минулому перед цілими колективами наукових співробітників.

Розвиток нових технологій і ріст застосування обчислювальної техніки привели до широкого впровадження ЕОМ для вирішення проблем, пов'язаних з обробкою інформації, її контролем і аналізом.

Впровадження підсистеми призведе до значної економії, яка виражається в наступних показниках:

підвищення продуктивності праці;

вивільнення робочого часу;

вивільнення працівників для виконання іншої роботи.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування ціни розроблюваної системи. Очевидно, що інформаційна система являє собою дуже специфічний товар з безліччю особливостей.

Створення інформаційної системи вимагає одноразових витрат на її розробку, придбання необхідних технічних засобів і поточних витрат на функціонування системи. Економія від функціонування системи визначається з урахуванням витрат на її експлуатацію. Відношення цієї економії до витрат на створення системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за оптовими цінами, що діють на момент розрахунку, тарифами і ставками заробітної плати.

4.2. Розрахунок інвестиційних вкладень на впровадження підсистеми короткострокового прогнозування кількості студентів

4.2.1 Кошторис витрат на розробку підсистеми короткострокового прогнозування кількості студентів

Витрати на розробку підсистеми складаються з витрат на:

- заробітну платню розробника;
- амортизацію ЕОМ, на якій виконується розробка;
- експлуатацію цієї ЕОМ;
- засоби розробки;
- матеріали і комплектуючі.

Виходячи з того, що основна заробітна платня розробників програмного забезпечення складає 2500 грн./міс., вартість сучасної ПЕОМ складає 6000 грн. і вартість кіловат-години електроенергії складає 0,2436 грн., розрахуємо вартість розробки системи.

Розрахунок заробітної плати:

$$З_{зп} = ЗП_{роз} \cdot T_{роз}, \quad (4.1)$$

де $ЗП_{роз}$ – зарплатня розробника, за місяць; $T_{роз}$ – тривалість розробки (дослідження, створення, налагодження і впровадження).

Для розробки даної системи необхідно 4 місяці (за експертною оцінкою часу на розробку аналогічних систем). Кількість розробників – 1 людина.

З цього випливає, що загальна сума витрат на заробітну платню складе:

$$З_{зп} = ЗП_{роз} \cdot T_{роз} = 2500 \cdot 4 = 10000 \text{ (грн)}$$

Витрати на амортизацію ЕОМ, на якій виконується розробка, розраховується за формулою:

$$З_{амор} = Ц_{варт} \cdot A \cdot T_{роз}, \quad (4.2)$$

де $Ц_{варт} = 6000$ грн. – балансова вартість ЕОМ; $A = 60\%$ - амортизація за рік; термін служби ЕОМ – 5 років; $T_{роз} = 0,33$ року – час, необхідний для розробки системи.

З цього випливає, що загальна сума витрат на амортизацію складе:

$$З_{амор} = Ц_{варт} \cdot A \cdot T_{роз} = 6000 \cdot 0,6 \cdot 0,33 = 1188 \text{ (грн)}$$

Витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної енергії і розраховуються по формулі:

$$З_{елек} = П_{ЕОМ} \cdot T_{роз} \cdot N_{рд} \cdot N_{рг} \cdot E_{вл} \quad (4.3)$$

де $П_{ЕОМ} = 0,4$ кВт./год. – потужність ЕОМ; $T_{роз} = 4$ місяці – тривалість розробки; $N_{рд} = 20$ дні – число робочих днів у місяці; $N_{рг} = 8$ годин – число годин у робочому дні; $E_{вл} = 0,2436$ грн. – вартість 1 кВт/год електроенергії.

З цього випливає, що загальна сума витрат на електроенергію складе:

$$З_{елек} = 0,4 \cdot 4 \cdot 20 \cdot 8 \cdot 0,2436 = 62,36 \text{ (грн)}$$

Витрати на матеріали і комплектуючі вироби враховують витрати на папір для друкувальних пристроїв, на картридж і тонер для принтера, на флеш-накопичувач даних, на літературу (доступ до Internet), на програмні засоби розробки, а також на непередбачені витрати. Розрахунок приведений у таблиці 1 (ціни взяті відповідно до прайс-листів та договірних цін на даний вид матеріалів).

Таблиця 4.1 – Розрахунок витрат на матеріали і комплектуючі вироби

№ з/п	Найменування виробів	Вартість, грн.
1	Флеш-накопичувач	100,00
2	Папір для принтера	40,00
3	Картридж та тонер для принтера	250,00
4	Література (доступ до Internet)	100,00
5	NetBeans IDE	0
6	Непередбачені витрати	60,00
Разом		550,00

Загальний кошторис витрат на створення системи з урахуванням вищевказаного представлений у таблиці 4.2.

Таблиця 4.2 – Загальний кошторис витрат на створення системи

№ з/п	Найменування витрат	Вартість, грн.
1	Заробітна платня основна	10 000,00
2	Заробітна плата додаткова (20% від п.1)	2 000,00
3	Відрахування на соціальне страхування (38% від пп. 1 та 2)	4560
4	Витрати на амортизацію ЕОМ	1 188,00
5	Витрати на експлуатацію ЕОМ	62,36
6	Витрати на матеріали і комплектуючі вироби	550,00
7	Адміністративні витрати (50% від основної заробітної платні)	5 000,00
Разом на створення системи		23 360,36

4.2.2 Кошторис на технічні засоби, програмне забезпечення, які необхідно придбати для впровадження проекту

До витрат на впровадження системи можна віднести витрати на придбання технічного забезпечення та вартість програмного забезпечення.

Оскільки параметри технічних засобів, які вже є, відповідають вимогам, то їх вартість при розрахунку витрат враховувати не будемо.

Виходячи з вимог до програмного забезпечення, а також проаналізувавши цінову політику, можемо прийняти наступне (табл. 4.3).

Таблиця 4.3 – Загальний кошторис витрат на технічні засоби та програмне забезпечення

№	Найменування ПО	Кількість	Вартість, грн.
1	MS Windows XP	1	1520,00
2	Вартість устаткування		0
	Разом		1520,00

Витрати на програмне забезпечення склали 1520 грн.

4.2.3 Розрахунок витрат на монтаж і настройку системи, підготовку кадрів та ін.

До витрат на впровадження системи можна віднести витрати навчання кадрів, монтаж і налаштування мережі.

Згідно з досвідом створення аналогічних підсистем, приймаємо, що вартість підготовки кадрів дорівнює 300,00 грн, монтаж і налаштування локальної мережі – 1000, 00 грн. Таким чином затрати на монтаж і настройку системи, підготовку кадрів разом становлять: $1000, 00 + 300,00 = 1300, 00$ грн.

Таким чином інвестиційні витрати на створення та впровадження системи будуть становити: $23\ 360,36 + 1520,00 + 1300,00 = 26\ 180,36$ грн.

4.3 Розрахунок витрат на супроводження (експлуатацію) підсистеми

Після здачі проекту в експлуатацію система переходить на етап супроводження, який продовжується протягом всього строку її експлуатації . системи. В ході експлуатації можливе надходження запитів від замовника на зміну і доробку системи.

Витрати на супроводження системи дорівнюватимуть 2% від витрат на створення та впровадження системи: 523,6 грн.

4.4 Розрахунок показників економічної ефективності від впровадження результатів проектування

Основним показником економічної ефективності функціонування системи є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Економічна ефективність впровадження підсистеми короткострокового прогнозування кількості студентів очікується за рахунок вивільнення робочого часу працівників та підвищення продуктивності праці. Крім того, не піддається прямій грошовій оцінці зменшення кількості помилкових та необережних рішень, підвищення оперативності керування, поліпшення організації роботи та

своєчасне отримання необхідної інформації про пріоритетність варіантів рішень тощо.

Використання даної системи дозволяє вивільнити 0,6 робочого часу. Так як з системою працює один робітник, то система умовно вивільнить 0,6 робітника.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження системи вивільнить 0,6 робітника.

Зарплата 0,6 робітника у рік складає $\Delta C = 2500 \cdot 12 \cdot 0,6 = 18000$ (грн)

Річний економічний ефект розраховується за формулою 4.4.

$$E_{\text{рік}} = \Delta C - C_{\text{супр}} - E_{\text{п}} \cdot k \quad (4.4)$$

де $C_{\text{супр}}$ – вартість супроводження системи; $E_{\text{п}}$ – нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь – для обчислювальної техніки приймається рівним 0,5; k – додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи.

Підрахуємо річний економічний ефект:

$$E_{\text{рік}} = 18000 - 523,60 - 0,5 \cdot 26180,36 = 4386,22 \text{ (грн)}$$

Строк окупності системи розраховується за формулою 4.5.

$$I = \frac{k}{\Delta C - C_{\text{супр}}} \quad (4.5)$$

$$I = \frac{26180,36}{18000 - 523,60} = 1,5 \text{ (року)}$$

Тобто, система окупиться через 1,5 року, що дорівнює 18 місяцям.

За отриманим значенням річного економічного ефекту і строку окупності можна зробити висновок, що розробка підсистеми короткострокового прогнозування кількості студентів є економічно вигідною.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ ПРИ РОБОТІ З ПК

5.1 Загальні визначення та актуальність питання охорони праці при роботі з ПК

Згідно до ст.1 Закону "Про охорону праці", охорона праці є системою правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людей в процесі праці.

Основний принцип у галузі охорони праці – це пріоритет життя і здоров'я працівника по відношенню до результатів виробничої діяльності. При цьому роботодавці зобов'язані забезпечувати здорові і безпечні умови праці для своїх працівників, вживати всіх заходів щодо запобігання ризиків на робочих місцях, навчати працівників правилам і нормам охорони праці і виконувати інші вимоги законодавства.

Відповідно до вимог Закону «Про безпеку та охорону праці» на кожному підприємстві, незалежно від організаційно-правових форм і видів власності, повинна бути створена служба безпеки та охорони праці.

Основні цілі і завдання служби безпеки та охорони праці: розробка і здійснення комплексу соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів зі створення та забезпечення безпечних і здорових умов праці на виробництві.

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

5.2 Аналіз небезпечних і шкідливих факторів при роботі з ПК

З точки зору впливу небезпечних і шкідливих факторів при роботі з ПК можна виділити наступні:

- вплив електромагнітного та електростатичного випромінювань;
- вплив недостатньої освітленості робочого місця;
- вплив електричного струму;
- вплив шуму;
- вплив несприятливих умов мікроклімату;
- вплив нераціонального розташування устаткування і неправильної організації робочого місця.

Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може привести до професійного захворювання. Розглянемо більш детально вплив означених факторів на користувача ПК.

1) Вплив електромагнітного та електростатичного випромінювань.

При роботі ПК виникають два типи випромінювань: електростатичне та електромагнітне. Навколо електростатично зарядженого монітора підвищується концентрація пилу. Електромагнітне випромінювання створюється магнітними котушками, що знаходяться біля цокольної частини ЕПТ. Результати медичних досліджень показують, що електризований пил викликає запалення шкіри, а електромагнітне випромінювання приводить до порушення біологічних процесів (порушення сну, зниження кров'яного тиску, уповільнення серцевої діяльності, зміни складу крові).

2) Вплив недостатньої освітленості робочого місця.

Напружена зорова робота внаслідок нераціонального освітлення може бути причиною функціональних порушень у зоровому аналізаторі і привести до розладу зору, а у важких випадках – і до повної втрати. Втома органів зору залежить від ступеня напруженості процесів, що супроводжують зорове сприйняття. При роботі оператора ПК одним з основних навантажень на організм є навантаження на зір, тому дуже важливо приділити особливу увагу освітленості робочого місця.

У залежності від джерела світла виробниче висвітлення може бути трьох видів: природним, штучним і сполученим. При роботі з ПК вдень у ясну погоду використовується природне освітлення, яке здійснюється через світлові віконні прорізи. При нестачі природного світла, а також при роботі у вечірній час використовується штучне освітлення, джерелами якого є газорозрядні лампи. При такому освітленні величезне значення має правильне розташування джерел світла, їхній тип і кількість.

Правильна освітленість робочого місця сприяє високій працездатності користувача ПК і зводить до мінімуму шкідливий вплив на органи зору. Освітленість робочого місця, що рекомендується для користувача ПК складає 400–750 лк.

3) Вплив електричного струму.

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека ураження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватися правил електробезпеки.

Небезпека електричного струму на відміну від інших небезпек збільшується тим, що людина не в змозі без спеціальних приладів виявити напругу дистанційно, а також швидкоплинністю поразки – небезпека виявляється, коли людина вже уражена. Ураження струмом приводить до різних порушень в організмі, викликаючи як місцеву поразку тканин і органів, так і загальне ураження організму. Людина відчуває дратівну дію перемінного струму промислової частоти силою 0,6–1,5 мА і постійного струму 5–7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Для перемінного струму промислової частоти сила невідпускаючого струму знаходиться в межах 6–20 мА. Постійний струм не викликає невідпускаючого ефекту, але приводить до сильних болючих відчуттів, сила такого струму 15–80 мА і більше.

4) Вплив шуму.

Шумом прийнято називати всякий небажаний для людини звук, що заважає сприйняттю корисних сигналів. Шум являє собою безладне сполучення звуків різної інтенсивності і частоти. Він викликає труднощі з розпізнанням колірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5–12% продуктивність праці. Тривалий вплив шуму з рівнем звукового тиску 90 дБ знижує продуктивність праці на 30–60%.

Робота ПК може супроводжуватися шумами, викликаними деякими рухливими пристроями усередині комп'ютера, а також підключеними до нього. До таких пристроїв можна віднести вентилятори, пристрої читання дисків, принтери. Виникаючий шум, у залежності від інтенсивності, може привести до підвищення нервової напруги і зниження працездатності.

5) Вплив несприятливих умов мікроклімату.

Мікрокліматичні параметри (температура, відносна вологість і швидкість руху повітря) впливають на самопочуття і здоров'я людини, а також на надійність роботи засобів обчислювальної техніки.

При підвищеній температурі і вологості в приміщенні відбувається перегрівання тіла, що погіршує самопочуття і працездатність. Варто враховувати, що ПК, які знаходяться в приміщенні, сприяють збільшенню температури.

Для комфортної роботи приймається температура $23 (\pm 2)^{\circ}\text{C}$ й $18 (\pm 2)^{\circ}\text{C}$ в теплий і холодний час року відповідно, при відносній вологості $55 (\pm 5)\%$.

6) Вплив нераціонального розташування устаткування і неправильної організації робочого місця.

Робоче місце – це частина простору, у якій користувач ПК здійснює трудову діяльність і проводить велику частину робочого часу. Важливу роль грає планування робочого місця, яке повинне задовольняти вимогам зручності виконання робіт і економії енергії і часу користувача, зручності обслуговувань ЕОМ, дотримання правил охорони праці.

При плануванні робочого місця необхідно враховувати зони досяжності при розташуванні моніторів, клавіатури, маніпулятора «миша». Ці зони встановлюються на підставі антропологічних даних людського тіла. Під час роботи користувач ПК не повинен піддаватися впливові сторони подразників, якими можуть бути неадекватне фарбування пристроїв ЕОМ або приміщення.

5.3 Розрахунок системи загального освітлення приміщення з ПК

В приміщенні з ПК необхідно організувати змішане освітлення (комбінування природного і штучного освітлення). Природне освітлення створюється бічним освітленням через вікно. Штучне освітлення використовується при недостатньому природному освітленні. У даному приміщенні використовується загальне штучне освітлення. Розрахунок виконаємо за методом світлового потоку з урахування потоку, віддзеркаленого від стін та стелі.

Необхідно зробити розрахунок загального освітлення виробничої приміщення. Довжина $A = 10$ м, ширина $B = 7$ м, висота $H = 3$ м. Висота робочої поверхні $h_p = 1$ м. Для освітлення використовується світильні УПМ-5. Мінімальна освітленість лампи наколювання по нормах $E_{\min} = 100$ лк. Коефіцієнт відображення стелі $S_n = 50\%$, стін $S_c = 30\%$, робочої поверхні $S_p = 10\%$. Напруга мережі 220 В.

Відстань від стелі до робочої поверхні:

$$H_0 = H - h_p = 3 - 1 = 2 \text{ м.}$$

Відстань від стелі до світильника:

$$h_c = 0,2 \cdot H_0 = 0,2 \cdot 2 = 0,4 \text{ м.}$$

Висота підвісу світильника над освітленою поверхнею:

$$h = H_0 - h_c = 2 - 0,4 = 1,6 \text{ м.}$$

Висота підвісу світильника над підлогою:

$$H_n = h + h_p = 1,6 + 1 = 2,6 \text{ м.}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення $L/h = 1,5$.

Відстань між центрами світильників:

$$L = 1,5 \cdot h = 1,5 \cdot 1,6 = 2,4 \text{ м.}$$

Необхідна кількість світильників:

$$N = S / L^2 = (10 \cdot 7) / (2,4)^2 \approx 12.$$

Приймаємо кількість світильників: 12 (3 ряди по 4 світильники).

Індекс приміщення:

$$i = (A \cdot B) / (h \cdot (A + B)) = (10 \cdot 7) / (1,6 \cdot (10 + 7)) = 2,57.$$

Коефіцієнт використання світлового потоку $\eta = 0,69$.

Світловий потік однієї лампи:

$$\Phi = (E_{\min} \cdot S \cdot K_3 \cdot Z) / (N \cdot \eta) = (100 \cdot 70 \cdot 1,3 \cdot 1,15) / (12 \cdot 0,69) = 1264 \text{ лм.}$$

За знайденим світовим потоком із таблиці «Параметри ламп накаливання загального призначення з розрахунковими напругами 130 і 220 В» обираємо лампу накаливання потужністю 300 Вт, яка має світовий потік 2090, що найбільше близький до розрахункового.

Фактична освітленість:

$$E_{\Phi} = E_{\min} \cdot \Phi_{\text{л}} / \Phi_p = 100 \cdot 2090 / 1264 = 165,35 \text{ лк.}$$

Загальна потужність:

$$P_{\text{заг}} = P_{\text{л}} \cdot N = 300 \cdot 12 = 3600 \text{ Вт} = 3,6 \text{ кВт.}$$

Таким чином, для оптимальної роботи в приміщенні необхідно встановити 3 ряди по 4 світильники. Загальна потужність усіх світильників складе 3,6 кВт.

5.4 Техніка безпеки при роботі з ПК

При виконанні робіт на комп'ютерах необхідно дотримуватись вимог загальної та даної інструкції з охорони праці.

До самостійної роботи на комп'ютерах допускаються особи, які пройшли медичний огляд, навчання по професії, вступний інструктаж з охорони праці та первинний інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя, періодичні медичні огляди один раз на два роки.

Під час роботи на комп'ютерах можуть діяти такі небезпечні та шкідливі

фактори, як:

- фізичні;
- психофізіологічні.

Основним обладнанням робочого місця користувача комп'ютера є монітор, системний блок та клавіатура.

Робочі місця мають бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1м, між собою на відстані не менше 1,5 м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

Робочі місця слід розташовувати так, щоб уникнути попадання в очі прямого світла. Джерела освітлення рекомендується розташовувати з обох боків екрану паралельно напрямку погляду. Для уникнення світлових відблисків екрану, клавіатури в напрямку очей користувача, від світильників загального освітлення або сонячних променів, необхідно використовувати антиполюсківі сітки, спеціальні фільтри для екранів, захисні козирки, на вікнах - жалюзі.

Фільтри з металевої або нейлонової сітки використовувати не рекомендується, тому що сітка спотворює зображення через інтерференцію світла. Найкращу якість зображення забезпечують скляні поляризаційні фільтри. Вони усувають практично всі відблиски, роблять зображення чітким і контрастним.

При роботі з текстовою інформацією (в режимі введення даних та редагування тексту, читання з екрану) найбільш фізіологічним правильним є зображення чорних знаків на світлому (чорному) фоні.

Монітор повинен бути розташований на робочому місці так, щоб поверхня екрана знаходилася в центрі поля зору на відстані 400-700 мм від очей користувача. Рекомендується розміщувати елементи робочого місця так, щоб витримувалася однакова відстань очей від екрана, клавіатури, тексту.

Зручна робоча поза при роботі з комп'ютером забезпечується регулюванням висоти робочого столу, крісла та підставки для ніг. Рациональною робочою позою може вважатися таке положення, при якому ступні працівника

розташовані горизонтально на підлозі або підставці для ніг, стегна зорієнтовані у горизонтальній площині, верхні частини рук - вертикальні. Кут ліктьового суглоба коливається в межах 70-90°, зап'ястя зігнуті під кутом не більше ніж 20°, нахил голови 15-20°.

Для нейтралізації зарядів статичної електрики в приміщенні, де виконується робота на комп'ютерах, в тому числі на лазерних та світлодіодних принтерах, рекомендується збільшувати вологість повітря за допомогою кімнатних зволожувачів. Не рекомендується носити одяг з синтетичних матеріалів.

Вимоги безпеки перед початком роботи:

- увімкнути систему кондиціонування в приміщенні;
- перевірити надійність встановлення апаратури на робочому столі. Повернути монітор так, щоб було зручно дивитися на екран – під прямим кутом (а не збоку) і трохи зверху вниз, при цьому екран має бути трохи нахиленим, нижній його край ближче до оператора;
- перевірити загальний стан апаратури, перевірити справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана;
- відрегулювати освітленість робочого місця;
- відрегулювати та зафіксувати висоту крісла, зручний для користувача нахил його спинки;
- приєднати до системного блоку необхідну апаратуру. Усі кабелі, що з'єднують системний блок з іншими пристроями, слід вставляти та виймати при вимкненому комп'ютері;
- ввімкнути апаратуру комп'ютера вимикачами на корпусах в послідовності: монітор, системний блок, принтер (якщо передбачається друкування);
- відрегулювати яскравість свічення монітора, мінімальний розмір світної точки, фокусування, контрастність. Не слід робити зображення надто яскравим, щоб не втомлювати очей.

Рекомендується:

- яскравість свічення екрана - не менше 100Kг/M2;
- відношення яскравості монітора до яскравості оточуючих його поверхонь в робочій зоні - не більше 3:1;
- мінімальний розмір точки свічення не більше 0,4 мм для монохромного монітора і не менше 0,6 мм для кольорового, контрастність зображення знаку – не менше 0,8.

При виявленні будь-яких несправностей роботу не розпочинати, повідомити про це керівника.

Вимоги безпеки під час виконання роботи:

- необхідно стійко розташовувати клавіатуру на робочому столі, не опускати її хитання. Під час роботи на клавіатурі сидіти прямо, не напружуватися;

- для забезпечення несприятливого впливу на користувача пристроїв типу "миша" належить забезпечувати вільну велику поверхню столу для переміщення "миші" і зручного упору ліктьового суглоба;

- не дозволяються по сторонні розмови, подразнюючі шуми;

- періодично при вимкненому комп'ютері прибирати ледь змоченою мильним розчином бавовняною ганчіркою порох з поверхонь апаратури. Екран ВДТ та захисний екран протирають ганчіркою, змоченою у спирті. Не дозволяється використовувати рідинні або аерозольні засоби чищення поверхонь комп'ютера.

Забороняється:

- самостійно ремонтувати апаратуру. Ремонт апаратури здійснюється спеціалістами з технічного обслуговування комп'ютера;

- класти будь-яку предмети на апаратуру комп'ютера;

- закривати будь-чим вентиляційні отвори апаратури, що може призвести до її перегрівання і виходу з ладу.

Для зняття статичної електрики рекомендується час від часу доторкатися до металевих поверхонь.

Розташувати принтер необхідно поруч з системним блоком таким чином,

щоб з'єднувальний шнур не був натягнутий. Забороняється ставити принтери на системний блок.

Для досягнення найбільш чистих, з високою розподільністю зображень і щоб не зіпсувати апарат, має використовуватися папір, вказаний в інструкції до принтера. При застряганні паперу потрібно відкрити кришку і обережно витягнути лоток з папером.

Згідно з інструкцією фірми-виробника потрібно дотримуватися правил зберігання картриджа.

Забороняється:

- зберігати картриджі без упаковки;
- ставити картриджі вертикально;
- перевертати картридж етикеткою донизу;
- відкривати кришку валика і доторкатися до нього;
- самому заповнювати використаний картридж.

Вимоги безпеки після закінчення роботи:

- закінчити та записати у пам'ять комп'ютера файл, що знаходиться в роботі;

- вимкнути принтер та інші периферійні пристрої. Штепсельні вилки витягнути з розеток. Накрити клавіатуру кришкою запобігання попаданню в неї пилу;

- прибрати робоче місце;

- ретельно вимити руки теплою водою з милом;

- вимкнути кондиціонер, освітлення і загальне електроживлення;

- пройти в спеціально обладнаному приміщенні сеанс психофізіологічного розвантаження і зняття втоми з виконанням спеціальних вправ аутогенного тренування.

РОЗДІЛ 6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступна частина

Під охороною навколишнього середовища розуміють сукупність міжнародних, державних і регіональних правових актів, інструкцій і стандартів, які доводять загальні юридичні вимоги до кожного конкретного забруднювача і забезпечують його зацікавленість у виконанні цих вимог, конкретних природоохоронних заходів по втіленню в життя цих вимог.

Тільки якщо всі ці складові частини відповідають один одному за змістом і темпами розвитку, тобто складаються в єдину систему охорони навколишнього природного середовища, можна розраховувати на успіх.

Оскільки не була вирішена вчасно завдання охорони природи від негативного впливу людини, тепер все частіше постає завдання захисту людини від впливу змінилася природного середовища. Обидва ці поняття інтегруються в терміні «охорона навколишнього (людини) природного середовища».

Охорона навколишнього природного середовища складається з:

правової охорони, що формулює наукові екологічні принципи у вигляді юридичних законів, обов'язкових для виконання;

матеріального стимулювання природоохоронної діяльності з ціллю зробити її економічно вигідною для підприємств;

інженерної охорони, що розробляє природоохоронну та ресурсозберігаючу технологію і техніку.

Основними принципами охорони навколишнього природного середовища є:

- пріоритет забезпечення сприятливих екологічних умов для життя, праці та відпочинку населення;

- науково обґрунтоване поєднання екологічних і економічних інтересів суспільства;

- облік законів природи і можливостей самовідновлення і самоочищення її ресурсів;

- недопущення незворотних наслідків для охорони природного середовища і здоров'я людини;
- право населення та громадських організацій на своєчасну і достовірну інформацію про стан навколишнього середовища та негативному впливі на неї і на здоров'я людей різних виробничих об'єктів;
- невідворотність відповідальності за порушення вимог природоохоронного законодавства.

6.2 Забруднення навколишнього середовища біля малого підприємства, розташованого у центрі міста Миколаєва.

Забруднення навколишнього середовища (ЗОС) – надходження в природне середовище будь-яких твердих, газоподібних речовин, мікроорганізмів або енергій (у вигляді звуків, шумів, електромагнітного й радіоактивного випромінювання) у кількостях, що викликають зміну складу й властивостей компонентів природи й (або) чинять шкідливий вплив на людину, флору й фауну.

Види ЗОС різні й різноманітні: викиди в атмосферу хімічних сполук і сумішей; надходження у водне середовище всіляких виробничих і комунально-побутових відходів, влучення в неї нафтових продуктів; засмічування ландшафтів сміттям і пакувальними матеріалами; забруднення полів, луків, лісів і водних об'єктів пестицидами; підвищення рівня іонізуючої радіації, виробничих і побутових шумів, вібрацій, а також нагромадження тепла в атмосфері.

По характері походження або впливу на організми й екосистеми виділяють наступні види ЗОС: антропогенне, природне, механічне, фізичне, біологічне, хімічне, шумове, радіоактивне, електромагнітне

Відомо більше 200 забруднюючих атмосферу речовин, причому в міру освоєння нових технологічних процесів і створення нових продуктів виробництва число забруднюючих речовин зростає. Цьому сприяють також

реакції в атмосфері й інших середовищах між забруднюючими речовинами під впливом сонячного світла, окислювально-відновних процесів і інших реакцій

В центрі міст як правило дуже інтенсивний рух автомобільного транспорту, тому розглянемо забруднення, які пов'язані з цим. Найбільш значимі фактори негативного впливу автомобільного транспорту на людину і навколишнє середовище наступні:

- забруднення повітря;
- забруднення навколишнього середовища;
- шум, вібрація;
- виділення тепла (розсіювання енергії).

Основна причина забруднення повітря полягає у неповному і нерівномірному згорянні палива. Всього 15% його витрачається на рух автомобіля, а 85% «летить на вітер». До того ж камери згорання автомобільного двигуна – це своєрідний хімічний реактор, що синтезує отруйні речовини і викидає їх в атмосферу.

Рухаючись зі швидкістю 80-90 км/год в середньому автомобіль перетворює на вуглекислоту стільки ж кисню, скільки 300-350 чоловік. Але справа не тільки в вуглекислоті. Річний вихлоп одного автомобіля – це 800 кг окису вуглецю, 40 кг окислів азоту і більше 200 кг різних вуглеводнів. У цьому наборі дуже підступна окис вуглецю. Через високу токсичність її допустима концентрація в атмосферному повітрі не повинна перевищувати 1 мг/м³.

Рівень загазованості магістралей залежить від інтенсивності руху автомобілів, ширини і рельєфу вулиці, швидкості вітру, частки вантажного транспорту та автобусів в загальному потоці і інших факторів. При інтенсивності руху 500 транспортних одиниць на годину концентрація окису вуглецю на відкритій території на відстані 30-40 м від автомагістралі знижується в 3 рази і досягає норми.

Ускладнене розсіювання викидів автомобілів на тісних вулицях. У результаті практично всі жителі міста відчувають на собі шкідливий вплив забрудненого повітря.

Чадний газ і окисли азоту, настільки інтенсивно виділяються на перший погляд безневинним голубуватим димком глушника автомобіля – ось одна з основних причин головних болів, втоми, невмотивованого роздратування, низькою працездатності. Сірчистий газ здатний впливати на генетичний апарат, сприяючи безпліддя і природженої потворності, а всі разом ці чинники ведуть до стресів, нервовим проявам, прагненню до усамітнення, байдужості до найближчих людей. У великих містах також більш широко поширені захворювання органів кровообігу і дихання, інфаркти, гіпертонія і новоутворення. За розрахунками фахівців, «внесок» автомобільного транспорту в атмосферу становить до 90% по окису вуглецю і 70% по окису азоту. Автомобіль також додає в ґрунт і повітря важкі метали та інші шкідливі речовини.

В умовах сильного міського шуму відбувається постійна напруга слухового аналізатора. Це викликає збільшення порогу чутності (10 дБ для більшості людей з нормальним слухом) на 10-25дБ.

Високі рівні шуму в міському середовищі є одним з агресивних подразників центральної нервової системи. Міський шум справляє негативний вплив і на серцево-судинну систему. Ішемічна хвороба серця, гіпертонічна хвороба, підвищений вміст холестерину в крові зустрічаються частіше в осіб, що проживають у гучних районах.

Найбільші рівні шуму 90-95 дБ відзначаються на магістральних вулицях міст з середньою інтенсивністю руху 2-3 тис. і більше транспортних одиниць на годину.

Рівень вуличних шумів обумовлюється інтенсивністю, швидкістю і характером (складом) транспортного потоку. Крім того, він залежить від планувальних рішень (поздовжній і поперечний профіль вулиць, висота і щільність забудови) і таких елементів благоустрою, як покриття проїзної частини та наявність зелених насаджень. Кожен з цих факторів здатний змінити рівень транспортного шуму в межах до 10 дБ.

Шум, що виникає на проїжджій частині магістралі, поширюється не тільки на при магістральну територію, але і вглиб житлової забудови. Так, в зоні найбільш сильного впливу шуму знаходяться частини кварталів і мікрорайонів, розташованих уздовж магістралей загальноміського значення (еквівалентні рівні шуму від 67,4 до 76,8 дБ). Рівні шуму, заміряні в житлових кімнатах при відкритих вікнах, орієнтованих на зазначені магістралі, всього на 10-15 дБ нижче.

6.3 Заходи щодо зменшення забруднення

Увага вчених, інженерів, агрономів, інших фахівців і громадськості зосереджено на усуненні всіх видів забруднюючих речовин, що представляють найбільшу небезпеку для здоров'я людей. Способи й шляхи боротьби із ЗОС різноманітні: пристрої очисних споруджень, установка фільтрів, створення й впровадження безвідхідних технологій, замкнених циклів водокористування, заміна хімічних способів боротьби зі шкідниками й хворобами сільськогосподарських і лісових рослин біологічними, звукопоглинаючі пристрої, використання відходів для одержання корисної продукції, пошуки нових видів палива, використання для технічних і побутових потреб тепла геотермальних джерел, сили вітру, морських припливів, сонячного тепла, установлення режиму користування автомобільним і авіаційним транспортом

Проблема забруднення навколишнього середовища автомобільним транспортом цілком очевидно є глобальною. По всьому світу кількість автомобілів з кожним днем збільшується в геометричній прогресії. Все більше і більше людей мають свою власну машину. Але багато хто зовсім не замислюється про те, до чого все це врешті-решт приведе. Для того, щоб зберегти людству автомобіль, необхідно якщо не виключити, то звести до мінімуму шкідливі викиди. Робота в цьому напрямі ведеться в усьому світі і дають певні результати. Автомобілі випускаються в даний час в промислово розвинених країнах, викидають шкідливих речовин у 10-15 разів менше, ніж 10-15 років тому. У всіх розвинених країнах відбувається посилення нормативів на шкідливі викиди при роботі двигуна. У 2000 р. введені більш суворі норми.

Відбувається не тільки кількісне посилення норм, але і їх якісна зміна. Так, замість обмежень по димності введено нормування твердих частинок, на поверхні яких адсорбуються небезпечні для здоров'я людини ароматичні вуглеводні і зокрема, канцерогенний бенз(а)пірен. Постійно розширюється перелік речовин, зміст яких має знаходитися під контролем.

В даний час йде боротьба з автомобільною небезпекою. Конструюються фільтри, розробляються нові види пального, що містять менше свинцю.

Загальні шляхи зниження екологічних збитків від транспорту полягають в наступному

- 1) оптимізація руху міського транспорту
- 2) розробка альтернативних енергетичних джерел;
- 3) опалювання і очищення органічного палива;
- 4) створення (модифікація) двигунів, що використовують альтернативні палива;
- 5) захист від шуму;
- 6) економічні ініціативи з управління автомобільним парком і рухом.

ВИСНОВКИ

Прогнозування контингенту абітурієнтів є складним, але необхідним процесом для успішного функціонування будь-якого закладу вищої освіти. Проведене дослідження дозволило зробити такі висновки:

- Прогнозування є багатофакторним процесом. На кількість та якість абітурієнтів впливає широкий спектр факторів, включаючи демографічні зміни, економічні умови, соціальні тренди, державну політику в галузі освіти, конкуренцію між ЗВО, імідж університету, а також індивідуальні характеристики абітурієнтів.

- Існують різноманітні методи прогнозування. Від простих статистичних методів до складних моделей машинного навчання. Вибір методу залежить від наявності даних, поставлених завдань та ресурсів університету.

- Точність прогнозування можна підвищити за рахунок використання комплексних моделей. Комбінування різних методів прогнозування та включення в модель якомога більшої кількості факторів дозволяє отримати більш точні результати.

- Важливим етапом є збір та обробка даних. Якість прогнозу безпосередньо залежить від якості та повноти використаних даних.

- Прогнозування має бути динамічним процесом. Світ постійно змінюється, тому моделі прогнозування потребують регулярної перевірки та коригування.

- Прогнозування є не лише інструментом для планування, але й для прийняття стратегічних рішень. На основі прогнозів університет може розробляти нові освітні програми, адаптувати навчальні плани, змінювати маркетингову стратегію та ефективніше використовувати ресурси.

Рекомендації для ЗВО:

- Створити систему збору та аналізу даних про абітурієнтів. Це дозволить виявити тренди та закономірності у виборі спеціальностей, географії вступників, мотивації тощо.

- Залучати до процесу прогнозування фахівців з різних галузей. Це можуть бути математики, статистики, соціологи, маркетинологи.

- Регулярно оцінювати точність прогнозів та вносити необхідні корективи в моделі.

- Використовувати результати прогнозування для розробки стратегії розвитку університету.

- Співпрацювати з іншими університетами та дослідницькими інститутами для обміну досвідом та розробки спільних моделей прогнозування.

Перспективи подальших досліджень:

- Розробка інтерактивних систем прогнозування, які дозволять користувачам самостійно змінювати параметри моделей та отримувати персоналізовані прогнози.

- Вивчення впливу нових технологій (наприклад, штучного інтелекту) на процес прогнозування.

- Дослідження поведінки абітурієнтів в умовах невизначеності (наприклад, під час пандемії).

Прогнозування контингенту абітурієнтів є складним, але необхідним інструментом для забезпечення сталого розвитку закладів вищої освіти. Застосування сучасних методів прогнозування та комплексного підходу до аналізу даних дозволяє підвищити точність прогнозів та приймати більш обґрунтовані управлінські рішення.

Список використаної літератури

1. Найман Э. Л. Трейдер-Инвестор. – Київ: ВІРА-Р, 2000. – 640 с.
2. Холден К., Піл Д.А., Томпсон Дж.Л. Економічне прогнозування: Вступ. – К.: Інформтехніка ЕМЦ, 1996. – 216с.
3. Черняк О.І., Ставицький А.В. Динамічна економетрика: Навчальний посібник. – К.:КВІЦ, 2000. – 120 с.
4. Афанасьєв В.Н., Юзбашев М.М.: Аналіз часових рядів і прогнозування: Підручник. – К.: Финанси і статистика, 2001. – 228 с.
5. Дуброва Т.А. Статистичні методи прогнозування. – К.: Юніті-Дана, 2003. – 206 с.
6. Лукашин Ю.П. Адаптивні методи короткострокового прогнозування часових рядів: Підручник. – К.: Фінанси і статистика, 2003. – 416 с.

ДОДАТОК А

Загальний опис підсистеми короткострокового прогнозування кількості студентів

Підсистема короткострокового прогнозування кількості студентів призначена для здійснення прогнозування, на основі якого приймаються рішення по плануванню бюджетних місць, розрахунку затрат. Метою створення підсистеми короткострокового прогнозування кількості студентів є підвищення ефективності прогнозування кількості студентів.

Підсистема здійснює прогнозування кількості студентів на основі нечіткого моделювання Т-моделі, проте існує можливість її розширення шляхом написання плагінів, які реалізують інші методи прогнозування.

Підсистема виконує наступні задачі:

- прогнозування кількості студентів на один рік;
- ведення історії прогнозів.

Дана підсистема може бути розгорнута одному комп'ютері, на якому необхідно встановити віртуальну машину Java. Підсистема короткострокового прогнозування кількості студентів може працювати під керівництвом різних платформ, так як при її створенні використовувалася мова Java та СУБД H2, яка також написана на Java. Розробка і тестування підсистеми проводилися на Windows 7, використовуючи Sun JDK 1.6.

Зокрема підсистема короткострокового прогнозування кількості студентів може використовуватись керівниками інститутів, кафедр для зменшення ризику. Вона може бути інтегрована в інформаційну систему вищого навчального закладу при замовленні бюджетних місць.

Вимоги до технічного забезпечення комп'ютера користувача (мінімальна конфігурація): процесор – Pentium IV 1,6 ГГц; оперативна пам'ять – 512 Мб; жорсткий диск – 40 Гб. Для імпорту даних з веб-ресурсу необхідне підключення до Internet.

ДОДАТОК Б

Керівництво користувача

Підсистема короткострокового прогнозування кількості студентів призначена для здійснення прогнозування, на основі якого приймаються рішення по затратам.

Основними функціями підсистеми короткострокового прогнозування є :

- імпорт даних (з веб-ресурсу, зі зовнішнього файлу);
- прогнозування кількості студентів певним методом;
- побудова графіків прогнозу й похибки прогнозу кількості студентів;
- ведення історії прогнозів кількості студентів .

Задля коректної роботи підсистеми необхідно коректно встановити все необхідне програмне забезпечення. Підсистема короткострокового прогнозування кількості студентів може працювати під керівництвом різних платформ, так як при її створенні використовувалася мова Java та СУБД H2, яка також написана на Java.

Програмне забезпечення підсистеми розгортається на комп'ютері користувача, на якому необхідно встановити віртуальну машину Java. Вимоги до технічного забезпечення комп'ютера користувача (мінімальна конфігурація):

- Процесор – Pentium IV 1,6 ГГц;
- Оперативна пам'ять – 512 Мб;
- Жорсткий диск – 40 Гб.

Для імпорту даних з веб-ресурсу необхідне підключення до Internet.

Запуск підсистеми виконується виконанням файлу Forecast.bat, який запускає програму, що міститься у Forecast.jar.

При запуску підсистеми короткострокового прогнозування кількості студентів на моніторі з'являється головна форма, зображена на рисунку Б.1.

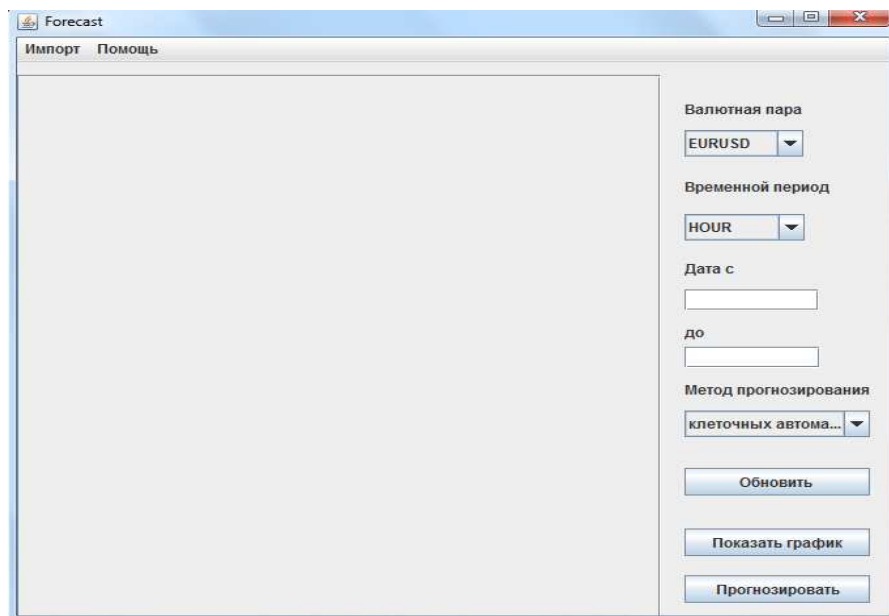


Рисунок Б.1 – Основне вікно програми

Імпорт даних про кількості студентів здійснюється шляхом вибору необхідного джерела (із веб-ресурсу або файлу) в меню «Імпорт».

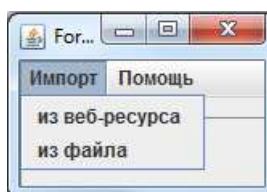


Рисунок Б.2 – Меню «Імпорт»

При виборі пункту «із веб-ресурсу» в меню «Імпорт» з'являється діалогове вікно (рисунок Б.3) для вводу необхідних даних для імпорту. Дані про кількість студентів загружаються з сайту

При виборі пункту «із файлу» в меню «Імпорт» з'являється діалогове вікно (рисунок Б.4) для вибору файлу формату CSV. В файлі в першому рядку через пробіл повинно бути вказано кількість студентів. Далі в кожному наступному рядку через крапку з комою повинні бути записані дані. Після вибору необхідного файлу потрібно натиснути кнопку «Імпортувати файл». Для скасування імпорту файлу необхідно натиснути «Cancel».

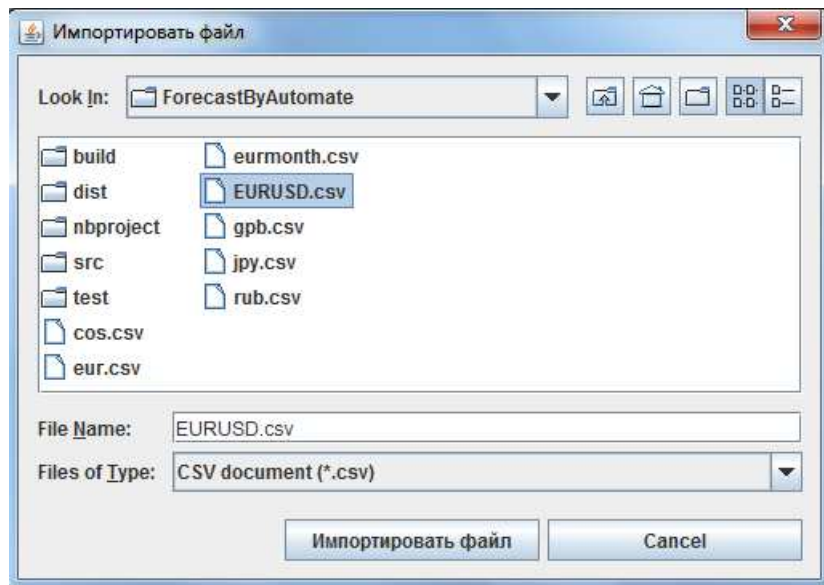


Рисунок Б.4 – Діалогове вікно для імпорту даних з файлу

Справа на головній формі знаходиться панель, яка містить необхідні елементи для перегляду даних про кількості студентів, здійснення прогнозування та перегляду історії прогнозування. Для отримання табличного відображення кількості студентів необхідно вибрати із розкривного списку певну валютну пару та часовий період, ввести дати, з яку і по яку необхідно відобразити дані, та натиснути кнопку «Оновити». Після вказаних дій на головній формі справа з'явиться таблиця з обраними даними (рисунок Б.5).

Forecast					Импорт Помощь	
Дата	Открытия	Закрытия	Минимальная	Максимальная		
26.10.2010 00:00	1.39539	1.38621	1.38251	1.3981		
27.10.2010 00:00	1.38618	1.37739	1.37338	1.38763		
28.10.2010 00:00	1.37739	1.39282	1.37671	1.39448		
29.10.2010 00:00	1.39284	1.39459	1.38068	1.39502		
01.11.2010 00:00	1.39703	1.38977	1.38636	1.40106		
02.11.2010 00:00	1.38975	1.40312	1.38812	1.40573		
03.11.2010 00:00	1.4031	1.41248	1.39916	1.41797		
04.11.2010 00:00	1.41238	1.42125	1.4102	1.42813		
05.11.2010 00:00	1.42124	1.40314	1.40238	1.42478		
08.11.2010 00:00	1.40538	1.39123	1.38874	1.40846		
09.11.2010 00:00	1.39131	1.37633	1.37483	1.3973		
10.11.2010 00:00	1.37637	1.37791	1.36711	1.38248		
11.11.2010 00:00	1.37792	1.36611	1.36369	1.382		
12.11.2010 00:00	1.36619	1.369	1.35735	1.37758		
15.11.2010 00:00	1.36829	1.35789	1.35634	1.37497		
16.11.2010 00:00	1.35789	1.34821	1.34475	1.36547		
17.11.2010 00:00	1.34817	1.35294	1.34602	1.35646		
18.11.2010 00:00	1.3529	1.36427	1.35289	1.36666		
19.11.2010 00:00	1.3643	1.36718	1.36081	1.37308		
22.11.2010 00:00	1.37355	1.36236	1.35765	1.37853		
23.11.2010 00:00	1.36236	1.33802	1.3361	1.3628		
24.11.2010 00:00	1.338	1.33364	1.32845	1.34202		
25.11.2010 00:00	1.33362	1.33443	1.32862	1.3386		
26.11.2010 00:00	1.33453	1.32402	1.32003	1.33569		
29.11.2010 00:00	1.3275	1.31061	1.30639	1.33009		
30.11.2010 00:00	1.31063	1.29811	1.29694	1.31491		
01.12.2010 00:00	1.29809	1.31254	1.29707	1.31806		
02.12.2010 00:00	1.31252	1.32149	1.30603	1.32462		
03.12.2010 00:00	1.32136	1.34136	1.31929	1.34373		
06.12.2010 00:00	1.34078	1.33088	1.32463	1.34216		
07.12.2010 00:00	1.33088	1.32652	1.32536	1.33007		

Валютная пара
EURUSD
Временной период
DAY
Дата с
26.10.2010 00:00
до
26.01.2011 00:00
Метод прогнозирования
клеточных автома...
Обновить
Показать график
Прогнозировать

Рисунок Б.5 – Табличне відображення кількості студентів

Для відображення графіку необхідно натиснути кнопку «Показати графік» (рисунок Б.6).

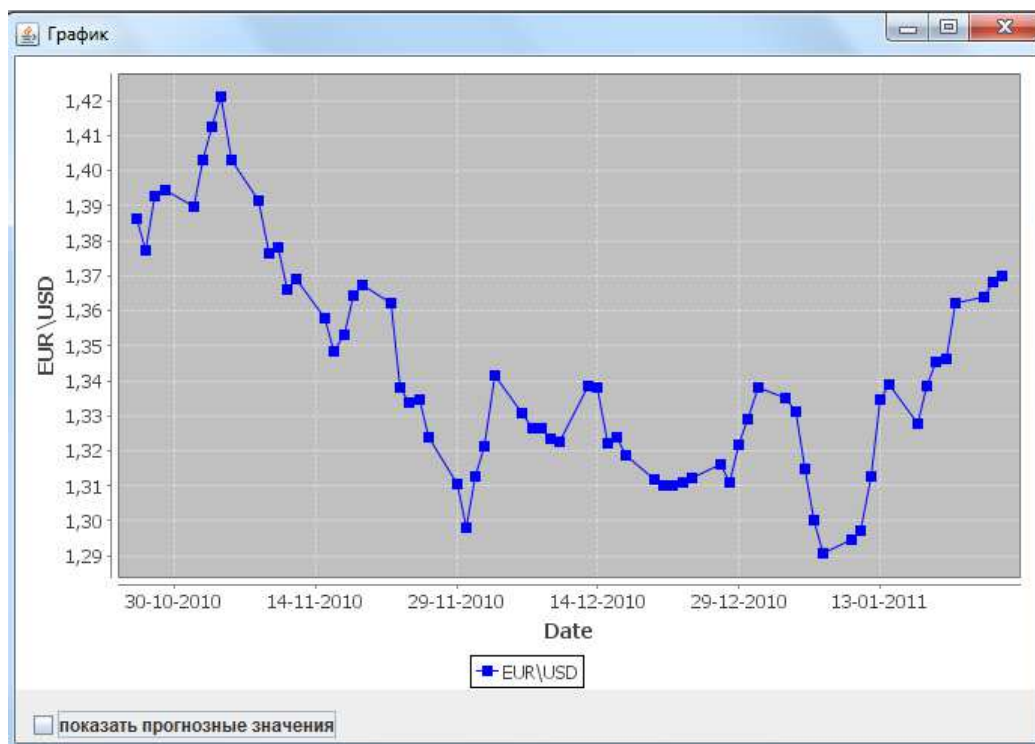


Рисунок Б.6 – Графік реальних значень кількості студентів

Для відображення історії прогнозів необхідно на формі, яка зображена на рисунку Б.6, відмітити прапорець «Історія прогнозів» (рисунк Б.7).

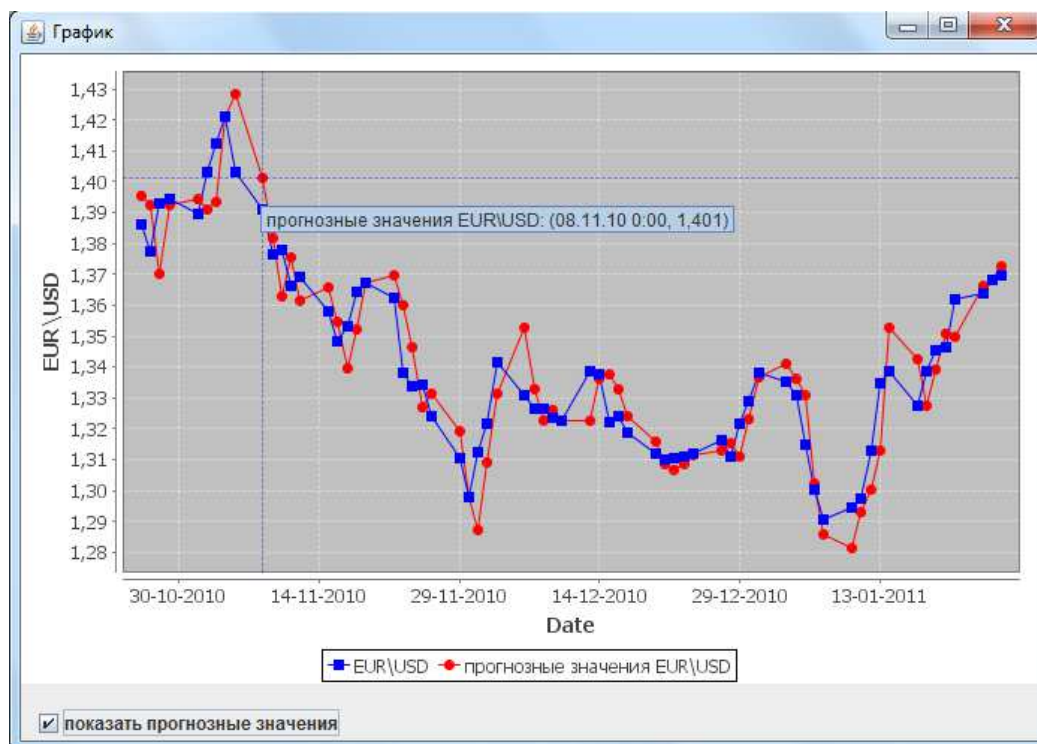


Рисунок Б.7 – Графік прогнозних та реальних значень кількості студентів

Для отримання прогнозу на наступний часовий період необхідно обрати з розкривного списку певний метод прогнозування та натиснути кнопку «Прогнозувати» (рисунк Б.8).

Рисунок Б.8 – Форма «Прогноз»

Додаток В

Програма і методика випробувань

В.1 Об'єкт та мета випробувань

Об'єкт випробувань – програмне забезпечення підсистеми короткострокового прогнозування кількості студентів.

Мета випробувань: перевірка придатності системи до використання по призначенню, її відповідність заданим вимогам і документації; перевірка працездатності і правильності роботи системи.

В.2 Вимоги до програми

Програма повинна задовольняти наступним функціональним характеристикам:

- імпорт даних (із зовнішнього файлу, веб-ресурсу);
- прогнозування кількості студентів на наступний часовий період (рік);
- збереження історії прогнозів кількості студентів та побудова графіку прогнозів.

В.3 Вимоги до програмної документації

У комплект програмної документації повинні входити наступні документи:

- опис програми;
- керівництво користувача;
- програма й методика випробувань.

В.4 Засоби і порядок проведення випробувань

Випробування будуть проводитись на персональному комп'ютері з процесором Intel Core 2 Duo P8700 2,57 ГГц, оперативною пам'яттю 2Гб в операційній системі Windows 7.

Випробування програми проводяться протягом двох тижнів. Замовник і/або його представник самостійно проводять тестування усіх функцій підсистеми в присутності розробника. При наявності можливості, у процесі тестування можуть додатково брати участь програмісти, тестувальники, які не брали участь у розробці ПЗ підсистеми.

Перед початком (не пізніше чим за 3 дні) випробувань уточнюється кінцевий склад групи, що буде проводити випробування. Кожний член групи отримує необхідну документацію до ПЗ підсистеми.

Порядок проведення випробувань:

- Виконуються контрольні тести в довільному порядку. Результати заносяться у відповідний журнал.
- Робиться аналіз отриманих результатів і встановлюється відповідність програми заданим вимогам і системним документам.

При виявленні помилок у роботі ПЗ підсистеми складається їх перелік і обговорюється термін їх виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливо використання нових чи додаткових тестів).

В.2 Методи випробувань

Метод проведення випробувань наступний. Спочатку розробник повинен перевірити кожен функціональний компонент програми окремо; потім – протестувати взаємодію компонента з іншими компонентами системи на автономному комп'ютері; і тільки потім – перевірити роботу всієї програми. Усі ці дії виконуються розробником без участі майбутнього користувача. Розробник не зобов'язаний надавати тести і протоколи результатів тестування кому-небудь.

При прийманні замовник та/або його представники самостійно проводять випробування усіх функцій підсистеми методом «чорного ящика» у присутності розробника. Випробувальні тести складаються замовником та/або його

представниками. При цьому перевіряється функціонування програми з правильними (коректними) і помилковими даними.

Можливий перелік тестів, які дозволяють протестувати основні функції програми:

- перегляд форми імпорту даних;
- перевірка адекватності імпорту даних з БД;
- виконання імпорту даних з зовнішнього файлу;
- виконання імпорту даних з веб-ресурсу;
- прогнозування часового ряду кількості студентів ;
- побудова графіку часових рядів кількості студентів та їх прогнозів;

Якщо користувач вводить помилкові дані, то в контексті даної програми з'явиться повідомлення про помилку. Якщо програми працює некоректно також з'являється повідомлення про помилку.

Для визначення переліку вхідних даних для кожного з випробувальних тестів можна скористатися формами для введення відповідних даних. У кожній з форм визначено перелік необхідних для заповнення полів.

Вхідні дані чи результат при коректних даних повинні відповідати визначеним вимогам до програми у цілому, або до конкретної функції програми, яка тестується. Якщо за результатами роботи підсистеми були згенеровані помилкові вихідні дані – програма повинна видавати повідомлення про помилку.

Додаток Г

Текст програми

У цьому додатку наведено реалізації класів підсистеми, які відповідають за здійснення прогнозування методом клітинних автоматів.

Клас AutomateForecast:

```
package forecast;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;

public class AutomateForecast implements Predict {

    private double getF(double[] z) {
        int p = new RSanalysis().getMemory(z);
        int n = z.length;
        int optK = 0;
        int optM = 0;
        double minEps = Double.MAX_VALUE;
        int m = p;
        while (m < n) {
            int k = 5;
            while (k <= m / 2) {
                Converter c = new Converter(m, k);
                Automate a = new Automate(z, c);
                String s = a.getLingvo();
                double e = 0;
                for (int i = p; i < n; i++) {
                    String conf = s.substring(i - p, i);
                    e += Math.abs(c.toPreciseNumber(a.getPrognose(conf),
                        z[i - 1]) - z[i]);
                }
                if (e < minEps) {
                    optK = k;
                    optM = m;
                    minEps = e;
                }
                k += 3;
            }
            m += 3;
        }
        Converter c = new Converter(optM, optK);
        Automate a = new Automate(z, c);
        String s = a.getLingvo();
        String conf = s.substring(n - p);
        return c.toPreciseNumber(a.getPrognose(conf));
    }

    public Forecast getForecast(TimeSeries t) {
        Forecast f;
        List<Double> y = new ArrayList<Double>();
        List<Quote> q = t.getQoutes();
        for (Quote i : q) {
            y.add(i.getClose());
        }
        double[] a = new double[y.size()];
        int h = 0;
```

```

        for (Double i : y) {
            a[h] = i;
            h++;
        }
        Arrays.sort(a);
        double prog = getF(a);
        return new Forecast(t, new Date(t.getLastDate().getTime() +
            t.getTimeInMs()), "CLOSE", "AUTOMATE", prog);
    }
}

```

Клас RSanalysis:

```

public class RSanalysis {

    private double average(double[] a) {
        double r = 0;
        for (int i = 0; i < a.length; i++) {
            r += a[i];
        }
        return r / a.length;
    }

    private double max(double[] a) {
        double r = Double.MIN_VALUE;
        for (int i = 0; i < a.length; i++) {
            if (r < a[i]) {
                r = a[i];
            }
        }
        return r;
    }

    private double min(double[] a) {
        double r = Double.MAX_VALUE;
        for (int i = 0; i < a.length; i++) {
            if (r > a[i]) {
                r = a[i];
            }
        }
        return r;
    }

    private int max(int[] a) {
        int r = Integer.MIN_VALUE;
        for (int i = 0; i < a.length; i++) {
            if (r < a[i]) {
                r = a[i];
            }
        }
        return r;
    }

    private int min(int[] a) {
        int r = Integer.MAX_VALUE;
        for (int i = 0; i < a.length; i++) {
            if (r > a[i]) {
                r = a[i];
            }
        }
        return r;
    }

    private double S(double[] a) {

```

```

        double r = 0;
        for (int i = 0; i < a.length; i++) {
            r += Math.pow(a[i] - average(a), 2);
        }
        return Math.sqrt(r / a.length);
    }

    public double[][] getRSandH(double[] input) {
        double[][] x;
        int n;
        double[] r;
        double[] s;
        double[] rs;
        double[] h;
        double[][] z;
        double[][] rez;
        n = input.length;
        x = new double[n - 2][];
        r = new double[n - 2];
        s = new double[n - 2];
        rs = new double[n - 2];
        h = new double[n - 2];
        z = new double[n - 2][];
        rez = new double[2][n - 2];
        for (int i = 0; i < n - 2; i++) {
            z[i] = new double[i + 3];
            x[i] = new double[i + 3];
        }
        for (int j = 0; j < n; j++) {
            z[n - 3][j] = input[j];
        }
        s[n - 3] = S(z[n - 3]);
        for (int i = 0; i < n - 3; i++) {
            for (int j = 0; j < z[i].length; j++) {
                z[i][j] = z[n - 3][j];
            }
            s[i] = S(z[i]);
        }
        for (int i = 0; i < x.length; i++) {
            for (int j = 0; j < x[i].length; j++) {
                for (int k = 0; k <= j; k++) {
                    x[i][j] += (z[i][k] - average(z[i]));
                }
            }
            r[i] = max(x[i]) - min(x[i]);
        }
        for (int i = 0; i < rs.length; i++) {
            rs[i] = Math.log10(r[i] / s[i]);
            h[i] = rs[i] / Math.log10(0.5 * (i + 3));
        }
        rez[0] = rs;
        rez[1] = h;
        return rez;
    }

    public int[] getArrayL_r(double[] x) {
        int[] r = new int[x.length];
        int n = 0;
        int[] l_r;
        for (int i = 0; i < x.length; i++) {
            double[] a = new double[x.length - i];
            for (int j = i; j < x.length; j++) {
                a[j - i] = x[j];
            }
        }
    }

```

```

        r[i] = getL_r(a);
        if (r[i] == -1) {
            n = i;
            break;
        }
    }
    l_r = new int[n];
    for (int i = 0; i < n; i++) {
        l_r[i] = r[i];
    }
    return l_r;
}

private int getL_r(double[] x) {
    double[][] rsh = new double[2][x.length - 2];
    rsh = getRSandH(x);
    int r = -1;
    double[] t = new double[x.length - 2];
    for (int i = 0; i < x.length - 2; i++) {
        t[i] = Math.log10(i + 3);
    }
    for (int i = 0; i < x.length - 2; i++) {
        if ((rsh[1][i] > 0.5) && (i != x.length - 3)) {
            if (i != 0) {
                if ((rsh[1][i + 1] < rsh[1][i])
                    && (((rsh[0][i - 1] < rsh[0][i])
                        && (rsh[0][i + 1] < rsh[0][i]))
                    || ((rsh[0][i - 1] > rsh[0][i])
                        && (rsh[0][i + 1] > rsh[0][i])))) {
                    r = i + 3;
                    break;
                }
            }
            } else {
                if ((i + 1 <= x.length - 4) && (rsh[1][i + 1]
                    < rsh[1][i])
                    && (((rsh[0][i] < rsh[0][i + 1])
                        && (rsh[0][i + 2] < rsh[0][i + 1]))
                    || ((rsh[0][i] > rsh[0][i + 1])
                        && (rsh[0][i + 2] > rsh[0][i + 1])))) {
                    r = i + 3;
                    break;
                }
            }
        }
    }
    return r;
}

```