

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

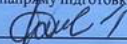
МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія аналізу статистики баскетбольних матчів»

Виконав: студент 2-го курсу, групи 1КН-22м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)



Гульчак Д. С.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

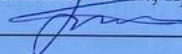


Паночишин Ю.М.

(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., професор каф. КСУ



Биков М. М.

(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

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

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.

29. 08. 2023 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ


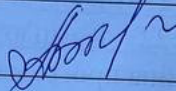
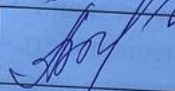
Гульчаку Денису Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія аналізу статистики баскетбольних матчів

керівник роботи к.т.н., доцент кафедри КН Паночишин Ю.М
затверджені наказом вищого навчального закладу від "18" вересня 2023 року № 247
2. Строк подання студентом роботи 13 листопада 2023 року
3. Вихідні дані до роботи:
Вхідні дані – мова програмування об'єктно-орієнтована; кількість факторів – не менше 5; формат файлів – xlsx, тестовий набір – не менше 100 прикладів.
4. Зміст текстової частини:
Вступ, предметна область аналізу статистики баскетбольних матчів, проектування інформаційної технології аналізу статистики баскетбольних матчів, програмна реалізація інформаційної технології аналізу статистики баскетбольних матчів, тестування та аналіз результатів роботи програми аналізу статистики баскетбольних матчів, економічна частина, висновки, спмсок використаних джерел, додатки
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
Загальна структура програми аналізу статистики баскетбольних матчів, Граф-схема загального алгоритму роботи програми аналізу статистики баскетбольних матчів, UML-діаграма класів програмного забезпечення аналізу статистики баскетбольних матчів, результати роботи програми аналізу статистики баскетбольних матчів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Паночишин Ю.М., к.т.н., доц. каф. КН		
5	Адлер О. О., к.т.н., доц. каф. ЕПВМ		

7. Дата видачі завдання 29.08. 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного рівня інформаційних технологій аналізу статистики баскетбольних матчів. Постановка задач дослідження	01.09.23 - 07.09.23	
2	Побудова моделей аналізу статистики баскетбольних матчів на основі нечіткої логіки та функціонування нейронної мережі	08.09.23 - 24.09.23	
3	Практичне застосування та оцінка ефективності розроблених моделей	25.09.23 - 15.10.23	
4	Підготовка економічної частини	16.10.23 - 20.10.23	
5	Апробація та/або впровадження результатів дослідження	21.10.23 - 05.11.23	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	

Студент

Керівник роботи


(підпис)

Гульчак Д. С.


(підпис)

Паночишин Ю.М.

АНОТАЦІЯ

УДК 004.8

Гульчак Д. С. Інформаційна технологія аналізу статистики баскетбольних матчів. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 100 с.

На укр. мові. Бібліогр.: 33 назв; рис.: 20; табл. 7.

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для аналізу статистики баскетбольних матчів. Виконано аналіз сучасних програм-аналогів, які використовуються для аналізу статистики баскетбольних матчів, досліджено методи, які можуть бути використані для реалізації поставленої задачі. Проведено обґрунтування моделі аналізу статистики баскетбольних матчів. Як базовий метод запропоновано застосувати метод нечіткої логіки. Розроблено алгоритм роботи інформаційної технології та відповідне програмне забезпечення на мові програмування Java, в середовищі IntelliJ IDEA. Аналіз роботи програмного забезпечення показав підвищення достовірності прогнозу для аналізу статистики баскетбольних матчів на 4% порівняно з аналогічною програмою.

Графічна частина складається з 6 плакатів.

У економічному розділі визначено, що згідно узагальненого коефіцієнту конкурентоспроможності, науково-технічна розробка переважає існуючі аналоги. Термін окупності становить 0,16 р., що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Ключові слова: прогнозування, нечітка логіка, статистика, баскетбольний матч.

ABSTRACT

Hulchak D. S. Information technology for basketball matches statistics analyzing. Master's thesis in the specialty 122 – «Computer sciences», educational program – «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 100 p.

In Ukrainian language. Bibliogr .: 33 titles; fig . 20; table 7.

This master's thesis is devoted to the development of software for the analysis of statistics of basketball matches. An analysis of modern analog programs used to analyze the statistics of basketball matches was performed, and methods that could be used to implement the given task were investigated. The substantiation of the model for the analysis of statistics of basketball matches was carried out. As a basic method, it is proposed to apply the method of fuzzy logic. The algorithm of the intelligent module and the corresponding software in the Java programming language, in the IntelliJ IDEA environment, were developed. The analysis of the software showed an increase in the reliability of the forecast for the analysis of statistics of basketball matches by 4% compared to a similar program.

The graphic part consists of 6 posters.

In the economic section, it is determined that according to the generalized coefficient of competitiveness, scientific and technical development prevails over existing analogues. The payback period is 0.16 years, which indicates the commercial attractiveness of the scientific and technical development and may encourage a potential investor to finance the implementation of this development and its introduction to the market.

Keywords: prediction, fuzzy logic, statistics, basketball match.

ЗМІСТ

ВСТУП.....	4
1 ПРЕДМЕТНА ОБЛАСТЬ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ.....	8
1.1 Постановка задачі.....	8
1.2 Дослідження факторів впливу на статистику баскетбольних матчів	11
1.3 Огляд відомих методів, що застосовуються для вирішення поставленої задачі.....	14
1.4 Обґрунтування вибору аналогу до програми аналізу статистика баскетбольних матчів.....	17
1.5 Висновок до розділу 1.....	22
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ	23
2.1 Обґрунтування застосування нечіткої логіки для аналізу статистики баскетбольних матчів.....	23
2.2 Модель аналізу статистики баскетбольних матчів із застосуванням нечіткої логіки.....	29
2.3 Розробка структури інформаційної технології аналізу статистики баскетбольних матчів.....	36
2.4 Розробка алгоритму інформаційної технології аналізу статистики баскетбольних матчів.....	38
2.5 Висновок до розділу 2.....	40
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ	41
3.1 Обґрунтування вибору мови та середовища програмування.....	41
3.2 Використання фреймворків та бібліотек.....	48
3.3 Програмна реалізація аналізу статистики баскетбольних матчів	51
3.4 Висновок до розділу 3	54
4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ.....	56
4.1 Тестування програми аналізу статистики баскетбольних матчів	56

4.1 Аналіз результатів роботи програми аналізу статистики баскетбольних матчів	59
4.3 Висновок до розділу 4	60
5 ЕКОНОМІЧНА ЧАСТИНА	62
5.1 Проведення комерційного та технологічного аудиту інформаційної технології аналізу статистики баскетбольних матчів	62
5.2 Розрахунок витрат на здійснення науково-дослідної роботи	63
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	68
5.4 Висновок до розділу 5	73
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	80
Додаток Б (обов'язковий) Лістинг програми	81
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА	91
Додаток Г (довідниковий) Інструкція користувача	98

ВСТУП

Актуальність. Баскетбол є одним з найпопулярніших командних видів спорту. Баскетбольні матчі та турніри цікавлять не тільки вболівальників, а й гравців букмекерських контор. Букмекери зазвичай пропонують вигідні котирування на топові баскетбольні ігри, пропонують багато результатів у лінії та зручні ліміти ставок. Іноді гравці можуть скористатися професійними прогнозами - готовими ставками, зробленими експертами в цьому виді спорту. Десятки готових прогнозів на основі аналізу статистики баскетбольних матчів зараз можна знайти в мережі за пару кліків, але їх якість часто викликає сумніви.

Як і в інших видах спорту, в баскетболі передбачення необхідні для швидкого вибору матчів на лінії та ставок. Новачки можуть використовувати готові прогнози, щоб отримати більше прибутку, а досвідчені гравці можуть порівняти власні прогнози з думкою експертів, щоб зробити певний висновок. Професійні баскетбольні прогнози можуть бути особливо корисними, коли у вас немає достатньо часу, щоб самостійно аналізувати лінії та матчі.

У більшості випадків прогнозування залежить від великої кількості факторів, вплив яких не є постійним. Щоб забезпечити максимально правдивий результат прогнозу необхідно звернутися до великої кількості людських ресурсів. Такі випадки потребують втручання інформаційних технологій, за допомогою яких можна здійснювати тисячі логічних та математичних операцій, при чому з урахуванням всіх деталей та факторів.

Кожен баскетбольний матч несе в собі великий об'єм даних. Їх обробка – важкий та трудомісткий процес. Актуальним завданням є саме автоматизація роботи з цими даними. Інформацію, що несе в собі зіграний матч, можна поділити на такі типи:

- інформація про матч в цілому;
- статистика за гру кожного гравця;
- дані про тренерів граючих команд.

Прогнозування результатів спортивних змагань саме по собі є важливим завданням, складовою основи букмекерського бізнесу. Практичне значення отриманих результатів полягає у можливості використання системи в сфері беттінгу з використанням як ліцензійного так і офіційно безкоштовного програмного забезпечення.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення достовірності аналізу статистики баскетбольних матчів шляхом побудови нечіткої системи прогнозування результатів баскетбольних матчів.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз предметної області аналізу статистики баскетбольних матчів;
- розглянути існуючі методи аналізу статистики баскетбольних матчів та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити математичну модель аналізу статистики баскетбольних матчів;
- розробити структуру та алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об'єкт дослідження – процес комп'ютеризованого аналізу статистики баскетбольних матчів із застосуванням інтелектуальних технологій.

Предмет дослідження – інформаційна технологія та програмні засоби аналізу статистики баскетбольних матчів та достовірність прогнозування.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, теорії нечіткої логіки для реалізації інформаційної технології, методи математичної статистики для розробки процесу розв'язання задачі аналізу статистики баскетбольних матчів та обрахунків результатів експериментів із програмним засобом, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія аналізу статистики баскетбольних матчів, яка відрізняється використанням теорії нечіткої логіки, що дозволило підвищити достовірність програмних засобів аналізу статистики баскетбольних матчів.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення аналізу статистики баскетбольних матчів.

Запропонована інформаційна технологія сприяє підвищенню достовірності програмних засобів аналізу статистики баскетбольних матчів, зокрема:

- розроблено алгоритм роботи програмного забезпечення аналізу статистики баскетбольних матчів;
- розроблено програмні засоби для аналізу статистики баскетбольних матчів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології аналізу статистики баскетбольних матчів. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу отримання статистики баскетбольних матчів та методів підвищення достовірності програмних засобів аналізу статистики баскетбольних матчів [1, 2].

Апробація результатів роботи. Результати роботи були апробовані на Всеукраїнській науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)» (м. Вінниця, Україна, 2023-2024 рр.) [1, 2].

Публікації. За результатами досліджень опубліковано двоє тез доповідей на науково-технічних конференціях [1, 2].

1 ПРЕДМЕТНА ОБЛАСТЬ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ

1.1 Постановка задачі

Якщо відомостей про походження футболу чи тенісу багато, то про походження баскетболу ми знаємо мало. Відомо, що в 1891-1892 роках доктор Джеймс Нейсміт, який викладав у YMCA Training College, створив перші правила баскетболу. У той час Нейсміт прагнув забезпечити фізичну підготовку студентів у міжсезоння – після футбольного сезону та перед бейсболом.

Перші правила баскетболу були занадто простими: два кошика по різні боки майданчика і завдання кинути м'яч в чужий кошик, не даючи суперникам зробити те саме. Оскільки у Нейсмита було 18 учнів, було вирішено грати 9 на 9. Пізніше в правилах було зазначено, що на паркет може вийти будь-яка кількість спортсменів, щоб у командах була рівна кількість гравців. З поширенням баскетболу в США кількість спортсменів у команді стала фіксованою – п'ять гравців. Баскетбол до Європи «привезли» військові США під час Першої світової війни [3]. Минули десятиліття, і баскетбол став чи не найпопулярнішим видом спорту у світі.

Баскетбольна статистика є надзвичайно важливою. Допомагає аналітикам і гравцям дізнатися закономірності, актуальні тенденції, останні тенденції, вивчити всі показники команд і спортсменів, спрогнозувати можливий результат матчу і тотал.

У підсумку, аналізуючи ресурси з статистикою баскетбольних матчів, необхідно порівняти статистичні показники двох команд, дізнатися про поточну форму, наявність мотивації, попередні результати, очні протистояння, а також досконало вивчити склади команд.

Захисникам важливо враховувати підбирання, перехоплення, блокування, роздачі та фоли. Для нападників – виконання ударів з поля,

виконання штрафних кидків, передач, фолів і поразок. Для центрів важливо врахувати все, але найголовніше – точність і оперативність передачі.

Такий аналіз вимагає багато вільного часу, але такий підхід дозволить сформулювати максимально об'єктивний і точний прогноз на баскетбол.

У командних видах спорту ігрове спостереження є дуже важливою зброєю для характеристики індивідуальних і колективних якостей, необхідних для гарної гри. Еволюція методів аналізу ігор дозволила підвищити якість спостережень, а також збільшити кількість спостережуваних факторів і достовірність даних. Серед усіх можливостей аналізу індивідуальної та колективної гри можна виділити фізичні, психологічні, технічні та тактичні аспекти. Всі ці аспекти дуже важливі для розуміння динаміки баскетболу як змагального виду спорту, що складається з технічних елементів (основ), які виконуються з урахуванням обмежень часу, простору та участі товаришів по команді та суперників, що характеризує баскетбол як дуже непередбачуваний і динамічний вид спорту..

На думку Гліка і Маркотта [4], баскетболіст повинен вміти розуміти і виконувати складні рухи та їх варіації, передбачати дії товаришів по команді та суперника, що веде до швидких рішень і високої ефективності у виконанні технічних елементів або показників продуктивності під час гри.

Показниками результативності, за [5], є: кидки (3, 2 і штрафні кидки, виконані та відсоток), набрані очки, можливі очки, ефективність (відношення між набраними і можливими очками), підбирання, передачі, крадіжки, блоковані удари, подачі та фоли.

Одним з найбільш важливих і поширених способів аналізу та оцінки ефективності гравця та команди є статистика гри. Це фактор, який використовується на всіх важливих турнірах у всьому світі, і його використовують тренери та спортсмени для планування сезонів та ігор і визначення процедур для досягнення найкращих результатів.

Статистичні дані з баскетбольної гри показують ефективність гравців та її взаємозв'язок з продуктивністю команди.

Використовуючи цей вид аналізу, можна врахувати багато аспектів. Як приклади:

- Індивідуальні та командні результати;
- Виступ відповідно до конкретних позицій (захисники, нападники та центрові);
- Виступ відповідно до місця гри (домашня та гостьова команда);
- Виступ за результатом гри (команди-переможці та програли).

Одним з найважливіших аспектів є розташування гри. Багато досліджень показують, що це дуже важливий фактор у визначенні кінцевого результату гри в різних командних видах спорту, включаючи баскетбол. Загалом домашні команди виграють близько 65% ігор, а також кращі майже у всіх технічних елементах гри, особливо у відсотках забитих м'ячів, підбирань, передач, крадіжок і передач [6].

У даній роботі необхідно реалізувати інформаційну технологію аналізу статистики баскетбольних матчів.

З допомогою аналізу статистики баскетбольних матчів можна спрогнозувати ймовірність виграшу команди у наступному матчі. Загалом вирішення задачі прогнозування зводиться до вирішення таких підзадач:

- вибір моделі прогнозування;
- аналіз адекватності і достовірності побудованого прогнозу [7].

Для вирішення задачі необхідно розробити опції, якими зможе оперувати користувач. Дана задача передбачає введення користувачем нового матчу, або відкриття існуючого з бази даних. Коли вибрана опція введення нового матчу, відкривається вікно для вводу статистики матчу, після чого можна зберегти її до бази даних, або ж опрацювати введену статистику та вивести ймовірність перемоги.

Коли вибрана опція відкриття існуючого матчу з бази даних, відображується список матчів. Коли користувач обирає конкретний матч, відображається його статистика та кнопка опрацювання і виведення ймовірності перемоги.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм аналізу статистики баскетбольних матчів, що забезпечить покращення аналогічних додатків шляхом підвищення достовірності прогнозування результатів.

Для вирішення проблеми, описаної в даному розділі, необхідно вирішити наступні завдання:

1. Розробити математичну модель;
2. Спроекувати структуру інформаційної технології;
3. Програмно реалізувати інформаційну технологію;
4. Розробити інструкцію користувача;
5. Провести тестування інформаційної технології.

1.2 Дослідження факторів впливу на статистику баскетбольних матчів

Включаючи такі невизначеності, як результативність команд, показники результативності гравців та якість конкурентів, існує множина факторів, які впливають на результат гри. Тому прогнозування результатів гри є досить складною і помітною проблемою дослідження.

Прибуток від ставок на баскетбол можна отримати лише при комплексному аналізі лінії та команд, які зіграють у наступному матчі. Опишемо, на що необхідно звернути увагу при виборі прогнозу.

Мотивація команд та окремих гравців на матч або турнір в цілому. Наприклад, у регулярному сезоні НБА багато провідних клубів не грають на повну, щоб вийти у кращу форму для плей-офф. При цьому в звичайних матчах статус фаворита дуже завищений.

Нинішня форма суперників. У баскетболі часто буває серія з кількох перемог або поразок. Щоб дізнатися більше про форму команд, вивчіть поточну таблицю обраного чемпіонату та зверніть увагу на те, як ви чи інші команди граєте вдома та на виїзді.

Щільність графіка. Ігри НБА поділяються на серії. Часто командам доводиться грати наступного дня після виїзної серії, при цьому долаючи значну дистанцію. Саме завдяки глибині складу і можна припустити, як команда вийде з такого непростого графіка.

Травми та дискваліфікації. Важливо відслідковувати ці фактори в ЗМІ, адже втрата ключового гравця неминуче вплине на загальний результат.

Звичайно, важливо проаналізувати сумісність тактики та стратегії, обраної тренерським штабом. Крім того, на кожному турнірі є команда середнього рівня, з якою важко грати навіть з найкращими клубами. У цьому випадку важливо проаналізувати, чи підходить тактика фаворита, щоб перемогти незграбного аутсайдера.

Дані, отримані для аналізу статистики баскетбольних матчів можна розділити на три типи: дані коробки рахунку (box score), дані відтворення за відтворенням (play-by-play), дані відстеження камери.

Коробка рахунку містить інформацію про гру між двома командами. Приклад коробки рахунку (з NBA.com) містить таблицю, у клітинках для кожного гравця якої записується така статистика:

- Кількість зіграних хвилин (MIN).
- Кількість забитих і спробованих голів з поля (FGMA).
- Кількість 3-очкових м'ячів з поля та спроб (3PMA).
- Кількість виконаних штрафних кидків і спроб (FTMA).
- Оцінка плюс/мінус, яка визначає, як команда виступає з цим гравцем (+/-).
- Кількість підбирань у нападі (OFF).
- Кількість підбирань у захисті (DEF).
- Кількість загальних підборів (TOT).
- Кількість асистів (AST).
- Кількість особистих фолів (PF).
- Кількість крадіжок (ST).
- Кількість оборотів (TO).

- Кількість разів, коли гравець блокував кидок (BS).
- Кількість разів, коли гравець блокував удар (BA).
- Кількість балів (PTS).

Оцінка коробки також містить інформацію на рівні команди. Ми можемо побачити загальну статистику, таку як загальна кількість набраних очок, відсоток забитих м'ячів, 3-очкових голів з гри, штрафних кидків і підбирань команди, які є підбираннями, розподіленими на команду в цілому, а не окремо на одного гравця. Офіційний сайт НБА [8] містить дані про результати з сезону 1946-1947 років до теперішнього часу, хоча дуже рідко використовуються або вимагаються дані, які є надалі, оскільки правила, команди та загальна ліга змінювалися та розвивалися з часом. Іншим популярним джерелом є Basketball Reference [9], який містить дані про результати з сезону 1985-1986 років по теперішній час, хоча і в дещо іншому форматі.

Дані відтворення за відтворенням подібні до даних коробки рахунку, за винятком того, що вони більш детальні. Зокрема, він відстежує кожне відтворення та ставить його часові позначки, щоб мати точне уявлення, коли відбувається кожне відтворення.

За допомогою даних play-by-play можна дізнатися не лише про кількість очок або підбирань у гравця з даних коробки рахунку, але й коли він набрав ці очки або коли він здійснив ці підбирання. Інші речі, які можна дізнатися з даних play-by-play, які не можна отримати з даних про рахунок у боксі, це, наприклад, коли гравці замінюються, коли об'єднуються тайм-аути, хто кому заблокував або хто кому надав асист, тип спроби забитого голу (триочкові, відкладення, данк і т. д.), фоли, які були нічийні, і час зміни володіння.

Найважливішою характеристикою даних play-by-play є те, що вони надають контекст для подій, що відбуваються. Знання гравців на майданчику в різний час дозволяє нам визначити цінність наявності різних комбінацій гравців або цінність, яку надають окремі гравці.

Дані відстеження камерою містять (x, y, z) розташування кожного гравця на майданчику та м'яча відносно фіксованої точки. Ця інформація оновлюється 25 разів на секунду. Станом на травень 2013 року 15 з 30 команд НБА відслідковували ці дані [10], але зараз кожна команда робить це. Ці дані пропонують набагато більш точнішу деталізацію, ніж дані коробки або дані play-by-play. Недоліком використання цих даних є те, що вони стали доступними лише з 2010 року, а до 2013 року лише половина команд НБА відстежувала їх. Тому він неповний і даних для роботи не так багато. Крім того, дані дуже низького рівня; існує необхідність конвертувати дані у формат вищого рівня, щоб зробити з ними щось корисне, а виконання цього перетворення є проблемою та ще одним проектом. В кінці кінців, дані не є загальнодоступними, їх потрібно придбати [11].

1.3 Огляд відомих методів, що застосовуються для вирішення поставленої задачі

Системи прогнозування результату спортивних ігор розробляються багато років. Більшість із них були винайдені для роботи в комерційній сфері, а не в освітніх цілях. Проте в літературі є лише кілька публікацій на цю тему. Більшість із цих систем використовують:

- 1) стохастичні методи для опису невизначеності,
- 2) регресійний та авторегресійний аналіз,
- 3) байєсівські мережі в поєднанні з ланцюгами Маркова та метод Монте-Карло [12],
- 4) методи машинного навчання (також використовувалися, хоча і в меншій мірі, для прогнозування спортивних результатів),
- 5) нечіткі моделі,
- 6) нейронні методи,
- 7) генетичне програмування [13],

8) нечітка модель з поєднанням технологій генетичної та нейронної оптимізації [14].

Нижче наведено короткий опис методів, які використовуються для вирішення цієї проблеми, які поширені в літературі.

Регресія Риджа (RR) була найбільш часто використовуваним алгоритмом регресії для апроксимації відповіді рівняння без єдиного рішення. RR зменшує стандартні помилки, додаючи певний ступінь зміщення до прогнозів регресії [15]. У RR для регуляризації потрібне значення альфа.

Дерева рішень (DT) подібні до регресійних моделей тим, що вони містять кілька незалежних змінних і одну залежну змінну [16]. Ви можете розпізнавати різні функціональні моделі з даних, щоб використовувати їх замість моделей регресії. Алгоритми DT мають тенденцію створювати більш точні моделі з точки зору передбачення та легші для розуміння. Перший вузол, проміжні вузли та останній вузол DT називаються відповідно кореневим вузлом, листовим вузлом і кінцевим вузлом. Використовуючи алгоритм поділу, коефіцієнти в кожному вузлі DT оцінюються з набором навчальних даних. Алгоритм DT будує дерево з нуля, починаючи з питання, яку змінну оцінювати в корені дерева. Номери значень вибраних змінних і конкретний алгоритм використовуються для визначення номерів гілок, які генеруються для кожного вузла.

Штучні нейронні мережі здатні моделювати складні нелінійні функції і відомі як біологічно натхненні математичні методи. У цьому дослідженні використовувалася одна з найбільш широко використовуваних архітектур NN, багат шаровий перцептрон (MLP) з керованим алгоритмом навчання зворотного поширення. MLP може створювати як прогнози регресії, так і моделі класифікації, де вихідна змінна може відрізнитися в залежності від номінальної чи числової ситуації для оцінки та класифікації регресії [17].

Нахил градієнта (GB) утворює адитивний шаблон, який дає вигравш. Це дозволяє оптимізувати випадково диференційовані функції втрат. На кожному кроці дерево регресії розміщується на негативному градієнті цієї

функції втрат. Це точна і практична модель, яку можна використовувати як для задач класифікації, так і для задач регресії. Ітераційний процес експериментування визначає кількість рівнів удосконалення. Збільшення градієнта дуже стійке до перестановок, що означає, що велика кількість зазвичай призводить до кращої продуктивності.

Лінійна регресія (LR) – це лінійний метод моделювання зв'язку між залежною змінною (скалярною реакцією) і незалежною змінною, яка може бути однією або кількома пояснювальними змінними. Проста лінійна регресія включає одну пояснювальну змінну. Однак, якщо існує кілька пояснювальних змінних, процес називається множинною лінійною регресією.

Машина опорних векторів (SVM) – це практичний і конкурентний підхід, який виявляє гіперплощину за допомогою невеликих навчальних даних (допоміжних векторів) на краю розподілів класів і виявляє межі найкращих рішень. Він не вимагає припущень даних, часто мінімізує неправильну класифікацію та уникає переобладнання.

Глибоке навчання стверджує, що моделі складаються з кількох шарів. Глибокі згорткові нейронні мережі включають велику кількість навчальних даних для зближення. Будь-які функції не потрібно заздалегідь визначати, оскільки згорткові шари мережі використовують процес вилучення ознак, а оптимальні структури досягаються під час навчального курсу в глибоких CNN.

Модель нечіткої логіки – це наближення відносини введення-виведення на основі лінгвістичних правил процедур if-then та нечіткої логіки [17]. Щоб створити нечітку модель для завдання прогнозування баскетбольної гри, потрібно виконати такі дії:

- подати вхідні і вихідні змінні у мовній формі;
- формалізувати експертні лінгвістичні судження щодо співвідношення між вхідними та вихідними значеннями в нечіткій базі знань;
- навчити нечітку модель на основі навчального набору шляхом регулювання функцій належності та правил зважування.

Модель на основі нечіткої логіки дає досить точний результат передбачення, тому ви можете використовувати цю модель для вирішення проблеми.

Проведений аналіз запропонованих методів для розв'язання задачі аналізу статистики баскетбольних матчів показує конкурентоспроможність методу на основі нечіткої логіки, який і буде використаний для розробки математичної моделі інформаційної технології.

1.4 Обґрунтування вибору аналогу до програми аналізу статистика баскетбольних матчів

Завдяки однотипному функціоналу кожен сервіс для аналізу баскетбольних матчів має сотні аналогів. Існує безліч різних видів ПЗ для аналізу статистики баскетбольних матчів та для здійснення ставок. У кожного є переваги і недоліки.

В основному програмні продукти для аналізу статистики баскетбольних матчів представлені у вигляді спеціальних сайтів. Розглянемо та проаналізуємо найпопулярніші з них.

«24score» – проект, що висвітлює результати та статистику за декількома видами спорту: футбол, баскетбол, хокей, теніс, гандбол, бейсбол, волейбол (рисунок 1.1) [18].

Сайт 24score – досить просунутий майданчик. Окрім результатів, турнірних таблиць, очних протистоянь та майбутніх матчів, ресурс пропонує статистику за тоталами, половинами, чвертями. На сайті доступні десятки чемпіонатів різних країн, а також Євроліга, НБА, Єдина ліга ВТБ, Чемпіонат Світу тощо.

Також існують мобільні додатки, що забезпечують аналіз статистики баскетбольних матчів. Таким є додаток «NBA: Live Games & Scores» (рисунок 1.2) [19].

24SCORE Обратная связь Полезные ссылки База знаний

🏠 Футбол **Баскетбол** Хоккей Теннис Волейбол Гандбол Серии команд 🇺🇸

Главные Национальные Международные

🇺🇸 США - NBA 🇪🇺 Евролига ULEB 🇬🇷 Греция - HEBA A1 🇪🇸 Испания - ACB Лига
 🇮🇹 Италия - А Лига 🇹🇷 Турция - TBL 🇫🇷 Франция - Лига PRO A 🇧🇪 Лига ВТБ

Баскетбол: расписание и результаты матчей, результаты сегодня, статистика. 21 марта 2022 16:32 GMT+3

Кэффициенты **1хСтавка**

США NBA		1	2
01:10	Атланта Новый Орлеан	112:117 (18:35, 32:25, 26:31, 36:26)	1.515 2.73 📊
01:10	Сакраменто Финикс	124:127 (26:22, 33:27, 28:34, 23:27, 14:17) ОТ	3.96 1.29 📊
01:10	Орlando Оклахома	90:85 (31:25, 14:18, 17:21, 28:21)	1.32 3.72 📊
02:40	Нью-Йорк Юта	93:108 (31:26, 22:34, 20:25, 20:23)	2.464 1.615 📊
03:10	Денвер Бостон	104:124 (27:33, 16:35, 32:30, 29:26)	2.275 1.705 📊
03:40	Голден Стэйт Сан-Антонио	108:110 (22:36, 35:27, 27:20, 24:27)	1.392 3.24 📊
03:40	Филадельфия Торонто	88:93 (37:27, 17:30, 20:12, 14:24)	1.3 3.88 📊

Рисунок 1.1 – Приклад работы сайта «24score»

←

Dallas Mavericks
42 - 30 | 5th in Western
★ FAVORITE

INFO SCHEDULE ROSTER

TOTAL	CONFERENCE	DIVISION
42 - 30 .583 Win%	21 - 21 5th in Western	7 - 5 1st in Southwest

Home: 21-15 Away: 21-15

Last 10: 7-3 Streak: L1

Schedule

Wed. 10/06	vs	UTA	American Airlines Center	8:30 PM	>
Fri. 10/08	vs	LAC	American Airlines Center	8:30 PM	>
Wed. 10/13	at	CHA	Spectrum Center	6:30 PM	>

Рисунок 1.2 – Приклад работы додатку «NBA: Live Games & Scores»

Він дозволяє отримати доступ до офіційного розкладу НБА, результатів, статистики та турнірної таблиці, переглядати основні моменти в грі та після гри, попередньо переглядати та підсумки гри, переглядати останні новини та статті з усієї ліги, отримати прямий доступ до окремих прес-конференцій та подій, переглядати персоналізований контент про ваших улюблених гравців і команди, вибирати редактора для найактуальніших новин та оновлень з усієї ліги. Однак основним недоліком додатку є те, що він не є безкоштовним, що робить його неконкурентноспроможним порівняно з іншими аналогами.

Отже, додаток «NBA: Live Games & Scores» дозволяє щохвилини стежити за останніми результатами, основними моментами, статистикою та новинами.

Наступним розглянемо інструмент для аналізу статистики баскетбольних матчів «FlashScore» - простий і функціональний сайт для відстеження результатів і статистики матчів в лайв-режимі, також є версія додатку для Android (рисунок 1.3).

Величезне охоплення змагань, текстові трансляції, оперативно викладати відео голів, стартові склади задовго до початку гри, позначення кадрових втрат, коефіцієнти від провідних букмекерів, турнірні таблиці, кращі бомбардири, поточна форма команд і інформація по тотал - до цього списку складно щось додати [20].

Сайт FlashScore – мабуть, найпопулярніший сервіс серед шанувальників ставок на спорт, який надає результати і статистику спортивних матчів. На даний момент ресурс висвітлює 32 види спорту для популярних і менш відомих турнірів, ліг, змагань. Свою роботу сервіс почав в 2006 році і за час існування став незаперечним лідером в своєму напрямку.

Баскетбольні результати онлайн - live результати баскетбол, рахунки та результати матчів

FLASHSCORE

ОБРАНЕ 0 ФУТБОЛ БАСКЕТБОЛ ТЕНІС ХОКЕЙ ВОЛЕЙБОЛ ГАНДБОЛ СНУКЕР БІЛЬШЕ

★ МОЇ ЛІГИ

- АКБ
- Ліга А
- НБА
- Суперліга
- ФБУ Суперліга
- ЛНБ
- Євробаскет
- Євроліга
- Єврокубок
- Ліга чемпіонів
- Чемпіонат світу

★ МОЇ КОМАНДИ

+ ДОДАТИ КОМАНДУ

КРАЇНИ

Австралія

Шановні користувачі, засуджуючи російське вторгнення в Україну, ми наразі припиняємо висвітлення російських змагань.

ВСІ LIVE КОЕФІЦІЄНТИ ЗАВЕРШЕНІ ЗАПЛАНОВАНІ

27/04 CP

★ ЄВРОПА: Євроліга · Плей-оф

☆	21:30	Баварія	-	ⓘ	💬
		Барселона	-		
☆	22:00	Монако	-	ⓘ	💬
		Олімпіакос	-		

★ ЄВРОПА: Єврокубок · Плей-оф

☆	21:00	Цедевіта Олімпія	-		
		Бурсаспор	-		
☆	21:30	Валенсія	-		
		Леваллуа	-		

Рисунок 1.3 – Приклад роботи програми «FlashScore»

Стартова сторінка ресурсу зустрічає користувачів списком баскетбольних матчів, які проходять в сьогоднішній день. Після реєстрації на сайті, користувач може додати необхідні ліги в обраний список, який буде публікуватися в першочерговому порядку.

З лівого боку головної сторінки, представлені найпопулярніші європейські баскетбольні ліги і міжнародні змагання. Нижче розташовується список турнірів в алфавітному порядку. В цілому, футбол є профільним видом спорту для даного ресурсу. Однак окрім нього розглядається ще 34 види спорту, в тому числі і баскетбол.

Для більшості матчів FlashScore веде онлайн статистику, у поєдинках відомих турнірів ведеться текстова онлайн трансляція на різних мовах. Повноцінна версія сайту також публікує відео-повтори забитих голів.

Наступним розглянемо додаток для Android – «Basketball total predictor» – програма для розрахунку тотала чверті у баскетбольних матчах [20].

Програма робить прогноз на тотал результат матчу чверті баскетболу на основі обчислень з використанням нейромереж; прогноз для спортивних подій; прогноз на результати спортивної події; прогноз на кінець чверті; прогноз тоталу чверті. Допомагає дізнатися тотал чверті баскетбол, прогноз результату в режимі он-лайн, прогноз чверті лайв (рисунок 1.4).

НБА	10'										Лайв		
Очки команда1	20	18	19	20	21	22	23	24	25	26	27	28	29
Очки команда2	28	18	19	20	21	22	23	24	25	26	27	28	29
Время четверти	09	00											
РЕЗУЛЬТАТ	прогноз тотала = 61												
РАССЧИТАТЬ ПРОГНОЗ НА ТОТАЛ													

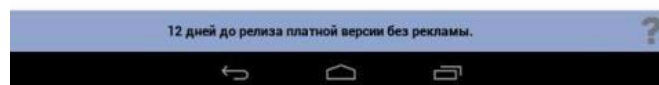


Рисунок 1.4 – Приклад роботи програми «Basketball total predictor»

Даний додаток є одним з небагатьох, здатних не лише отримувати і аналізувати статистику, а й прогнозувати результати баскетбольних матчів. Проте вагомим недоліком програми є те, що вона досить застаріла – останнє оновлення її було 2016 року, а також вона має досить низький рейтинг – лише три зірки, що вказує на недостатню достовірність прогнозування та незадоволеність користувачами даною програмою.

Отже, більшість з розглянутих інструментів для аналізу спортивних подій – платні з можливістю тимчасового і часткового безкоштовного використання. Перед тим, як платити гроші за програмне забезпечення, слід ознайомитися з функціоналом і можливостями. Варто зазначити, що більшість програмного забезпечення не містить прогнозування, а лише приймає ставки на матчі. Інші ж програми містять лише статистичний аналіз баскетбольних матчів. Єдиною програмою, здатною прогнозувати результати баскетбольних матчів на основі статистики – додаток «Basketball total predictor», однак зважаючи на описані недоліки, існує потреба у вдосконаленні додатку за рахунок використання більшої кількості факторів, що впливають на прогноз за допомогою методу на основі нечіткої логіки.

1.5 Висновок до розділу 1

В даному розділі було проведено аналіз сучасних програм-аналогів, які використовуються для задач аналізу статистики баскетбольних матчів, а також інших спортивних подій. Було наведено короткий опис основних функцій, які виконують дані програми, що виявило їх нездатність до точного прогнозування результатів баскетбольних матчів. Також досліджено основні фактори впливу на статистику та результати баскетбольних матчів. Було досліджено методи, що можуть бути використані для аналізу статистики баскетбольних матчів, для чого запропоновано використати методи нечіткої логіки. Здійснено постановку задачі.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ

2.1 Обґрунтування застосування нечіткої логіки для аналізу статистики баскетбольних матчів

У світі все взаємопов'язане. Знайти цей зв'язок означає робити точні прогнози і приймати правильні рішення. Відповідно до знаменитої теореми FAT (Fuzzy Approximation Theorem), доведеної Коско, будь-яка математична система може бути апроксимована системою, яка базується на нечіткій логіці. Зазвичай система прогнозування будується на основі таких двох функцій [21]:

- моделі прогнозування, що описується за допомогою системи нечітких відносин,
- функції навчання, що дає послідовні зміни до коефіцієнтів цієї моделі. Функція навчання необхідна для підтримки високої точності прогнозування.

Правила нечіткого виведення.

У 1973 році Лотфі Заде виклав новий підхід до аналізу складних систем, у якому Заде запропонував описати процедуру людського висновку у нечітких правилах.

Нечітке правило може бути визначено як умовний вираз у формі:

ЯКЩО (x) дорівнює (A) ТОДІ (y) дорівнює (B) ,

де (x) і (y) - лінгвістичні змінні, (A) і (B) - лінгвістичні величини, визначені нечіткими множинами на універсумах (X) і (Y) відповідно.

У загальному випадку міркування якщо тоді включає дві окремі частини: оцінка правила (якщо частина правила) і застосування результату до наступного (тоді частина правила). Класичне правило якщо використовує двійкову логіку, тобто, якщо перша частина вірна, вірна і друга і навпаки. У

нечітких системах, всі правила певною мірою спрацьовують, іншими словами вони спрацьовують частково.

У разі вірності з деяким ступенем умови ЯКЩО, отримуємо, що з тим самим ступенем це буде вірно і для умови ТОДІ.

Результатом кожного правила є нечітка множина, але зазвичай нам потрібно отримати однину. Іншими словами, ми хочемо отримати точне рішення, а не нечітке. Щоб отримати одне чітке рішення для вихідної змінної, нечітка експертна система спочатку агрегує всі вихідні нечіткі множини в одну вихідну нечітку множину, а потім збирає отриманий нечіткий набір в одне число.

Найчастіше використовуваним методом нечіткого висновку є так званий метод Мамдані. У 1975 році професор Лондонського університету Ебрахім Мамдані побудував одну з перших нечітких систем для управління комбінацією парових двигунів і котлів.

Процес нечіткого виведення в стилі Мамдані виконується в чотири етапи: фазифікація вхідних змінних, оцінка правил, агрегація правил виведення і, нарешті, дефазифікація, тобто приведення до чіткості.

Алгоритми нечіткого виведення.

Нехай (x, y) – вхідні змінні, що мають чіткі значення (x_0, y_0) та (z) – вихідна змінна.

Задано функції приналежності $(\mu_{A_1}, \mu_{A_2}, \mu_{B_1}, \mu_{B_2}, \mu_{C_1}, \mu_{C_2})$ та правила:

Якщо $(x) \in (A_1)$ і $(y) \in (B_1)$, ТО $(z) \in (C_1)$,

Якщо $(x) \in (A_2)$ і $(y) \in (B_2)$, то $(z) \in (C_2)$.

Алгоритм Mamdani.

У системах типу Mamdani база знань будується з нечітких висловлювань виду « $z \in \alpha$ » за допомогою зв'язок «І», «ЯКЩО ТО» (рис. 2.1).

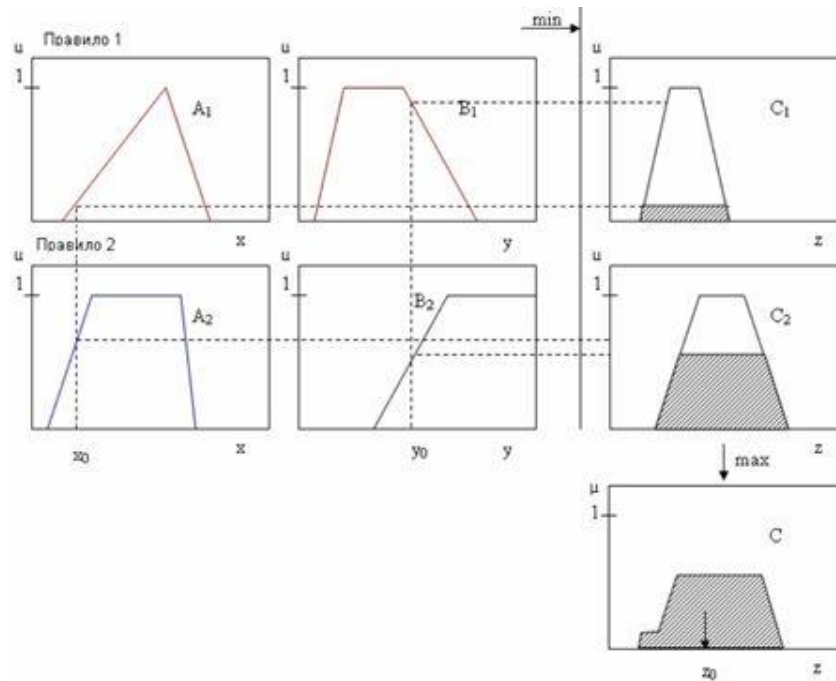


Рисунок 2.1 – Алгоритм Mamdani

Етапи нечіткого висновку реалізуються так:

1) Фаззифікація: знаходяться ступеня істинності для передумов кожного правила: $(\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{B_1}(x_0), \mu_{B_2}(x_0))$.

2) Висновок: знаходяться рівні відсікання для передумов кожного з правил з використанням операції мінімум: $[\alpha_1 = \min\{\mu_{A_1}(x_0), \mu_{B_1}(y_0)\}, \alpha_2 = \min\{\mu_{A_2}(x_0), \mu_{B_2}(y_0)\}]$. Потім знаходяться усічені функції приналежності: $[\mu_{C_1'}(z) = \min\{\alpha_1, \mu_{C_1}(x_0)\}, \mu_{C_2'}(z) = \min\{\alpha_2, \mu_{C_2}(x_0)\}]$.

3) Композиція: з використанням операції максимум проводиться об'єднання знайдених усічених функцій, що призводить до отримання підсумкової нечіткої підмножини для змінної виходу з функцією приналежності: $[\mu_{\Sigma} = \mu_C(z) = \max\{\mu_{C_1'}, \mu_{C_2'}\} = \max\{\min\{\alpha_1, \mu_{C_1}(z)\}, \min\{\alpha_2, \mu_{C_2}(z)\}\}]$.

4) Приведення до чіткості для отримання (z_0) проводиться з використанням центру тяжіння.
$$\left[z_0 = \frac{\int_{\text{Min}}^{\text{Max}} z \cdot \mu_{\Sigma}(z) dz}{\int_{\text{Min}}^{\text{Max}} \mu_{\Sigma}(z) dz}, \right]$$
 де (Min) і (Max) - межі універсуму нечітких змінних.

Алгоритм Tsukamoto.

Вихідні дані та база знань такі ж, як і в алгоритмі Mamdani, але передбачається, що функції $(\mu_{C_1}(z), \mu_{C_2}(z))$ є монотонними.

Етапи нечіткого висновку (рис. 2.2):

1. Фаззифікація: знаходяться ступеня істинності для передумов кожного правила: $(\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{B_1}(y_0), \mu_{B_2}(y_0))$.

2. Висновок: Знаходяться рівні відсікання для передумов кожного з правил з використанням операції мінімум:
$$\left[\alpha_1 = \min\{\mu_{A_1}(x_0), \mu_{B_1}(y_0)\}, \alpha_2 = \min\{\mu_{A_2}(x_0), \mu_{B_2}(y_0)\}. \right]$$

3. Потім знаходяться чіткі значення z_1 і z_2 з рівнянь $(\alpha_1 = \mu_{C_1}(z), \alpha_2 = \mu_{C_2}(z))$.

4. Визначається чітке значення змінної висновку, як зважене середнє (z_1) і (z_2)
$$\left[z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}. \right]$$

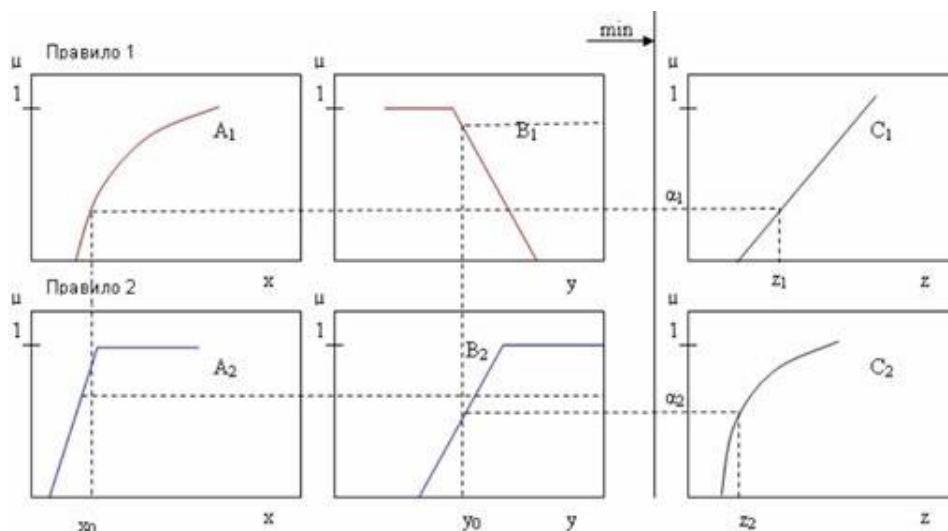


Рисунок 2.2 – Алгоритм Tsukamoto

У загальному випадку чітке значення (z_0) визначається з використанням центру тяжіння.

Алгоритм Sugeno (рис. 2.3).

В алгоритмі Sugeno база знань будується з правил у такій формі:

Якщо $(x) \in (A_1)$ і $(y) \in (B_1)$, ТО $(z_1 = a_1x + b_1y)$,

Якщо $(x) \in (A_2)$ і $(y) \in (B_2)$, то $(z_2 = a_2x + b_2y)$.

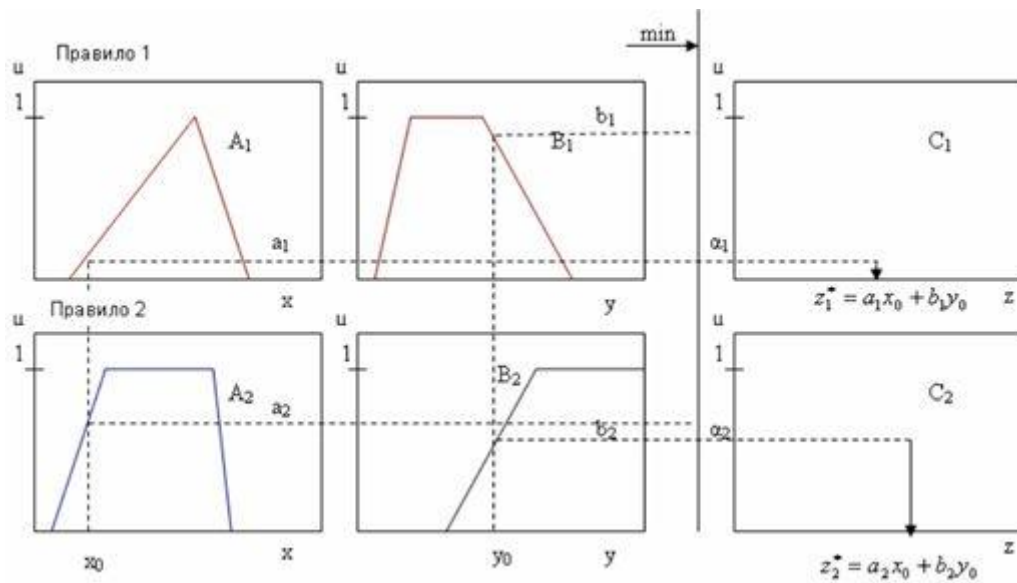


Рисунок 2.3 – Алгоритм Sugeno

Етапи нечіткого виведення:

1) Фаззифікація: знаходяться ступеня істинності для передумов кожного правила: $(\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{B_1}(y_0), \mu_{B_2}(y_0))$.

2) Висновок: Знаходяться рівні відсікання для передумов кожного з правил з використанням операції мінімум: $(\alpha_1 = \min\{\mu_{A_1}(x_0), \mu_{B_1}(y_0)\})$, $(\alpha_2 = \min\{\mu_{A_2}(x_0), \mu_{B_2}(y_0)\})$.

3) Знаходяться індивідуальні виходи правил: $(z_1^* = a_1x + b_1y)$, $(z_2^* = a_2x + b_2y)$.

4) Визначається чітке значення змінної висновку:

$$z_0 = \frac{\alpha_1 z^*_1 + \alpha_2 z^*_2}{\alpha_1 + \alpha_2}.$$

Алгоритм Larsen (рис. 2.4).

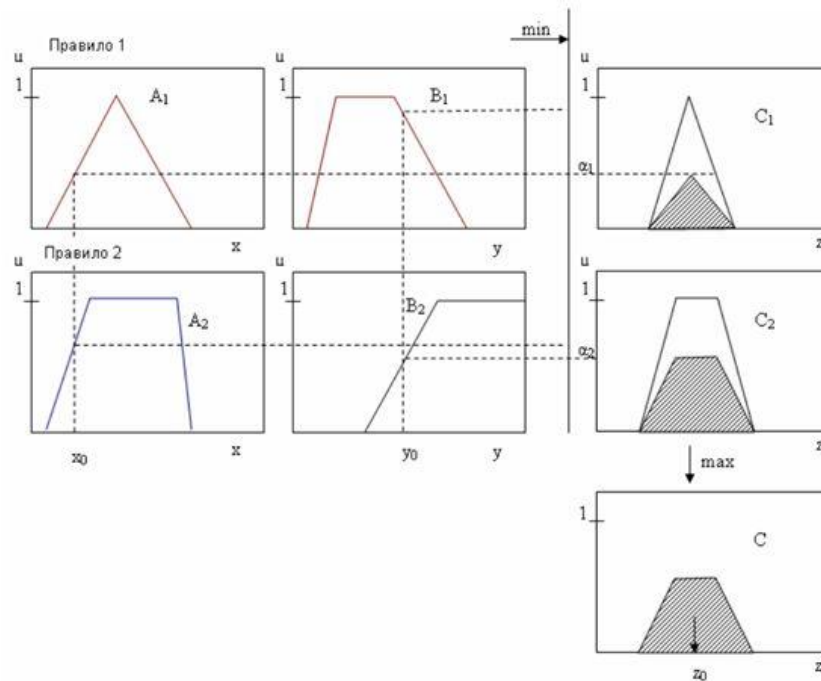


Рисунок 2.4 – Алгоритм Larsen

Вид бази знань збігається з видом бази знань алгоритму Mamdani.

1) Нечіткість: знаходяться ступеня істинності для передумов кожного правила: $(\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{B_1}(y_0), \mu_{B_2}(y_0))$.

2) Висновок: знаходяться рівні відсікання для передумов кожного з правил з використанням операції мінімум:
 $(\alpha_1 = \min\{\mu_{A_1}(x_0), \mu_{B_1}(y_0)\}), \quad (\alpha_2 = \min\{\mu_{A_2}(x_0), \mu_{B_2}(y_0)\})$.

3) В алгоритмі Larsen нечітке підмножина змінної виводу для кожного правила знаходиться з використанням оператора множення за формулою: $(\mu_{C_1'}(z) = \alpha_1 \cdot \mu_{C_1}(z), \mu_{C_2'}(z) = \alpha_2 \cdot \mu_{C_2}(z))$.

4) Композиція: з використанням операції максимум проводиться об'єднання знайдених приватних нечітких підмножин. Знаходиться підсумкове нечітке підмножина для змінної виходу з функцією приналежності:

$$\mu_{\Sigma}(z) = \mu_C(z) = \max\{\mu_{C_1}(z), \mu_{C_2}(z)\} = \max\{\alpha_1 \cdot \mu_{C_1}(z), \alpha_2 \cdot \mu_{C_2}(z)\}.$$

Приведення до чіткості також провадиться методом центру тяжіння.

Нечітку теорію у цій роботі варто використовувати для аналізу важливості ігрових параметрів, виділення більш важливих параметрів і прогнозування результату гри.

2.2 Модель аналізу статистики баскетбольних матчів із застосуванням нечіткої логіки

Спортивний ринок швидко зростає протягом останніх кількох десятиліть із розвитком технологій, мовлення, преси та соціальних медіа. Прогнозування спортивних результатів має вирішальне значення на багатьох спортивних ринках, таких як ставки на спорт, управління клубами та операції, а також керування трансляціями, оскільки точне прогнозування спортивних результатів забезпечує точну довідку про ставки, інформацію про управління та операції, а також підвищує інтерес глядачів. Наприклад, у спортивних лотереях буде більше інтересу до прогнозування результатів, оскільки основні ставки зосереджені на рахунках, таких як очки розповсюдження, гандикапи, правильні рахунки або загальна кількість більше/менш голів. Тому розробка ефективної моделі прогнозування спортивних результатів, яка може досягати точних і надійних результатів прогнозування, є однією з важливих і привабливих завдань спортивної аналітики.

Оскільки багато баскетбольних ігор розглядаються з огляду на результати, підбирання тощо, неможливо встановити, який ігровий параметр є більш важливим у змаганнях. Запропоновано систему прогнозування

баскетбольної гри на основі нечіткої теорії. Його механізм прогнозування полягає в тому, щоб фіксувати параметри гри протягом певного періоду часу в минулому та спостерігати за їх змінами як основу для прогнозування гри. Нечітка теорія буде використовуватися для аналізу важливості ігрових параметрів, виділення більш важливих параметрів і прогнозування результату гри. Більш важливі параметри гри будуть визначені за допомогою нечіткої теорії та порівняні з іншими статистичними методами для аналізу точності передбачення вигідних параметрів гри. Особливо важливими є результати експерименту, визначені параметрами гри, і запропонований метод дозволяє отримати більш високу достовірність прогнозування гри.

Для формального визначення нечіткої множини відсутні обмеження на вибір деякої функції належності з метою її представлення. Проте практика показує, що зручніше застосувати інструменти, які роблять можливим аналітичне представлення в якості простої математичної функції. Така технологія не лише робить простішими відповідні числові розрахунки, але й забезпечить зменшення обчислювальних ресурсів, необхідних для збереження окремих значень даних функцій. Потреба в характеристиках окремих функцій належності передбачається наявністю відповідних перетворень функцій у наступних інструментах [22].

Представлення нечіткої моделі полягає в апроксимації залежностей «вхід – вихід», що базується на лінгвістичних правилах типу «ЯКЩО – ТО», а також операціях нечіткого логічного виведення. Розглянемо блоки, які містяться в нечіткій моделі [23]:

- фазифікатор, який перетворює визначений вектор факторів (X), що мають вплив на вектор нечітких множин \bar{X} , потрібних для проведення нечіткого логічного висновку;

- нечітка база знань, що несе в собі дані про залежності $Y = f_1 X_1$, які мають вигляд лінгвістичних правил типу «ЯКЩО – ТО»;

- машина нечіткого логічного висновку, створена на основі правил нечіткої бази знань, що обраховує значення вихідної змінної нечіткої множини \bar{Y} , що відповідає нечітким значенням вхідних змінних (\bar{X});

- дефазифікатор, який перетворює вихідну нечітку множину \bar{Y} в чітке число.

Прогноз, що базується на нечіткій логіці, має за мету опис причинно-наслідкових взаємозв'язків між факторами ризику (причинами), які мають вплив на результат аналізованого баскетбольного матчу, та конкретним результатом прогнозування (наслідком) що виражається твердженням природною мовою, що означає надання можливості висловити думки експертів логічною формалізацією. Наприклад, в даному випадку організацію причинно-наслідкових зв'язків, які встановлюються, можна описати як наступну систему відносин (1-4):

1. $d = fd(y, w, \dots z)$;
2. $y = fy(x_1, x_2, \dots x_{10})$;
3. $w = fw(x_{11}, x_{12}, \dots x_{20})$;
4. $z = fz(x_{21}, x_{22}, \dots x_n)$;

де d – прогноз результату певного баскетбольного матчу,

$y, w, z, x_1 \dots x_n$ – визначені фактори, які описують статистичні дані щодо майбутнього баскетбольного матчу [24].

Аналіз статистики конкретного баскетбольного матчу відбувається за наступним алгоритмом прогнозування результату матчу[25]:

Крок 1. Визначити значення впливових факторів для даного матчу.

Отже, на першому кроці алгоритму відібрано наступні фактори:

x_1 – різниця у кількості підборів за гру;

x_2 – різниця у кількості передач за гру;

x_3 – різниця у кількості перехоплень за гру;

x_4 – різниця у кількості блокшитів за гру;

x_5 – різниця у кількості втрат за гру;

x_6 – різниця відсотку реалізації кидків з гри;

x_7 – різниця відсотку реалізації 3-х очкових кидків;

x_8 – різниця відсотку реалізації штрафних кидків.

Ці дані визначаються на основі статистики останніх п'яти турів.

Для лінгвістичної оцінки вхідних і вихідної змінних будемо використовувати терм-множини, наведені в табл. 2.1.

Крок 2. Обчислення ступеню приналежності значень визначених факторів впливу на нечіткі терми.

Крок 3. Підстановка знайдених на другому кроці значень в систему нечітких логічних рівнянь та визначення ступеню належності отриманого рішення до встановлених нечітких термів.

Крок 4. Визначення результируючих значень нечіткої множини, об'єднуючи відповідні нечіткі множини.

Крок 5. Проведення дефазифікації отриманої на четвертому кроці нечіткої множини, застосувавши метод центру тяжіння.

$$y = \frac{\int_{-3}^3 y \mu_{\bar{y}}(y) dy}{\int_{-3}^3 \mu_{\bar{y}}(y) dy} \quad (2.1)$$

Крок 6. Визначення результату баскетбольного матчу на основі знайденої різниці вхідних даних:

$$D = \begin{cases} d_1, \text{ if } y \in [0.5, 3] \\ d_2, \text{ if } y \in [-0.5, 0.5] \\ d_3, \text{ if } y \in [-3, -0.5] \end{cases} \quad (2.2)$$

Така система має на меті встановлення багатофакторної залежності, завдяки використанню результатів логічного висновку з нечіткої бази знань.

Таблиця 2.1 – Терм-множини вхідних і вихідних змінних

Змінна	Терм-множина	Параметри функції належності	
		c	b
x1	Програш (П1)	2.55	-6
	Рівність (Р1)	2.55	0
	Виграш (В1)	2.55	6
x2	Програш (П2)	4.25	-35
	Рівність (Р2)	4.25	0
	Виграш (В2)	4.25	35
x3	Програш (П3)	2.76	13
	Рівність (Р3)	2.76	0
	Виграш (В3)	2.76	-13
x4	Програш (П4)	0.7	8
	Рівність (Р4)	0.7	0
	Виграш (В4)	0.7	-8
x5	Програш (П5)	8.5	20
	Рівність (Р5)	8.5	0
	Виграш (В5)	8.5	-20
x6	Програш (П6)	5.5	20
	Рівність (Р6)	5.5	10
	Виграш (В6)	5.5	0
x7	Програш (П7)	4.25	8
	Рівність (Р7)	4.25	0
	Виграш (В7)	4.25	-8
x8	Програш (П8)	4.25	8
	Рівність (Р8)	4.25	0
	Виграш (В8)	4.25	-8
y	Значний програш (ЗП)	0.64	-3
	Програш (П)	0.44	-0.9
	Нічия (Н)	0.44	0
	Виграш (В)	0.44	0.9
	Значний виграш (ЗВ)	0.64	3

Створення і настройка нечіткої бази знань виступає однією з найголовніших задач під час розробки нечіткої системи.

Підвищення достовірності прогнозування для оцінювання вихідної змінної передбачає застосування п'яти лінгвістичних термів, таких як «Значний програш», «Програш», «Нічия», «Виграш», «Значний виграш», що застосовують на визначених терм-множинах між двома командами. У цьому випадку:

$d_1 = \text{Виграш} \cup \text{Значний виграш};$

$d_2 = \text{Нічия};$

$d_3 = \text{Програш} \cup \text{Значний програш}.$

За формалізацію лінгвістичних термів відповідає гаусівська функція приналежності:

$$\mu^t(x) = e^{-\frac{|x-b|^2}{2c^2}} \quad (2.3)$$

де $\mu^t(x)$ – функція приналежності змінної x до терму t ; b – параметр функції приналежності, відповідний координаті максимуму ($\mu^t(x) = 1$); c – параметр стиснення-розтягування функції приналежності [25].

Параметри функцій належності наведені в табл. 2.1 для кожного лінгвістичного терму вибиралися експертно. Нечіткий логічний висновок аналізу статистики баскетбольного матчу здійснюється за наступною системою нечітких логічних висловлювань:

$$\begin{aligned} \mu_{3B}(X) = & \mu_{B1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{B5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{B7}(x_7) \wedge \mu_{B8}(x_8) \\ & \vee \mu_{B1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{P8}(x_8) \\ & \vee \mu_{P1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{B8}(x_8) \\ & \vee \mu_{P1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{P8}(x_8) \\ \mu_B(X) = & \mu_{B1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{B5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{B7}(x_7) \wedge \mu_{B8}(x_8) \vee \\ & \mu_{B1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{P8}(x_8) \vee \end{aligned}$$

$$\begin{aligned} & \wedge \mu_{P1}(x_1) \wedge \mu_{P2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{P4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{B7}(x_7) \wedge \mu_{B8}(x_8) \vee \\ & \wedge \mu_{П1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{P8}(x_8) \end{aligned} \quad (2.4)$$

$$\mu_H(X) =$$

$$\begin{aligned} & \mu_{P1}(x_1) \wedge \mu_{P2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{P4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{P6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{P8}(x_8) \\ & \vee \wedge \mu_{B1}(x_1) \wedge \mu_{P2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{P4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{B8}(x_8) \\ & \vee \wedge \mu_{B1}(x_1) \wedge \mu_{П2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{П4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{B6}(x_6) \wedge \mu_{П7}(x_7) \wedge \mu_{B8}(x_8) \\ & \vee \wedge \mu_{П1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{B4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{П6}(x_6) \wedge \mu_{B7}(x_7) \wedge \mu_{П8}(x_8) \end{aligned}$$

$$\mu_{П}(X) =$$

$$\begin{aligned} & \mu_{П1}(x_1) \wedge \mu_{П2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{П4}(x_4) \wedge \mu_{П5}(x_5) \wedge \mu_{П6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{П8}(x_8) \\ & \vee \wedge \mu_{П1}(x_1) \wedge \mu_{P2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{П4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{П6}(x_6) \wedge \mu_{B7}(x_7) \wedge \mu_{П8}(x_8) \\ & \vee \wedge \mu_{B1}(x_1) \wedge \mu_{П2}(x_2) \wedge \mu_{П3}(x_3) \wedge \mu_{P4}(x_4) \wedge \mu_{B5}(x_5) \wedge \mu_{P6}(x_6) \wedge \mu_{П7}(x_7) \wedge \mu_{P8}(x_8) \\ & \vee \wedge \mu_{B1}(x_1) \wedge \mu_{П2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{П4}(x_4) \wedge \mu_{П5}(x_5) \wedge \mu_{P6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{B8}(x_8) \end{aligned}$$

$$\mu_{3П}(X) =$$

$$\begin{aligned} & \mu_{П1}(x_1) \wedge \mu_{П2}(x_2) \wedge \mu_{П3}(x_3) \wedge \mu_{П4}(x_4) \wedge \mu_{П5}(x_5) \wedge \mu_{П6}(x_6) \wedge \mu_{П7}(x_7) \wedge \mu_{П8}(x_8) \\ & \vee \wedge \mu_{П1}(x_1) \wedge \mu_{B2}(x_2) \wedge \mu_{P3}(x_3) \wedge \mu_{P4}(x_4) \wedge \mu_{P5}(x_5) \wedge \mu_{П6}(x_6) \wedge \mu_{П7}(x_7) \wedge \mu_{П8}(x_8) \\ & \vee \wedge \mu_{B1}(x_1) \wedge \mu_{П2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{P4}(x_4) \wedge \mu_{П5}(x_5) \wedge \mu_{П6}(x_6) \wedge \mu_{P7}(x_7) \wedge \mu_{P8}(x_8) \\ & \vee \wedge \mu_{P1}(x_1) \wedge \mu_{P2}(x_2) \wedge \mu_{B3}(x_3) \wedge \mu_{П4}(x_4) \wedge \mu_{П5}(x_5) \wedge \mu_{P6}(x_6) \wedge \mu_{П7}(x_7) \wedge \mu_{P8}(x_8) \end{aligned}$$

де $\mu_A(x)$ – ступінь приналежності значення змінної до лінгвістичного терму.

Можна зробити висновок, що кожне правило бази знань є висловлюванням «ЯКЩО-ТО». Правила, що на вході мають однаковий параметр, об'єднуються в рядках таблиці за допомогою операції нечіткого логічного висновку «АБО». Вага правила w показує суб'єктивну впевненість експерта стосовно розглянутого правила. Етап формування структури нечіткої вагової моделі передбачає визначення для всіх правил бази знань, рівних одиниці [26.27]. Реалізація нечіткого логічного висновку передбачає перехід від логічних висловлювань до рівнянь нечіткої логіки. При нечіткому виведенні паралельно складається велика кількість правил для подальшого їх об'єднанням в остаточний прогноз. Правила можуть мати за основу досвід та знання експертів, створювати шаблон дій оператора, метод навчання. Під час

проектування системи на основі нечіткої логіки слід переконатися, у їх адаптації до змін навколишнього середовища, беручи до уваги правила на базі експериментальних даних. Навчання складається з адаптивного вибору параметрів нечіткого набору та автоматичного генерування правил нечіткого висновку.

2.3 Розробка структури інформаційної технології аналізу статистики баскетбольних матчів

Задача аналізу статистики баскетбольних матчів в даному випадку полягає у прогнозуванні найбільш імовірного результату баскетбольного матчу, що повинно характеризуватися максимально можливою достовірністю.

Розроблена загальна архітектура інформаційної технології аналізу статистики баскетбольних матчів зображена на рисунку 2.5.

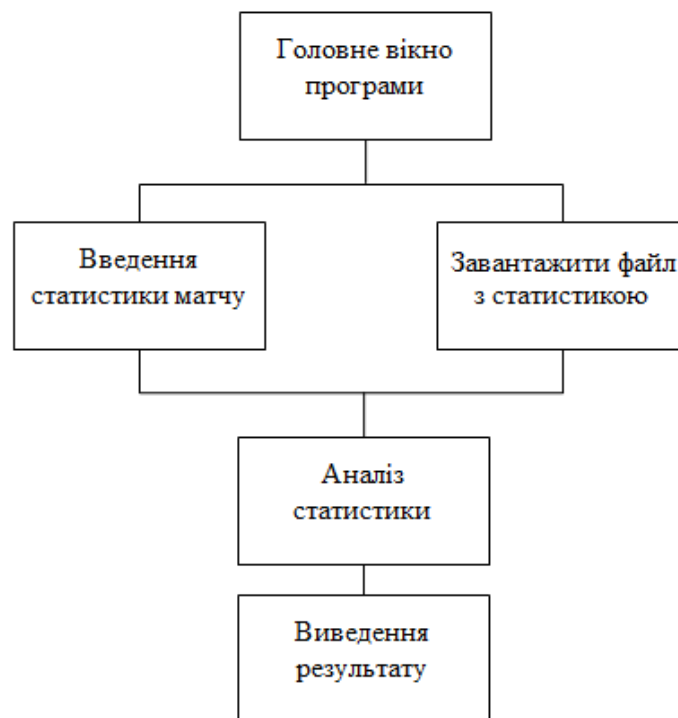


Рисунок 2.5 – Загальна структура інформаційної технології аналізу статистики баскетбольних матчів

Інформаційна технологія забезпечує введення статистики баскетбольних матчів безпосередньо в вікно програми у спеціальні поля та завантаження файлу з статистикою.

В програмі будуть представлені модуль інтерфейсу, модуль обробки статистики, та модуль прогнозування. Структура функціонування програмного забезпечення аналізу статистики баскетбольних матчів наведена на рисунку 2.6.

Завдяки графічному інтерфейсу, що пов'язаний з модулями обробки статистики та прогнозування користувач взаємодіє з програмою для отримання всієї необхідної йому інформації про статистику баскетбольного матчу, а також може отримати результат у вигляді прогнозу на матч, що його цікавить. Він відповідає за коректне відображення на екрані результату прогнозування, а також за введення вручну статистики певного матчу користувачем, якщо немає необхідної інформації в базі даних.

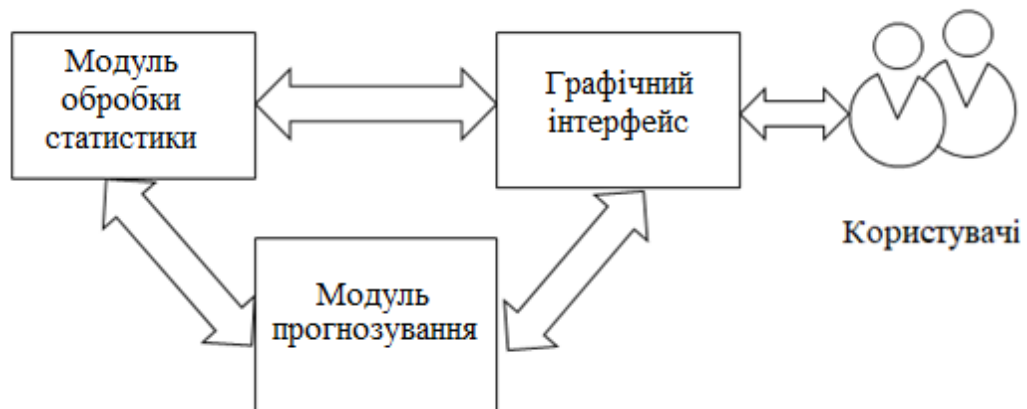


Рисунок 2.6 – Загальна структура програмного забезпечення аналізу статистики баскетбольних матчів

Модуль роботи із статистикою баскетбольних матчів пов'язаний з модулем прогнозування та модулем інтерфейсу. Введена за допомогою графічного інтерфейсу статистика перевіряється на коректність даних та

можливість їх опрацювання. Якщо дані коректні – відбувається прогнозування у наступному модулі.

Модуль прогнозування пов'язаний з модулем обробки статистики. Цей модуль є головним модулем програми та містить інтелектуальний алгоритм аналізу статистики баскетбольних матчів, що базується на нечіткій логіці.

2.4 Розробка алгоритму інформаційної технології аналізу статистики баскетбольних матчів

Розглянемо загальний алгоритм роботи додатку, що реалізує інформаційну технологію аналізу статистики баскетбольних матчів (рис. 2.7).

Даний алгоритм складається з наступних кроків:

Крок 1. На першому етапі користувач за допомогою графічного інтерфейсу входить до системи, щоб провести аналіз статистики баскетбольних матчів. На цьому етапі відбувається виведення головного меню програми.

Крок 2. Вибір режиму роботи.

Крок 3. При необхідності введення статистики вручну виводиться спеціальна форма з полями для введення відповідної статистики.

Крок 4. Відбувається формування сутностей з даних, введених користувачем. Перехід до кроку 8.

Крок 5. При вибраному режимі роботи з файлом відбувається завантаження через спеціальну форму документа з статистикою.

Крок 6. Обробка завантаженого файлу.

Крок 7. Формування даних.

Крок 8. Проведення валідації даних.

Крок 9. Перевірка заповненості полів та валідності даних. Якщо дані не валідні – перехід до кроку 13.

Крок 10. При успішній валідації даних проводиться їх аналіз.

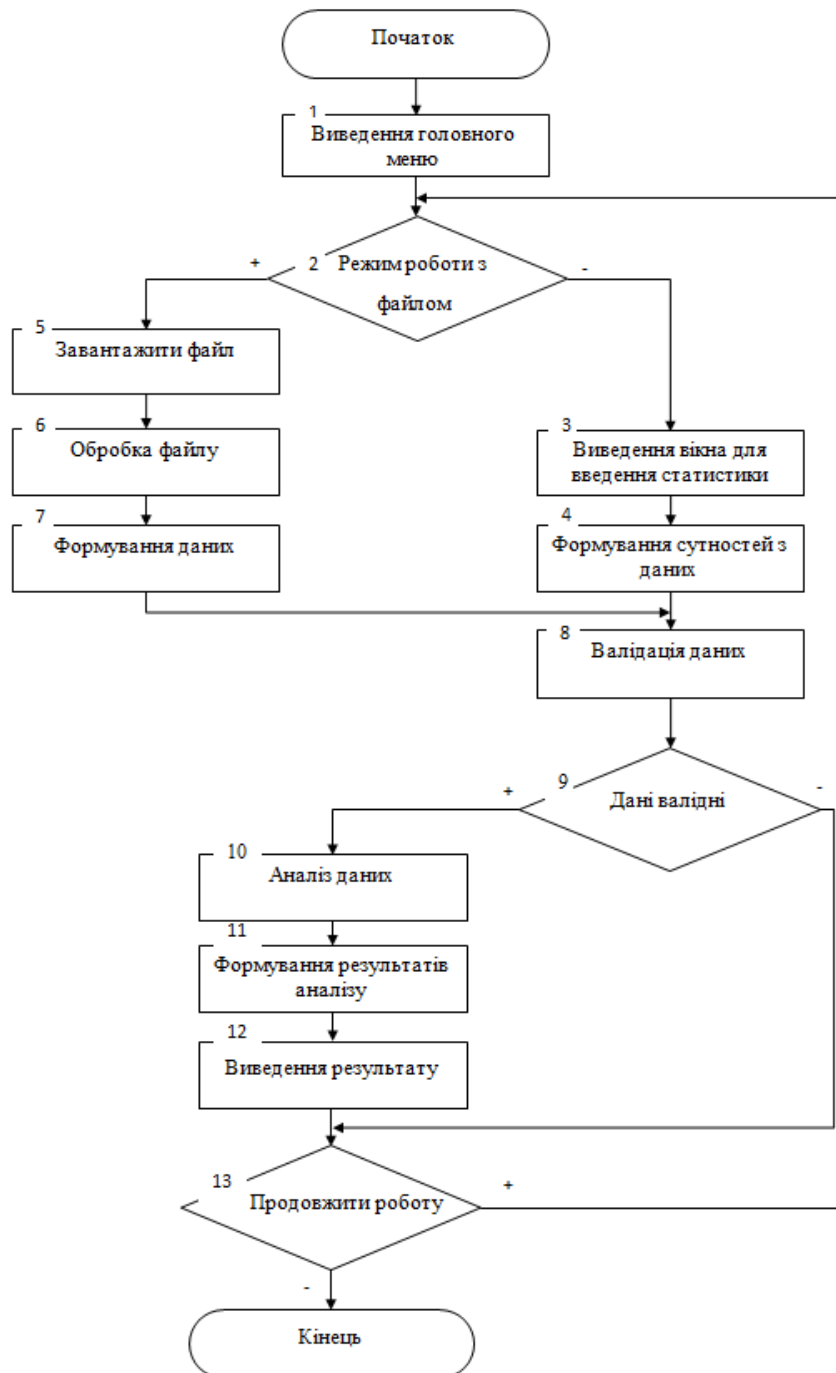


Рисунок 2.7 – Схема загального алгоритму функціонування системи

Крок 11. На наступному кроці відбувається формування результатів аналізу для прогнозування результату баскетбольного матчу.

Крок 12. Виведення результату аналізу у вигляді прогнозування результату баскетбольного матчу.

Крок 13. При необхідності продовжити роботу відбувається перехід до кроку 2.

Крок 14. Завершення діалогу з системою.

У наступному розділі описано процес програмної реалізації розробленої інформаційної технології аналізу статистики баскетбольних матчів.

2.5 Висновок до розділу 2

У другому розділі було розроблено математичну модель інформаційної технології для аналізу статистики баскетбольних матчів на основі методів нечіткої логіки. Цей метод є одним з найефективніших для вирішення подібних задач, а також робить можливим прогнозування баскетбольних матчів. Також було розроблено архітектуру інформаційної технології. У даному розділі було розроблено загальну структурну схему та схему алгоритмів функціонування інформаційної технології для аналізу статистики баскетбольних матчів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ

3.1 Обґрунтування вибору мови та середовища програмування

Для розробки програмної реалізації розробленої інформаційної технології аналізу статистики баскетбольних матчів необхідно вирішити завдання вибору мови програмування, оскільки доведення доцільності вибору мови програмування має високий вплив на швидкість та простоту розробки програмного забезпечення. Для реалізації інформаційної технології було розглянуто дві об'єктно орієнтовані мови – Java та Python. Наведемо їх основні характеристики.

Python – це високорівнева об'єктно-орієнтована мова програмування, яка інтерпретується за допомогою сильної динамічної типізації [28]. Розроблено Гвідо ван Россумом у 1990 році. Високорівневі структури даних разом із динамічною семантикою та динамічним зв'язуванням роблять його привабливим інструментом для швидкої розробки програмного забезпечення, а також способом інтеграції існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python і стандартні бібліотеки доступні у скомпільованому та рідному вигляді на всіх основних платформах. Мова програмування Python підтримує декілька парадигм програмування, включаючи об'єктно-орієнтовану, процедурну, функціональну та аспектно-орієнтовану.

Переваги Python:

1. Легко читати, вчитися та писати. Python — це мова програмування високого рівня з синтаксисом, подібним до англійської. Це полегшує читання та розуміння коду. Python дуже легко освоїти та вивчити, тому багато людей рекомендують Python для початківців. Вам потрібно менше рядків коду для виконання того ж завдання, ніж інші основні мови, такі як C/C++ і Java.

2. Підвищити продуктивність. Python — дуже продуктивна мова. Простота Python дозволяє розробникам зосередитися на вирішенні проблеми. Вам не доведеться витратити багато часу на розуміння синтаксису чи поведінки мови програмування. Ви пишете менше коду, а виконуєте більше.

3. Інтерпретована мова. Python є інтерпретованою мовою, що означає, що Python виконує код рядок за рядком безпосередньо. У разі помилки він зупиняє подальше виконання та повідомляє про помилку. Python показує лише одну помилку, навіть якщо програма містить кілька помилок. Це полегшує налагодження.

4. Динамічно типізований. Python не знає тип змінної, поки ми не запустимо код. Він автоматично призначає тип даних під час виконання. Програмісту не доведеться турбуватися про оголошення змінних та їх типів даних.

5. Вільний і відкритий. Python ліцензується за ліцензією з відкритим кодом, схваленою OSI. Це робить його безкоштовним у використанні та розповсюдженні. Ви можете завантажити вихідний код, змінити його і навіть розповсюдити свою версію Python. Це корисно для організацій, які хочуть змінити певну поведінку та використовувати власну версію для розробки.

6. Величезна підтримка бібліотеки

Стандартна бібліотека Python величезна, ви можете знайти майже всі функції, необхідні для вашого завдання. Тож вам не доведеться покладатися на зовнішні бібліотеки. Але навіть якщо ви це зробите, менеджер пакетів Python (pip) дозволяє легко імпортувати інші чудові пакунки з індексу пакетів Python (PyPi). Він складається з понад 200 000 упаковок.

7. Портативність. У багатьох мовах, наприклад В. С/С++, вам потрібно змінити свій код, щоб запустити програму на різних платформах. Це не те саме з Python. Ви пишете лише один раз і запускаєте його всюди. До недоліків мови Python можна віднести низьку швидкість, відсутність статичного введення, неможливість змінювати вбудовані класи та глобальне блокування інтерпретатора.

Однак, окрім переваг, Python має багато недоліків:

1. Повільна швидкість. Послідовне виконання коду часто призводить до повільного виконання. Динамічна природа Python також відповідає за повільність Python, оскільки він повинен виконувати додаткову роботу під час виконання коду. Тому Python не використовується для цілей, де швидкість є важливим аспектом проекту.

2. Неefективна пам'ять. Щоб полегшити роботу розробнику, Python має піти на невеликий компроміс. Мова програмування Python споживає багато пам'яті. Це може бути недоліком при створенні програм, якщо ви віддаєте перевагу оптимізації пам'яті.

3. Слабкий у мобільних обчисленнях. Python зазвичай використовується в програмуванні на стороні сервера. Ми не можемо бачити Python на стороні клієнта або в мобільних програмах з наступних причин. Python не ефективний з використанням пам'яті та має повільну потужність обробки порівняно з іншими мовами.

4. Доступ до бази даних. Програмування на Python під час взаємодії з базою даних не можна назвати легким. Рівень доступу до бази даних Python є примітивним і недостатньо розвиненим у порівнянні з поширеними технологіями, такими як JDBC і ODBC. Великі підприємства потребують плавної взаємодії зі складними застарілими даними, тому Python рідко використовується на підприємствах.

5. Помилка виконання. Як відомо, Python — це мова з динамічним типом, тому тип даних змінної може змінитися в будь-який момент. Ціла змінна може містити рядок у майбутньому, що може призвести до помилок під час виконання.

Java – об'єктно-орієнтована мова програмування, яку розробила компанія Sun Microsystems з 1991 року і офіційно випустила 23 травня 1995 року. Ця мова запозичила багато синтаксису з C і C++. Зокрема, він заснований на об'єктній моделі C++, але модифікований. Усуває можливість певних конфліктних ситуацій, які можуть виникнути через помилки

програмування та полегшує процес розробки об'єктно-орієнтованих програм. Набір дій, які програмістам необхідно виконати на C/C++, довірено віртуальній машині. Розроблена насамперед як мова, незалежна від платформи, Java має менше апаратних можливостей низького рівня, що знижує швидкість додатків порівняно, наприклад, із C++. При необхідності Java дозволяє викликати підпрограми, написані іншими мовами програмування [29].

- При створенні мови програмування Java було п'ять початкових цілей:
- Синтаксис мови має бути «простим, об'єктно-орієнтованим і знайомим».
- Реалізація має бути «бездоганною та безпечною».
- «Архітектурна незалежність і портативність» повинні бути збережені.
- Висока продуктивність
- Мова має бути «інтерпретованою, багатопоточною, з динамічним зв'язуванням модулів» [30].

Використання Java як мови програмування має багато переваг. Давайте обговоримо всі ці переваги один за іншим.

1. Java проста. Проста мова програмування, яку легко вивчити та зрозуміти. Завдяки простим і зрозумілим кодам Java є однією з найпростіших мов програмування для вивчення та впровадження. Крім того, Java видаляє всі складні функції C і C++, такі як покажчики, структури, агрегації, і спрощує реалізацію коду.

2. Java — це об'єктно-орієнтована мова програмування. Однією з головних переваг Java є те, що це об'єктно-орієнтована мова програмування. Використання концепції ООП зробило Java простішою у реалізації та значно безпечнішою. Концепції ООП допомагають Java вирішувати реальні проблеми. Це також допомагає керувати великим кодом, розбиваючи його на маленькі частини з іменами.

3. Java — безпечна мова. Такі мови, як C і C++, використовують покажчики, які дозволяють отримати доступ до місця зберігання. Це загрожує

безпеці, оскільки вказівники можуть призвести до несанкціонованого доступу до пам'яті. Java також використовувала такі концепції ООП, як інкапсуляція, абстракція та успадкування, щоб підвищити безпеку та запобігти несанкціонованому доступу користувачів.

4. Java не залежить від платформи. Java дотримується функції WORA (Write Once Run Anywhere). Програми Java, написані в системі, можна запускати в будь-якій іншій системі з Java. Сумісність Java не залежить від операційної системи чи апаратного забезпечення, що робить платформу Java незалежною та надзвичайно гнучкою.

5. Java — мова програмування високого рівня. Програми Java написані мовою високого рівня, людською мовою. Він схожий на англійську, з кількома простими і легко запам'ятовуваними синтаксисами. Java має інтерпретатор, який перекладає код на мову машинного рівня, щоб машина могла його зрозуміти.

6. Java підтримує функцію передачі. Java — дуже портативна мова. Це пов'язано з тим, що Java не залежить від платформи і не вимагає спеціального обладнання для роботи. Це буквально робить Java сумісною майже з усіма можливими пристроями.

7. Java пропонує автоматичний збір сміття. У C або C++ ми повинні були звільнити пам'ять за допомогою програми. У Java JVM автоматично керує пам'яттю. Тому щоразу, коли об'єкт не посилається на клас і йому потрібно назвати, JVM автоматично видаляє його з програми, тому нам не потрібно писати будь-який додатковий код. Тому Java підтримує автоматичне збирання сміття.

8. Java підтримує багатопотоковість. Пропускна здатність — це найменша можлива одиниця процесу. Багатопоточність є ключем до максимального використання ЦП. Java — це мова програмування, яка підтримує багатопотоковість. Java дозволяє нам запускати кілька потоків одночасно. Вони мають спільну пам'ять для підвищення ефективності та продуктивності програми. Нитки проходять незалежно один від одного.

9. Java регулярно отримує оновлення для виправлення помилок. Це робить Java однією з найстабільніших мов програмування. Оновлення негайно виправляють майже всі помилки. Тому важливо регулярно оновлювати Java.

10. Java забезпечує ефективну стратегію розподілу пам'яті. Java в основному ділить пам'ять на дві частини: область купи і область стеку. За потреби JVM виділяє пам'ять з однієї з двох частин. Це дозволяє ефективно керувати сховищем.

Отже, Java має кілька плюсів, однак вона теж не ідеальна. У Java також є свої недоліки.

1. Повільна і слабка продуктивність. Java споживає більше пам'яті, ніж рідні мови програмування, такі як C і C++. Java також повільніше в порівнянні з ними, що пов'язано з додатковою роботою інтерпретатора для перетворення коду на машинну мову. JVM виконує різні функції бекенда, які сповільнюють швидкість роботи програми. Оскільки Java підтримує автоматичне збирання сміття, вона завжди працює на стороні сервера, що впливає на продуктивність.

2. Java значно відстає, коли справа доходить до графічного інтерфейсу. Конструктор графічного інтерфейсу Java не може створити складний інтерфейс користувача. У Java є багато фреймворків графічного інтерфейсу, таких як Swing, SWT, JavaFX, JSF тощо. Сучасні мови, такі як Python, R, C# тощо, мають найкращі дизайнери графічного інтерфейсу.

3. Відсутність резервних коштів. Java абсолютно не має можливості зберігати дані користувача. В основному він зосереджений на зберіганні даних, але не захищений інструментами резервного копіювання.

4. Java займає більше пам'яті, ніж інші мови програмування, такі як C і C++. Погане керування пам'яттю Java. Java використовує збирач сміття, але це негативно впливає на продуктивність.

5. Детальний і складний код. Java має багато багатослівних і складних синтаксисів. Іноді ці складні синтаксиси важко запам'ятати. З цих причин

багато програмістів віддають перевагу Python або C++ перед Java, оскільки вони мають відносно прості речення.

Отже, Java має як переваги, так і недоліки. Мінуси Java відносно менші в порівнянні з плюсами, тому її доцільність для використання при розробці програмного забезпечення аналізу статистики баскетбольних матчів досить висока.

На основі описів двох мов програмування обрано Java для розробки даного програмного забезпечення аналізу статистики баскетбольних матчів. Тому зробимо огляд програмного середовища для написання на обраній мові.

IntelliJ є одним із найпотужніших і найпопулярніших інтегрованих середовищ розробки (IDE) для Java. Він розроблений та підтримується JetBrains і доступний у спільноті та в остаточному виданні. Ця багатофункціональна IDE забезпечує швидкий розвиток і допомагає покращити якість коду [31].

IDE – це інтегроване середовище розробки. Це поєднання кількох інструментів, які роблять процес розробки програмного забезпечення простішим, надійним і менш схильним до помилок. Він має наступні переваги перед простим текстовим редактором.

- Інтеграція з корисними інструментами, такими як компілятор, налагоджувач, система контролю версій, інструменти збірки, різні фреймворки, профайлери додатків тощо.

- Підтримує функції навігації по коду, доповнення коду, рефакторингу та генерації коду, що прискорює процес розробки.

- Підтримує модульне тестування, тестування інтеграції та покриття коду за допомогою плагінів.

- Надає багатий набір плагінів для подальшого покращення функціональності IDE .

IntelliJ IDEA має одні з найкращих продуктивних можливостей завершення коду Java. Його алгоритм прогнозування може з точністю припустити, що кодер намагається ввійти, і завершить його для цього, навіть

якщо він не знає точної назви конкретного класу, члена або іншого ресурсу. Середовище дійсно розуміє код і має глибоке розуміння контексту кодера, що робить його унікальним серед інших IDE Java. Також доступна підтримка завершення коду на основі контексту. IDE перелічує найбільш релевантні символи, застосовні в поточному контексті.

Перевірки та виправлення — щоразу, коли IntelliJ виявляє, що ви збираєтеся зробити помилку, у тому ж рядку з'являється невелике яскраве сповіщення. Натиснувши на нього, відобразиться список пропозицій.

IntelliJ IDEA заснована на принципі кодування, що дозволяє розробникам писати код з мінімальною кількістю відволікаючих факторів. З цієї причини в даному випадку редактор є єдиним, видимим на екрані, з виділеними ярликами для всіх інших некодуючих функцій.

Середовище, орієнтоване на редактора. Швидкі спливаючі вікна допомагають переглядати додаткову інформацію, не виходячи з поточного контексту.

Інтегровані засоби розробки. Щоб допомогти розробникам організувати свій робочий процес, IntelliJ IDEA надає їм неймовірний набір інструментів, включаючи декомпілятор, підтримку Docker, переглядач байт-коду, FTP та багато інших інструментів.

Отже, розглянувши середовище розробки мовою Java IntelliJ IDEA та виділивши його переваги, доведено доцільність його використання для програмної реалізації інформаційної технології аналізу статистики баскетбольних матчів.

3.2 Використання фреймворків та бібліотек

Для ефективної реалізації інформаційної технології аналізу статистики баскетбольних матчів на Java, потрібно знайти та використати спеціалізовані бібліотеки та фреймворки.

Spring Framework є одним з найпотужніших і широко використовуваних фреймворків Java. Завдяки концепції впровадження залежностей і аспектно-орієнтованих функцій програмування Spring захопив світ розробників штурмом [32].

Це платформа з відкритим кодом, яка використовується для корпоративних програм. Фреймворк зазвичай використовується для Enterprise Java. Середовище надзвичайно універсальне і може використовуватися для створення більшості типів програм Java. За допомогою Spring розробники можуть створювати слабо пов'язані модулі, де залежностями керує фреймворк, а не покладаючись на бібліотеки коду.

Spring найчастіше використовується в розробці веб-додатків. Його використовують ряд технічних гігантів, включаючи Netflix і Amazon. Основна перевага фреймворка полягає в тому, що він надзвичайно легкий і не вимагає виклику веб-сервера. З точки зору ефективності, це одна з найпопулярніших платформ Java. Він надзвичайно простий і, як правило, його можуть використовувати навіть новачки. Він також пропонує зворотну сумісність і можливості тестування.

Фреймворк Spring є всеосяжним і включає багато функцій, включаючи конфігурацію безпеки та легке в освоєнні підключення до бази даних. Крім того, оскільки це найпопулярніший веб-фреймворк, можна знайти багато документації та активну спільноту.

Swing — це набір інструментів Java для графічних інтерфейсів користувача (GUI). Це інтерфейс прикладного програмування (API) для забезпечення графічного інтерфейсу користувача (GUI) для програм Java.

Він є частиною JFC (Java Foundation Classes), API, який надає графічний інтерфейс користувача для додатків Java. Він використовується для створення графічного інтерфейсу за допомогою Java.

Потреба в гойдалках. Оскільки для графічного інтерфейсу існує клас AWT, виникає питання, що потрібно Swing для графічного інтерфейсу.

Ось деякі моменти, які описують необхідність Swing:

- легкий;
- налаштовується;
- незалежний від платформи;
- кастомізований;
- розширюваний [33].

Розглянемо приклад використання бібліотеки у розробленому програмному забезпеченні. Наступний код програми демонструє відображення вікна головного меню.

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

@Component
@RequiredArgsConstructor
public class NewMatchWindow {

    private final MatchService matchService;

    private JFrame frame;
    private JLabel label;

    private JTextField team1Selection, team2Selection,
        team1Pass, team2Pass,
        team1Interception, team2Interception,
        team1Blockage, team2Blockage,
        team1Loss, team2Loss,
        team1RealizationGame, team2RealizationGame,
        team1Realization3Point, team2Realization3Point,
        team1RealizationPenalt, team2RealizationPenalt;

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Введення статистики");
        frame.setSize(600, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

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

```

initFields(font1);

JButton btnBack = new JButton();
btnBack.setText("    Аналізувати    ");
btnBack.setFont(font1);
label.add(btnBack, new GridBagConstraints(0, 8, 3, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
btnBack.addActionListener(new AnalizListener());

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

}

```

Отже, дані фреймворки дозволяють програмно реалізувати інформаційну технологію аналізу статистики баскетбольних матчів завдяки перерахованим вище перевагам.

3.3 Програмна реалізація аналізу статистики баскетбольних матчів

На рисунку 3.1 зображено UML-діаграму класів програмного забезпечення аналізу статистики баскетбольних матчів. Були розроблені класи Team, Match, BasketballAnalysisApplication, MatchParser, MatchService, UIService, NewMatchWindow.

Клас BasketballAnalysisApplication викликає клас UIService. Даний клас використовується для завантаження та запуску програми Spring з основного методу Java.

Клас UIService відповідає за відображення вікна початкової активності програми. Завдяки бібліотеці графічного інтерфейсу Swing в даному класі реалізується формування вікна, кнопок, та їх взаємодії.

Клас Match відповідає за створення об'єктів, які виступають командами з баскетболу.

Клас Team описує усі необхідні параметри, що впливають на аналіз та прогнозування баскетбольних матчів.

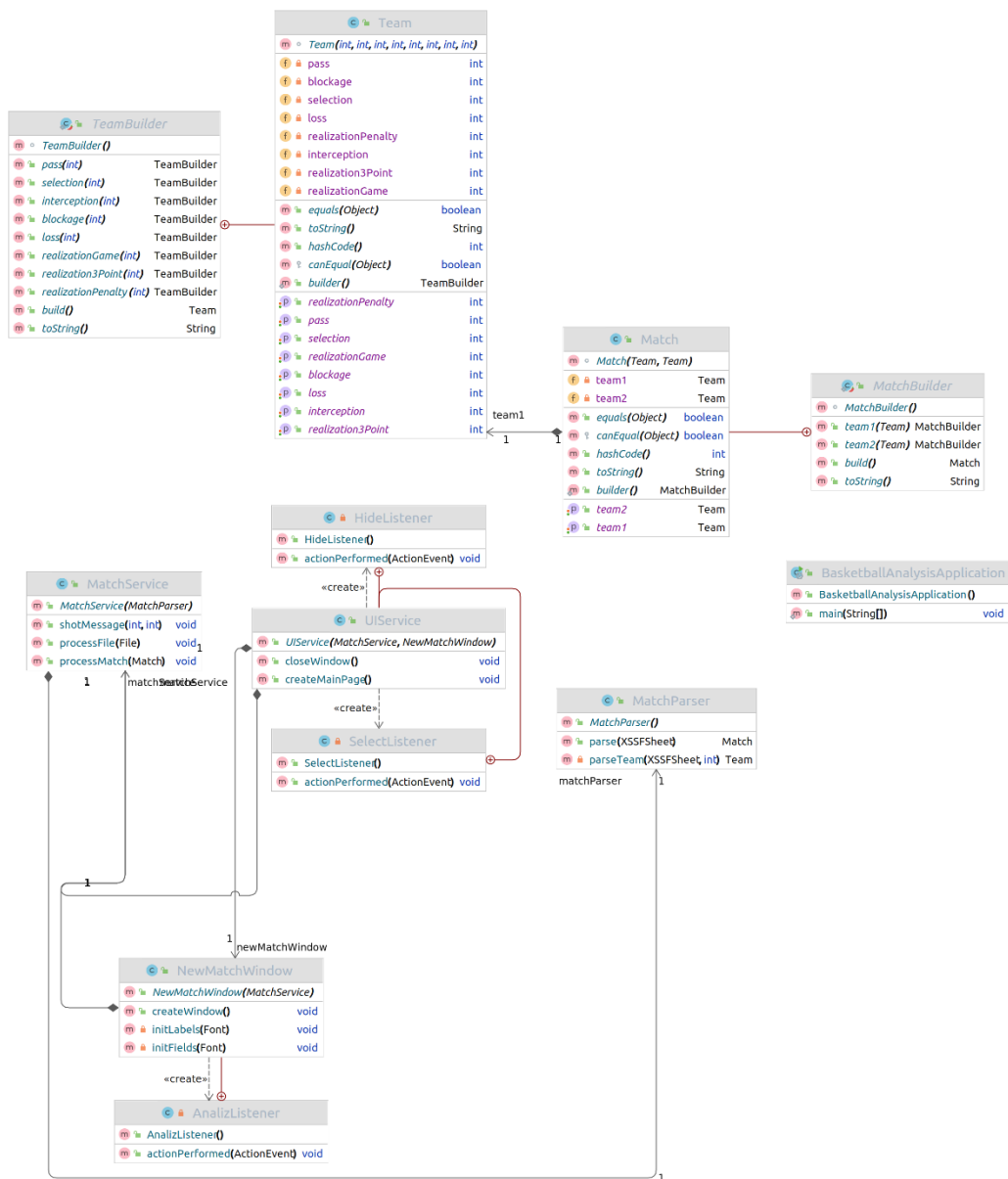


Рисунок 3.1 – UML-діаграма класів

Клас `MatchParser` відповідає за обробку даних електронної таблиці. Даний клас містить метод `XSSFSheet` для захисту клітин таблиці від змін. В даному класі відбувається повернення логічного рядка та клітинки таблиці, за допомогою методу `getRow` та `getCell`, з якого користувач отримує всю необхідну статистику, що формується у вигляді сунтостей даних.

Клас `NewMatchWindow` відповідає за отримання статистики, введеної користувачем, тобто в ньому реалізується створення вікна для додавання

нового матчу, необхідного для аналізу та за формування сутностей з даних, введених користувачем.

Клас MatchService, клас з основною логікою програми, дозволяє отримувати дані з електронної таблиці, а також дані, введені користувачем для їх подальшого аналізу. В класі реалізовано інтелектуальний алгоритм аналізу статистики баскетбольних матчів за допомогою застосування нечіткої логіки. Також клас відповідає за виведення результатів аналізу у вигляді прогнозованого результату баскетбольного матчу.

Щоб проаналізувати статистику баскетбольного матчу необхідно додати статистику про цей матч у відповідну форму. Додавання відбувається у формуванні введення даних у запропоновані поля, з допомогою доданої статистики можна проаналізувати матч та спрогнозувати його результат.

Діаграма діяльності сервера для введення статистики баскетбольного матчу безпосередньо через графічний інтерфейс користувача зображена на рисунку 3.2.

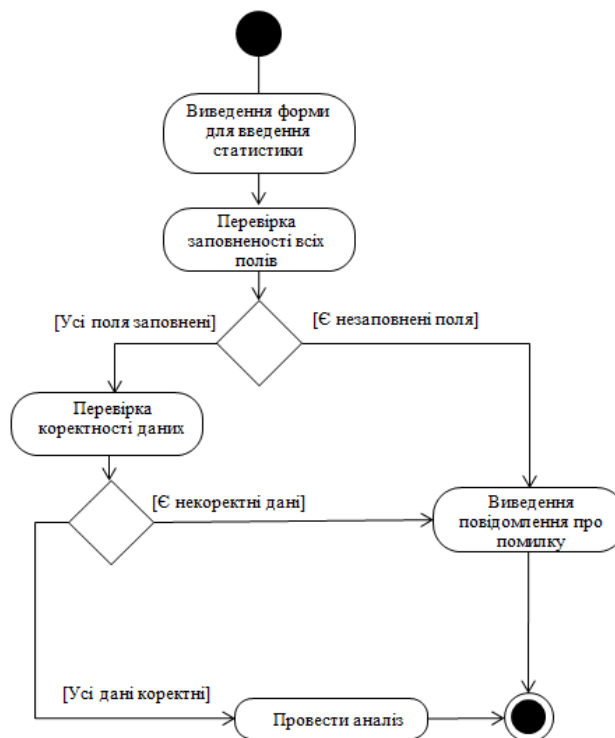


Рисунок 3.2 – Діаграма діяльності сервера для введення статистики матчу

Під час заповнення полів користувачем для аналізу статистики матчу сервер повинен перевірити коректність всіх заповнених даних.

Діаграма діяльності сервера для перевірки коректності введених даних зображена на рисунку 3.3.

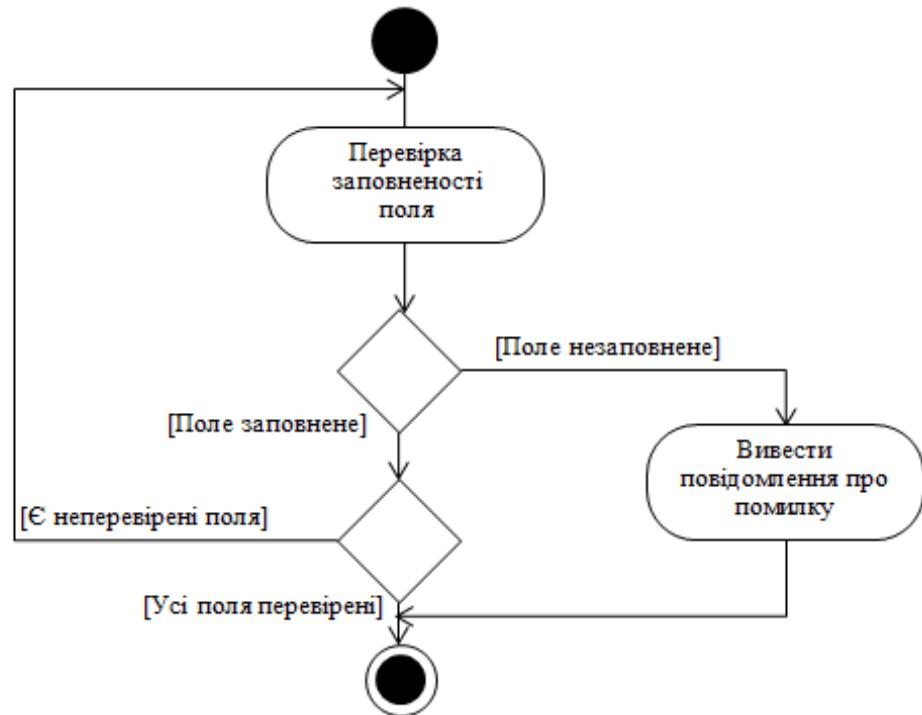


Рисунок 3.3 – Діаграма діяльності сервера для перевірки коректності введених даних

Відповідно до описаних діаграм реалізовано програмне забезпечення аналізу статистики баскетбольних матчів.

3.4 Висновок до розділу 3

В третьому розділі було здійснено обґрунтування вибору мови програмування для реалізації інформаційної технології аналізу статистики баскетбольних матчів. Програмне забезпечення розроблено на об'єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність, об'єктно-орієнтованість та простоту розробки, в середовищі розробки IntelliJ

IDEA, що має переваги у вигляді глибокого розуміння коду, задоволенні усіх вимог розробника, розумного доповнення коду. Було використано спеціалізований фреймворк Spring, а для розробки графічного інтерфейсу – набір інструментів Swing. Розроблено та описано діаграму класів, розглянуто методи та атрибути розроблених класів. Розроблено діаграму діяльності сервера для введення статистики матчу та діаграму діяльності сервера для перевірки коректності введених даних.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ АНАЛІЗУ СТАТИСТИКИ БАСКЕТБОЛЬНИХ МАТЧІВ

4.1 Тестування програми аналізу статистики баскетбольних матчів

Розроблене програмне забезпечення аналізу статистики баскетбольних матчів було протестовано, що підтвердило коректність його роботи.

Було проведено 500 запусків додатку, протестовано функціональні можливості його роботи, що дозволило адекватно оцінити його роботу.

Після запуску програми відкривається вікно початкової активності з двома запропонованими опціями – «Завантажити файл із статистикою» та «Ввести статистику» (рисунок 4.1).

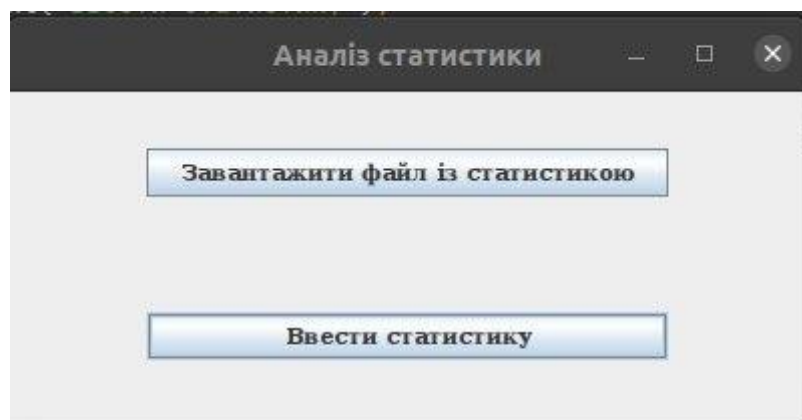


Рисунок 4.1 – Початкова активність

Аби почати роботу програми необхідно обрати одну із запропонованих опцій. При виборі опції «Ввести статистику» відкривається вікно з полями для введення статистики для подальшого аналізу (рисунок 4.2). Необхідно заповнити запропоновані поля та вибрати опцію «Аналізувати» (рисунок 4.3), після чого буде виведена інформація про результат аналізу (рисунок 4.4).

<input type="text"/>	Кількість підборів за гру	<input type="text"/>
<input type="text"/>	Кількість передач за гру	<input type="text"/>
<input type="text"/>	Кількість перехоплень за гру	<input type="text"/>
<input type="text"/>	Кількість блокшитів за гру	<input type="text"/>
<input type="text"/>	Кількість втрат за гру	<input type="text"/>
<input type="text"/>	Відсоток реалізації кидків з гри	<input type="text"/>
<input type="text"/>	Відсоток реалізації 3-х очкових кидків	<input type="text"/>
<input type="text"/>	Відсоток реалізації штрафних кидків	<input type="text"/>
<input type="button" value="Аналізувати"/>		

Рисунок 4.2 – Вікно введення статистики

44	Кількість підборів за гру	48
25	Кількість передач за гру	27
7	Кількість перехоплень за гру	7
3	Кількість блокшитів за гру	5
14	Кількість втрат за гру	11
47	Відсоток реалізації кидків з гри	47
38	Відсоток реалізації 3-х очкових кидків	37
81	Відсоток реалізації штрафних кидків	88
<input type="button" value="Аналізувати"/>		

Рисунок 4.3 – Таблиця з статистичними даними

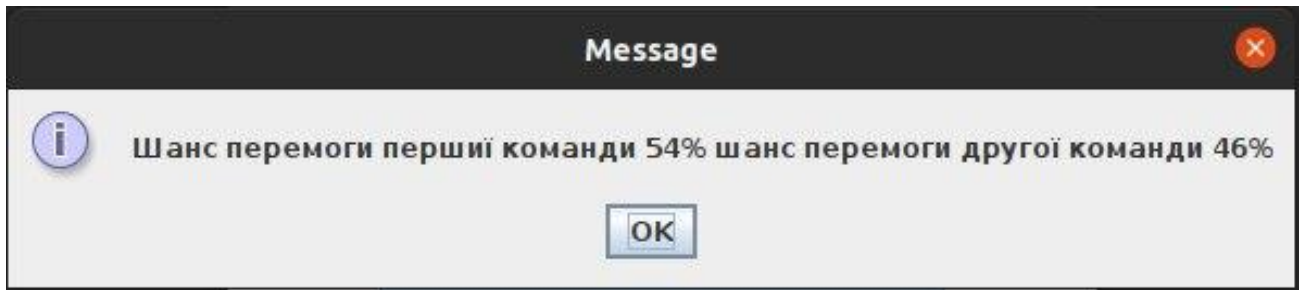


Рисунок 4.4 – Результат прогнозування матчу

Коли користувач обрав опцію «Завантажити файл із статистикою» - відкривається діалогове вікно для відкриття документів (рисунок 4.5). Необхідно вибрати потрібний документ та натиснути кнопку «Open», після чого завантажиться файл із статистикою (рисунок 4.6).

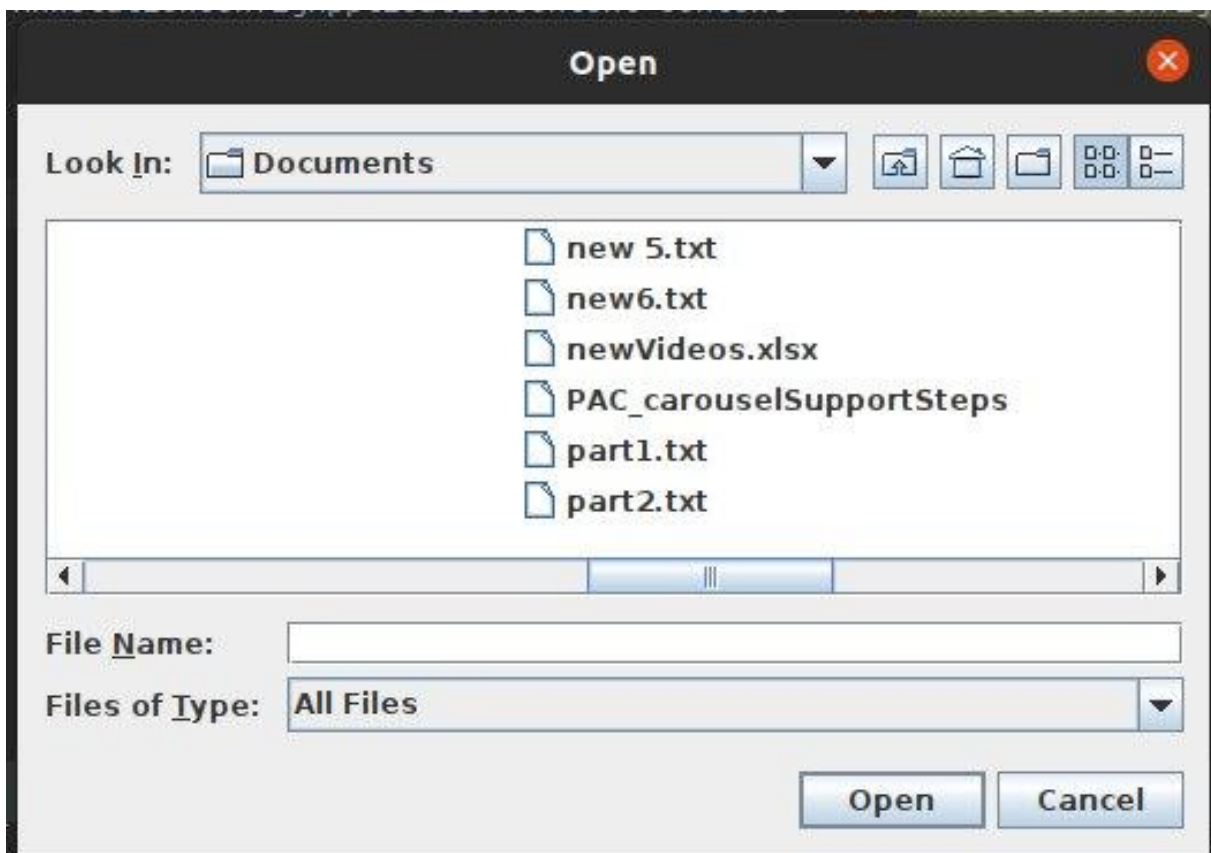


Рисунок 4.5 – Діалогове вікно для відкриття документів

	Команда 1	Команда 2
Кількість підборів за гру	45	46
Кількість передач за гру	27	24
Кількість перехоплень за гру	9	7
Кількість блокшитів за гру	5	6
Кількість втрат за гру	12	13
Відсоток реалізації кидків з гри	48	47
Відсоток реалізації 3-х очкових кидків	36	36
Відсоток реалізації штрафних кидків	79	82

Рисунок 4.6 – Перегляд статистики, що міститься в файлі

Після цього потрібно вибрати опцію «Аналізувати», після чого буде виведена інформація про результат аналізу (аналогічно як на рисунку 4.4).

4.1 Аналіз результатів роботи програми аналізу статистики баскетбольних матчів

Розроблений додаток має розширений функціонал, так як в ньому присутня функція прогнозування результатів баскетбольних матчів, якої немає у таких аналогів, як 24score , NBA: Live Games & Scores та FlashScore. Для визначення переваг по достовірності прогнозування розробленого програмного забезпечення перед аналогами було проведено його тестування на основі тестового набору даних, що складався із 100 прикладів реальних баскетбольних матчів, для яких були відомі всі аналізовані параметри та результати матчу. Розроблена програма із 100 матчів вірно спрогнозувала 81 матч, а програма-аналог Basketball total predictor із 100 матчів вірно спрогнозувала 77 матчів. В таблиці 4.1 наведено порівняльні результати роботи розробленої програми з програмами-аналогами.

Таблиця 4.1 – Порівняльний аналіз роботи розроблюваного додатку та аналогів

Програма	Автоматизований аналіз	Можливість додавання статистики	Точність прогнозування
Розроблений додаток	+	+	81%
24score	+	-	-
NBA: Live Games & Scores	+	-	-
FlashScore	+	-	-
Basketball total predictor	+	-	77%

З таблиці випливає, що розроблений додаток, як і програми-аналоги, здатний до автоматизованого аналізу статистичних даних, однак, окрім цього, він реалізує функцію додавання статистики вручну, що не реалізовано в розглянутих програмах, а також розроблений додаток, як і аналог «Basketball total predictor» здатний прогнозувати результати баскетбольних матчів. Порівняно з прототипом точність прогнозування розроблений додаток має вищу. Точність прогнозування зросла на 4% (81% проти 77%), що є хорошим результатом.

4.3 Висновок до розділу 4

У результаті тестування програми було доведено її повну працездатність та відповідність поставленому завданню. Розроблений додаток, як і програми-аналоги, здатний до автоматизованого аналізу

статистичних даних, однак, окрім цього, він реалізує функцію додавання статистики вручну, що не реалізовано в розглянутих програмах, а також розроблений додаток, як і аналог «Basketball total predictor» здатний прогнозувати результати баскетбольних матчів. Розроблений додаток має вищу достовірність прогнозування баскетбольних матчів (81%), ніж аналогічна програма (77%), а значить достовірність прогнозування баскетбольних матчів покращена на 4%, тобто мета роботи досягнута.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного та технологічного аудиту інформаційної технології аналізу статистики баскетбольних матчів

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів [34.35], якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 5.1.

Таблиця 5.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу засобу поляриметричного аналізу оптично активних рідни

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	2	2	2
Ринкові переваги (наявність аналогів)	3	3	3
Ринкові переваги (ціна продукту)	4	2	3
Ринкові переваги (технічні властивості)	3	2	2
Ринкові переваги (експлуатаційні витрати)	2	2	2
Ринкові перспективи (розмір ринку)	3	3	2
Ринкові перспективи (конкуренція)	2	2	3
Практична здійсненність (наявність фахівців)	2	3	2

Продовження табл. 5.1

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Практична здійсненність (наявність фінансів)	3	3	3
Практична здійсненність (необхідність нових матеріалів)	3	3	2
Практична здійсненність (термін реалізації)	3	3	2
Практична здійсненність (розробка документів)	2	2	2
Сума балів	32	30	28
Середньоарифметична сума балів, СБ	30		

За результатами розрахунків, наведених в таблиці 1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології аналізу статистики баскетбольних матчів – вище середнього.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці» [34,35].

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p},$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 5.2.

Таблиця 5.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	40000	1818	5	9090
Розробник	35000	1591	30	47730
Консультанти	30000	1364	7	9548
Всього:				66368

Додаткова заробітна плата. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,12 \cdot (Z_o + Z_p) = 0,12 \cdot (66368 + 0) = 7964 \text{ грн.}$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $N_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned}
 H_{зп} &= \beta \cdot (З_о + З_р + З_д) = \\
 &= 0,22 \cdot (66368 + 0 + 7964) = 16353 \text{ грн.}
 \end{aligned}$$

де $З_о$ – основна заробітна плата розробників, грн.; $З_р$ – основна заробітна плата робітників, грн.; $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Програмне забезпечення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i,$$

де $C_{\text{іпрг}}$ – ціна придбання програмного забезпечення і-го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1,1 \dots 1,12)$; k – кількість видів програмного забезпечення.

Таблиця 5.3 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
IntelliJ IDEA Ultimate	6591	1	6591
Всього, з врахуванням коефіцієнта інсталяції та налагодження			7250

Амортизація обладнання. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_б}{T_в} \cdot \frac{t}{12},$$

де $Ц_б$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; $T_в$ – термін корисного використання основного фонду, роки.

Таблиця 5.4 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	32000	5	1,5	800
Всього				800

Витрати на електроенергію для науково-виробничих цілей. Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 5.5 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук		

$$C_e = \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{\text{впі}}}{\text{ККД}} = \frac{0,15 \cdot 200 \cdot 7,5 \cdot 0,95}{0,95} = 810 \text{ грн.},$$

W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; $K_{\text{впі}}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Інші витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{N_{\text{ІВ}}}{100\%} = (66368 + 0) \cdot \frac{60}{100} = 39821 \text{ грн.},$$

де $N_{\text{ІВ}}$ – норма нарахування за статтею «Інші витрати».

Накладні (загальновиробничі) витрати. До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{\text{нзв}} = (З_0 + З_p) \cdot \frac{Н_{\text{нзв}}}{100\%} = (66368 + 0) \cdot \frac{120}{100} = 79642 \text{ грн.},$$

де $Н_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} В_{\text{заг}} &= З_0 + З_p + З_{\text{дод}} + З_{\text{н}} + В_{\text{прг}} + А_{\text{обл}} + В_e + \\ + I_v + В_{\text{нзв}} &= 66368 + 7964 + 16353 + 7257 + 800 + 810 + \\ &+ 39821 + 79642 = 99552 \text{ грн.} \end{aligned}$$

Загальні витрати. Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{\text{заг}}}{\eta} = \frac{99552}{0,5} = 199104 \text{ грн.},$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії експериментального зразка, то $\eta=0,5$.

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження

результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; C_0 – вартість послуги у році до впровадження інформаційної системи; $\pm\Delta C_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де $\pm\Delta C$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta C_0$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; C_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; C_0 – основний якісний показник, який визначає ціну реалізації існуючої (базової)

науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta\Pi = ((4500 - 4000) \cdot 10000 - (10000 - 10000) \cdot 4500) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 102495900 \text{ грн.}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t} = \frac{102495900}{(1 + 0,1)^1} = 93178091 \text{ грн.},$$

де $\Delta\Pi$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 12 \cdot 199104 = 2389248 \text{ грн.}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2\dots5$, але може бути і більшим; $ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 93178091 - 2389248 = 90788843 \text{ грн.},$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_B , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} = \sqrt{1 + \frac{90788843}{2389248}} = 6,1,$$

де $T_{ж}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{мін}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$ визначається за формулою:

$$\tau_{мін} = d + f = 0,9 + 0,5 = 1,4,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,9...0,12$; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05...0,5$, але може бути і значно вищою.

Оскільки $E_B = 6,1 > \tau_{мін} = 1,4$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_B} = \frac{1}{6,1} = 0,16 \text{ року.}$$

Оскільки $T_o=0,16 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

5.4 Висновок до розділу 5

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу статистики баскетбольних матчів» становить 30,0 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 0,16 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія аналізу статистики баскетбольних матчів».

ВИСНОВКИ

При виконанні даної магістерської кваліфікаційної роботи розв'язано задачу розробки інформаційної технології та програмного забезпечення для прогнозування результатів баскетбольних матчів з використанням нечіткої логіки.

У першому розділі було проведено аналіз сучасних програм-аналогів, які використовуються для задач аналізу статистики баскетбольних матчів, а також інших спортивних подій. Було наведено короткий опис основних функцій, які виконують дані програми, що виявило їх нездатність до точного прогнозування результатів баскетбольних матчів. Також досліджено основні фактори впливу на статистику та результати баскетбольних матчів. Було досліджено методи, що можуть бути використані для аналізу статистики баскетбольних матчів, для чого запропоновано використати методи нечіткої логіки. Здійснено постановку задачі.

У другому розділі магістерської кваліфікаційної роботи було розроблено математичну модель інформаційної технології для аналізу статистики баскетбольних матчів на основі методів нечіткої логіки. Цей метод є одним з найефективніших для вирішення подібних задач, а також робить можливим прогнозування баскетбольних матчів. У даному розділі було розроблено загальну структурну схему та схему алгоритмів функціонування інформаційної технології для аналізу статистики баскетбольних матчів.

У третьому розділі було здійснено обґрунтування вибору мови програмування для реалізації інформаційної технології аналізу статистики баскетбольних матчів. Програмне забезпечення розроблено на об'єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність, об'єктно-орієнтованість та простоту розробки, в середовищі розробки IntelliJ IDEA, що має переваги у вигляді глибокого розуміння коду, задоволенні усіх

вимог розробника, розумного доповнення коду. Було використано спеціалізований фреймворк Spring, а для розробки графічного інтерфейсу – набір інструментів Swing. Розроблено та описано діаграму класів, розглянуто методи та атрибути розроблених класів. Розроблено діаграму діяльності сервера для введення статистики матчу та діаграму діяльності сервера для перевірки коректності введених даних..

У четвертому розділі у результаті тестування програми було доведено її повну працездатність та відповідність поставленому завданню. Розроблений додаток, як і програми-аналоги, здатний до автоматизованого аналізу статистичних даних, однак, окрім цього, він реалізує функцію додавання статистики вручну, що не реалізовано в розглянутих програмах, а також розроблений додаток, як і аналог «Basketball total predictor» здатний прогнозувати результати баскетбольних матчів. Розроблений додаток має вищу достовірність прогнозування баскетбольних матчів (81%), ніж аналогічна програма (77%), а значить достовірність прогнозування баскетбольних матчів покращена на 4%, тобто мета роботи досягнута.

У п'ятому розділі було визначено рівень комерційного потенціалу розробки, який становить 30,0 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 0,16 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гульчак Д.С., Колесницький О.К. Дослідження функціональних характеристик методів та засобів для прогнозування результатів баскетбольних матчів. Матеріали всеукраїнської науково-практичної Інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2022)». [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/schedConf/presentations>.
2. Гульчак Д.С., Паночишин Ю. М. «Інформаційна технологія аналізу статистики баскетбольних матчів», в Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)», Вінниця, 2024, [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/viewFile/16829/14042>.
3. Баскетбол України та всього світу: новини, результати. URL: https://xsport.ua/ua/basketball_s/
4. Гринь Л.В. Деякі аспекти педагогічної майстерності тренера викладача // Фізичне виховання студентів творчих спеціальностей. / ХГАДІ (ХХІІІ). Харків, 2002. №7. С. 58-63.
5. Ківернік О., Городянський С., Пітін М. Організаційні особливості процесу фізичного виховання у вищих навчальних закладах України // Слобожанський науково-спортивний вісник. 2008. № 1-2. С. 20-22.
6. Принципи і функції прогнозування. URL: https://pidruchniki.com/1002050640769/buhgalterskiy_oblik_ta_audit/printsipi_funktsiyi_prognozuvannya.
7. Початковий пункт для аналізу баскетбольної статистики. Журнал кількісного аналізу у спорті / Кубатько Дж. та ін. 2007. №3 С 15.
8. Зеленцов А. М., Лобановский В. В. Методологические основы разработки моделей тренировочных занятий, Київ: Здоров'я, 1998. 200с.

9. A machine learning framework for sport result prediction. URL: <https://www.sciencedirect.com/science/article/pii/S2210832717301485#f0015>.

10. Статистика на баскетбол URL: <https://bukmekerov.net/stati/cool/statistika-na-basketbol-gde-iskat/>.

11. Введение в теорию нечетких множеств и нечеткую логику. URL: http://www.nsu.ru/matlab/MatLab_RU/fuzzylogic/book1/index.asp.htm.

12. Ku J., Kim J. Development of game indicators and winning forecasting models with game data. Journal of the Korean Data & Information Science Society. 2017 №28. P. 237-250.

13. Таранцев А. А., Чернов В. Г. Метод прогнозирования эффективности выступления спортсменов в соревнованиях // Тренер. 1999. № 4. С.35-37.

14. Darwiche A. Modeling and reasoning with Bayesian networks. Cambridge : Cambridge University Press, 2009. 548 p.

15. Baran D., Apostol I. Signification of biorhythms for human performance assessment // Rev Med Chir Soc Med Nat Iasi. 2007. № 111(1). С. 295-302.

16. Evgeniou T., Pontil M. Support Vector Machines with Clustering for Training with Very Large Datasets. Springer : Lecture Notes in Computer Science, 2002. Vol. 2308, P. 346-354

17. Ruping S., mySVM-Manual. Technical Report. University of Dortmund, Computer Science Department, 2000.

18. 24score URL: <https://24score.pro/basketball/>.

19. Додаток NBA: Live Games & Scores URL: <https://play.google.com/store/apps/details?id=com.nbaimd.gametime.nba2011&hl=uk&gl=US>.

20. FlashScore URL: <https://www.flashscore.ua/>

21. Ротштейн А. П. Идентификация нелинейных зависимостей нечеткими базами знаний // Кибернетика и системный анализ. 1998. № 5. С. 53-61.

22. Методика прогнозування з допомогою теорії нечітких множин.
URL: <http://rudbet.com/5-9-metodika-prognozirovaniya-s-pomoshhyu-teorii-nechetkih-mnozhestv/>.
23. Ross, Timothy J. Fuzzy Logic with Engineering Applications / England : John Wiley & Sons Ltd., 2004. 628 p.
24. Jager, R. Fuzzy logic in control. Delft : Technische Universitet, 1995. 313 p.
25. Mamdani, E. H. Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of the Institution of Electrical Engineers. 1974 Vol 121 №12, P. 1585.
26. Toma M., Sports Econ J. Missed shots at the free-throw line analyzing the determinants of choking under pressure. 2017. №18. P. 539-559.
27. M. Dowling, B. Leopkey, L. Smith, Governance in sport: a scoping review, J. Sport Manage. 2018, №32 P. 438-451.
28. Переваги мови Python. Дистанційна підтримка освіти URL: <https://disted.edu.vn.ua/courses/learn/7649>.
29. Копитко М.Ф., Іванків К. С. Основи програмування мовою Java : тексти лекцій. Львів : ЛНУ ім. Івана Франка, 2002. 83с.
30. Advantages of Java. URL: <https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java> (дата звернення: 18.09.2023).
31. Почему IntelliJ IDEA. Продуктивная разработка на Java. URL: <https://www.jetbrains.com/ru-ru/idea/>.
32. Class SpringApplication URL: <https://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/SpringApplication.html>.
33. Introduction To Swing In Java URL: <https://www.c-sharpcorner.com/UploadFile/fd0172/introduction-of-swing-in-java/>.
34. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

35.Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

36.Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (PDF, 58 с.)

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Інформаційна технологія аналізу статистики баскетбольних матчівТип роботи: магістерська кваліфікаційна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 85,7% Схожість 14,3%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Гульчак Д.С.

Керівник роботи



Паночішин Ю.М.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

Клас BasketballAnalysisApplication

```
package com.example.basketballanalysis;

import com.example.basketballanalysis.service.UIService;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@ComponentScan
@Configuration
public class BasketballAnalysisApplication {

    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(BasketballAnalysisApplication.class);

        UIService service = context.getBean(UIService.class);

        service.createMainPage();
    }
}
```

Клас Match

```
package com.example.basketballanalysis.entity;

import lombok.Builder;
import lombok.Data;

@Data
@Builder
public class Match {

    private Team team1;
    private Team team2;
}
```

Клас Team

```
package com.example.basketballanalysis.entity;

import lombok.Builder;
import lombok.Data;
```

```

@Data
@Builder
public class Team {

    private int selection;
    private int pass;
    private int interception;
    private int blockage;
    private int loss;
    private int realizationGame;
    private int realization3Point;
    private int realizationPenalty;

}

```

Клас MatchParser

```

package com.example.basketballanalysis.parser;

import com.example.basketballanalysis.entity.Match;
import com.example.basketballanalysis.entity.Team;
import lombok.RequiredArgsConstructor;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.springframework.stereotype.Component;

@Component
@RequiredArgsConstructor
public class MatchParser {

    public Match parse(XSSFSheet sheet) {
        Team team1 = parseTeam(sheet, 1);
        Team team2 = parseTeam(sheet, 2);
        return Match.builder()
            .team1(team1)
            .team2(team2)
            .build();
    }

    private Team parseTeam(XSSFSheet sheet, int i) {
        return Team.builder()
            .selection((int) sheet.getRow(1).getCell(i).getNumericCellValue())
            .pass((int) sheet.getRow(2).getCell(i).getNumericCellValue())
            .interception((int) sheet.getRow(3).getCell(i).getNumericCellValue())
            .blockage((int) sheet.getRow(4).getCell(i).getNumericCellValue())
            .loss((int) sheet.getRow(5).getCell(i).getNumericCellValue())
            .realizationGame((int) sheet.getRow(6).getCell(i).getNumericCellValue())
            .realization3Point((int) sheet.getRow(7).getCell(i).getNumericCellValue())
            .realizationPenalty((int) sheet.getRow(8).getCell(i).getNumericCellValue())
            .build();
    }
}

```

Клас MatchService

```

package com.example.basketballanalysis.service;

import com.example.basketballanalysis.entity.Match;
import com.example.basketballanalysis.entity.Team;
import com.example.basketballanalysis.parser.MatchParser;
import lombok.RequiredArgsConstructor;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.springframework.stereotype.Service;

import javax.swing.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

@Service
@RequiredArgsConstructor
public class MatchService {

    private final MatchParser matchParser;

    public void processFile(File file) {
        try {
            InputStream targetStream = new FileInputStream(file);
            XSSFWorkbook workbook = new XSSFWorkbook(targetStream);
            XSSFSheet sheetAt = workbook.getSheetAt(0);
            Match match = matchParser.parse(sheetAt);
            processMatch(match);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    public void processMatch(Match match) {
        Team team1 = match.getTeam1();
        Team team2 = match.getTeam2();
        List<Integer> team1Info = new ArrayList<>();
        team1Info.add(100);
        team1Info.add(100);
        team1Info.add(100);
        team1Info.add(100);
        team1Info.add(team1.getRealizationGame());
        team1Info.add(team1.getRealization3Point());
        team1Info.add(team1.getRealizationPenalty());

        List<Integer> team2Info = new ArrayList<>();
        team2Info.add(team2.getSelection() * 100 / team1.getSelection());
    }
}

```

```

team2Info.add(team2.getPass() * 100 / team1.getPass());
team2Info.add(team2.getInterception() * 100 / team1.getInterception());
team2Info.add(team2.getBlockage() * 100 / team1.getBlockage());
team2Info.add(team2.getRealizationGame());
team2Info.add(team2.getRealization3Point());
team2Info.add(team2.getRealizationPenalty());

```

```

Integer team1Sum = team1Info.stream().mapToInt(Integer::intValue).sum();
Integer team2Sum = team2Info.stream().mapToInt(Integer::intValue).sum();

```

```

int percentT1 = team1Sum * 100 / (team1Sum + team2Sum);
if (team1.getLoss() > team2.getLoss()){
    percentT1-=4;
}else{
    percentT1+=4;
}
int percentT2 = 100 - percentT1;
shotMessage(percentT1, percentT2);

```

```

}

```

```

public void shotMessage(int percent1, int percent2) {
    JOptionPane.showMessageDialog(null, "Шанс перемоги першої команди " + percent1 +
"% шанс перемоги другої команди " + percent2 + "%");
}

```

```

}

```

Клас NewMatchWindow

```

package com.example.basketballanalysis.service;

import com.example.basketballanalysis.entity.Match;
import com.example.basketballanalysis.entity.Team;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;

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

@Component
@RequiredArgsConstructor
public class NewMatchWindow {

    private final MatchService matchService;

    private JFrame frame;
    private JLabel label;

```



```

private JTextField team1Selection, team2Selection,
    team1Pass, team2Pass,
    team1Interception, team2Interception,
    team1Blockage, team2Blockage,
    team1Loss, team2Loss,
    team1RealizationGame, team2RealizationGame,
    team1Realization3Point, team2Realization3Point,
    team1RealizationPenalt, team2RealizationPenalt;

public void createWindow() {
    frame = new JFrame();
    frame.setTitle("Введення статистики");
    frame.setSize(600, 500);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLocationRelativeTo(null);

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

    initLabels(font1);

    initFields(font1);

    JButton btnBack = new JButton();
    btnBack.setText("        Аналізувати        ");
    btnBack.setFont(font1);
    label.add(btnBack, new GridBagConstraints(0, 8, 3, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
    btnBack.addActionListener(new AnalizListener());

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

private void initLabels(Font font1) {
    JLabel teamLabel = new JLabel();
    teamLabel.setText("Кількість підборів за гру");
    teamLabel.setFont(font1);
    label.add(teamLabel, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

    JLabel possessionLabel = new JLabel();
    possessionLabel.setText("Кількість передач за гру");
    possessionLabel.setFont(font1);
}

```

```

label.add(possessionLabel, new GridBagConstraints(1, 1, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

JLabel shotsOnTargetLabel = new JLabel();
shotsOnTargetLabel.setText("Кількість перехоплень за гру");
shotsOnTargetLabel.setFont(font1);
label.add(shotsOnTargetLabel, new GridBagConstraints(1, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

JLabel shotsLabel = new JLabel();
shotsLabel.setText("Кількість блокшитів за гру");
shotsLabel.setFont(font1);
label.add(shotsLabel, new GridBagConstraints(1, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

JLabel numberOfAttacksLabel = new JLabel();
numberOfAttacksLabel.setText("Кількість втрат за гру");
numberOfAttacksLabel.setFont(font1);
label.add(numberOfAttacksLabel, new GridBagConstraints(1, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

JLabel numberOfDangerousAttacksLabel = new JLabel();
numberOfDangerousAttacksLabel.setText("Відсоток реалізації кидків з гри");
numberOfDangerousAttacksLabel.setFont(font1);
label.add(numberOfDangerousAttacksLabel, new GridBagConstraints(1, 5, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

JLabel xGLabel = new JLabel();
xGLabel.setText("Відсоток реалізації 3-х очкових кидків");
xGLabel.setFont(font1);
label.add(xGLabel, new GridBagConstraints(1, 6, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));

JLabel ppdaLabel = new JLabel();
ppdaLabel.setText("Відсоток реалізації штрафних кидків");
ppdaLabel.setFont(font1);
label.add(ppdaLabel, new GridBagConstraints(1, 7, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.CENTER, new Insets(20, 10, 10, 10), 0, 0));
}

private void initFields(Font font1) {
team1Selection = new JTextField(80);
team1Selection.setFont(font1);
label.add(team1Selection, new GridBagConstraints(0, 0, 1, 1, 0.2, 0.9,
GridBagConstraints.CENTER,

```

```

        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2Selection = new JTextField(80);
team2Selection.setFont(font1);
label.add(team2Selection, new GridBagConstraints(2, 0, 1, 1, 0.2, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1Pass = new JTextField(80);
team1Pass.setFont(font1);
label.add(team1Pass, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2Pass = new JTextField(80);
team2Pass.setFont(font1);
label.add(team2Pass, new GridBagConstraints(2, 1, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1Interception = new JTextField(80);
team1Interception.setFont(font1);
label.add(team1Interception, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2Interception = new JTextField(80);
team2Interception.setFont(font1);
label.add(team2Interception, new GridBagConstraints(2, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1Blockage = new JTextField(80);
team1Blockage.setFont(font1);
label.add(team1Blockage, new GridBagConstraints(0, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2Blockage = new JTextField(80);
team2Blockage.setFont(font1);
label.add(team2Blockage, new GridBagConstraints(2, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1Loss = new JTextField(80);
team1Loss.setFont(font1);
label.add(team1Loss, new GridBagConstraints(0, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2Loss = new JTextField(80);
team2Loss.setFont(font1);
label.add(team2Loss, new GridBagConstraints(2, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1RealizationGame = new JTextField(80);

```

```

team1RealizationGame.setFont(font1);
label.add(team1RealizationGame, new GridBagConstraints(0, 5, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2RealizationGame = new JTextField(80);
team2RealizationGame.setFont(font1);
label.add(team2RealizationGame, new GridBagConstraints(2, 5, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1Realization3Point = new JTextField(80);
team1Realization3Point.setFont(font1);
label.add(team1Realization3Point, new GridBagConstraints(0, 6, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2Realization3Point = new JTextField(80);
team2Realization3Point.setFont(font1);
label.add(team2Realization3Point, new GridBagConstraints(2, 6, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

team1RealizationPenalt = new JTextField(80);
team1RealizationPenalt.setFont(font1);
label.add(team1RealizationPenalt, new GridBagConstraints(0, 7, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
team2RealizationPenalt = new JTextField(80);
team2RealizationPenalt.setFont(font1);
label.add(team2RealizationPenalt, new GridBagConstraints(2, 7, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));
}

```

```
private class AnalizListener implements ActionListener {
```

```
    @Override
```

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

```
        Team team1 = Team.builder()
```

```
            .selection(Integer.getInteger(team1Selection.getText()))
```

```
            .pass(Integer.getInteger(team1Pass.getText()))
```

```
            .interception(Integer.getInteger(team1Interception.getText()))
```

```
            .blockage(Integer.getInteger(team1Blockage.getText()))
```

```
            .loss(Integer.getInteger(team1Loss.getText()))
```

```
            .realizationGame(Integer.getInteger(team1RealizationGame.getText()))
```

```
            .realization3Point(Integer.getInteger(team1Realization3Point.getText()))
```

```
            .realizationPenalty(Integer.getInteger(team1RealizationPenalt.getText()))
```

```
            .build();
```

```
        Team team2 = Team.builder()
```

```
            .selection(Integer.getInteger(team2Selection.getText()))
```

```
            .pass(Integer.getInteger(team2Pass.getText()))
```

```

        .interception(Integer.getInteger(team2Interception.getText()))
        .blockage(Integer.getInteger(team2Blockage.getText()))
        .loss(Integer.getInteger(team2Loss.getText()))
        .realizationGame(Integer.getInteger(team2RealizationGame.getText()))
        .realization3Point(Integer.getInteger(team2Realization3Point.getText()))
        .realizationPenalty(Integer.getInteger(team2RealizationPenalt.getText()))
        .build();

    Match match = Match.builder()
        .team1(team1)
        .team2(team2)
        .build();

    matchService.processMatch(match);
}
}
}

```

Клас UIService

```

package com.example.basketballanalysis.service;

import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;

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

@Service
@RequiredArgsConstructor
public class UIService {

    private final MatchService matchService;
    private final NewMatchWindow newMatchWindow;

    private JFrame frame;
    private JLabel label;
    private JComboBox comboBox;

    public void createMainPage() {
        frame = new JFrame();
        frame.setTitle("Аналіз статистики");
        frame.setSize(400, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 11);
        label = new JLabel();
    }
}

```

```

label.setLayout(new GridBagLayout());
label.setVisible(true);
label.setBackground(Color.BLUE);

JButton hideBtn = new JButton();
hideBtn.setText("Завантажити файл із статистикою");
hideBtn.setFont(font1);
label.add(hideBtn, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.HORIZONTAL, new Insets(10, 0, 10, 0), 0, 0));

JButton hide = new JButton();
hide.setText("Ввести статистику");
hide.setFont(font1);
label.add(hide, new GridBagConstraints(0, 1, 2, 1, 0.0, 0.9, GridBagConstraints.CENTER,
GridBagConstraints.HORIZONTAL, new Insets(10, 0, 10, 0), 0, 0));

hideBtn.addActionListener(new SelectListener());
hide.addActionListener(new HideListener());

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

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

private class SelectListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.showDialog(label, "Open");
        File file = fileChooser.getSelectedFile();
        matchService.processFile(file);
    }
}

private class HideListener implements ActionListener {

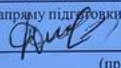
    @Override
    public void actionPerformed(ActionEvent JCom) {
        closeWindow();
        newMatchWindow.createWindow();
    }
}
}

```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ СТАТИСТИКИ
БАСКЕТБОЛЬНИХ МАТЧІВ

Виконав: студент 2-го курсу,
групи 1КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)
 Гульчак Д. С.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
 Паночішин Ю.М.
(прізвище та ініціали)
« 7 » 12 2023 р.

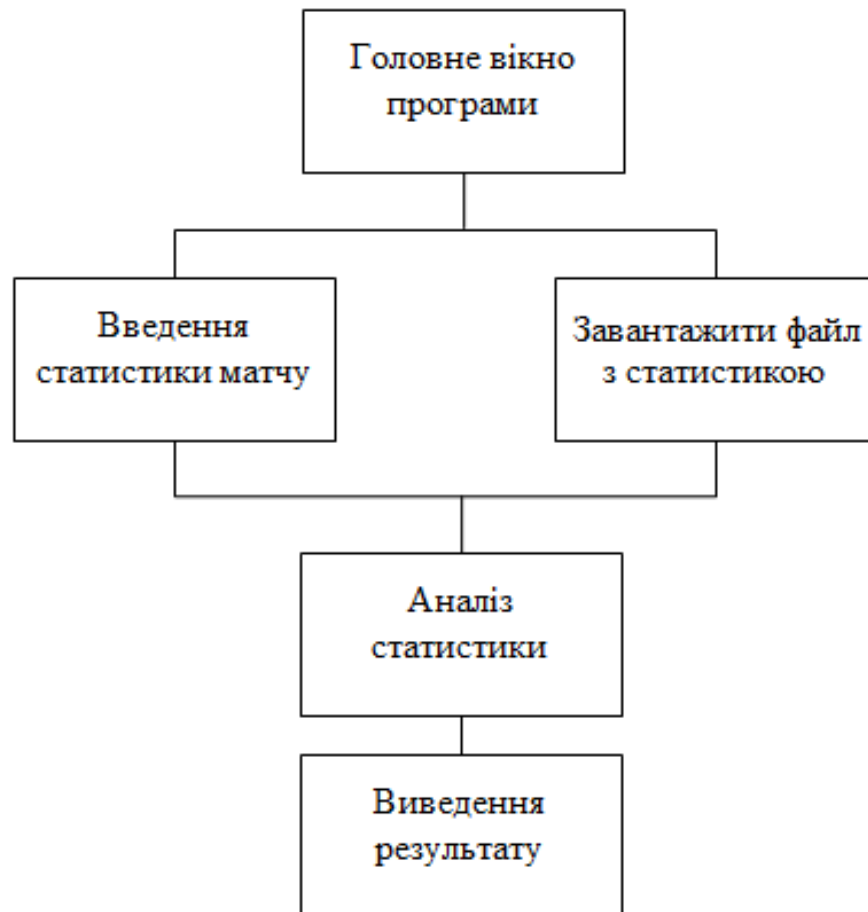


Рисунок В.1 – Архітектура програми аналізу статистики баскетбольних матчів



Рисунок В.2 – Загальна структура програми аналізу статистики баскетбольних матчів

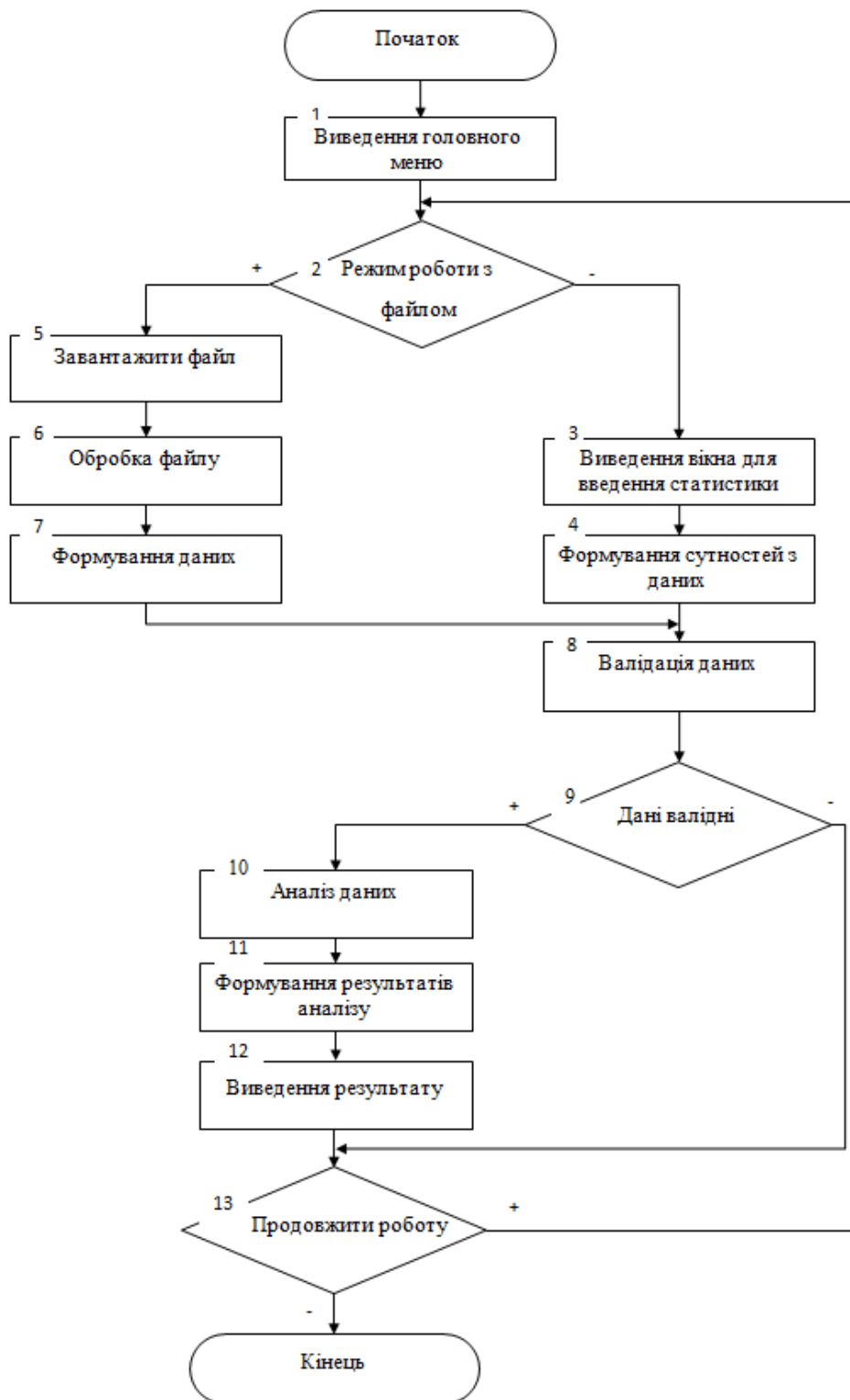


Рисунок В.3 – Граф-схема загального алгоритму роботи програми аналізу статистики баскетбольних матчів

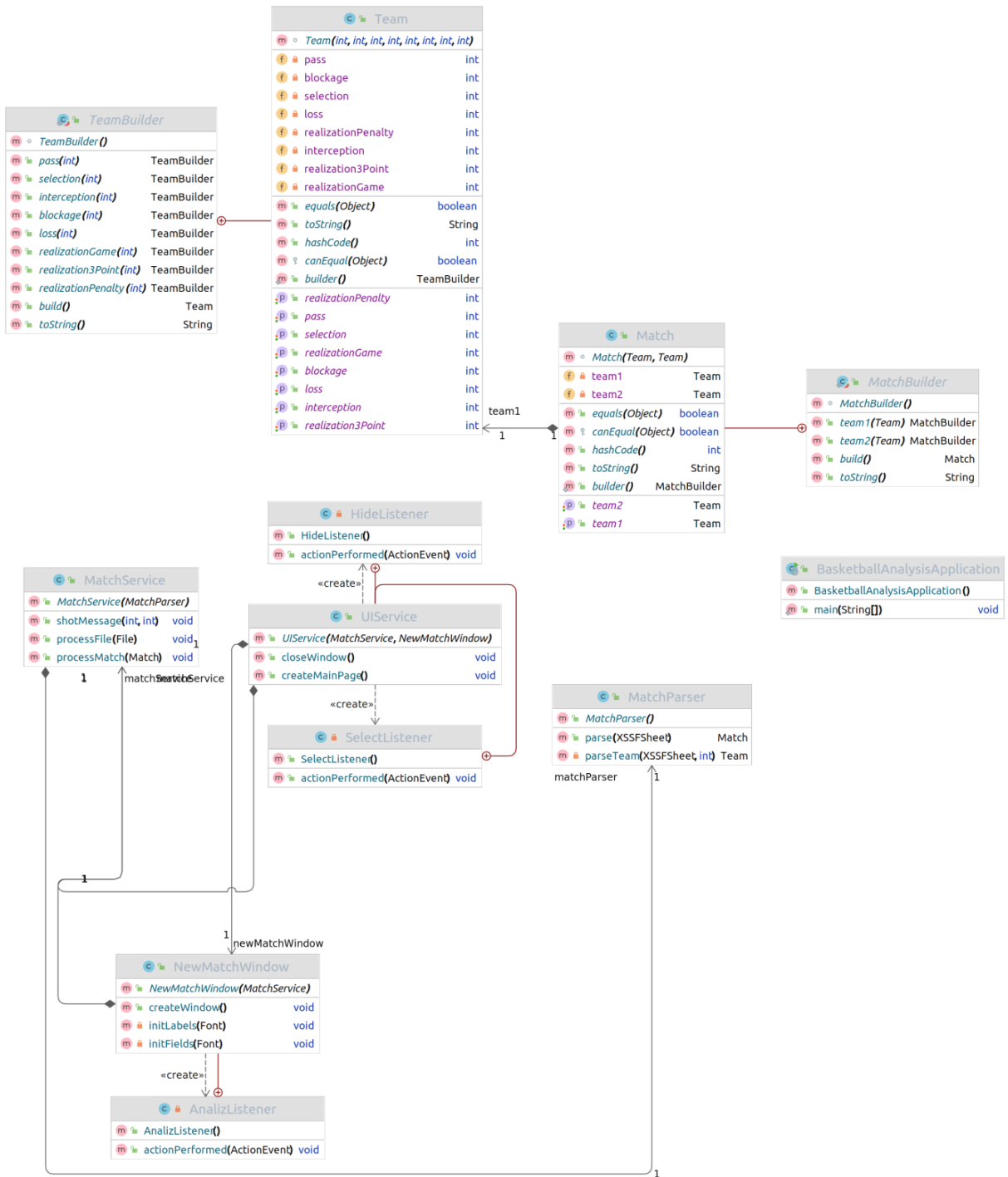


Рисунок В.4 – UML-діаграма класів програмного забезпечення аналізу статистики баскетбольних матчів

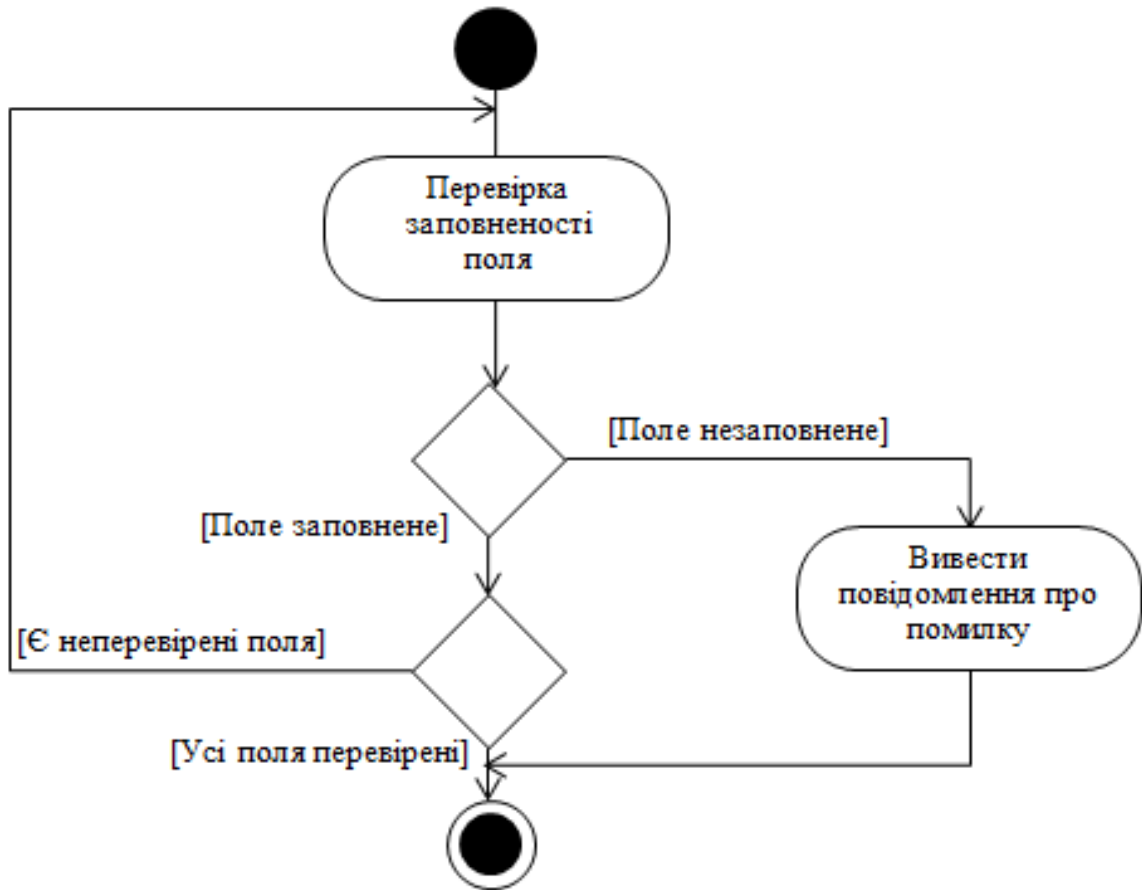


Рисунок В.5 – Діаграма діяльності сервера для перевірки коректності введених даних

The screenshot shows a window titled "Введення статистики" (Statistics Input) with a list of basketball statistics. Each statistic has two input fields, one on the left and one on the right. At the bottom of the window is a button labeled "Аналізувати" (Analyze).

Left Input	Statistic	Right Input
44	Кількість підборів за гру	48
25	Кількість передач за гру	27
7	Кількість перехоплень за гру	7
3	Кількість блокшитів за гру	5
14	Кількість втрат за гру	11
47	Відсоток реалізації кидків з гри	47
38	Відсоток реалізації 3-х очкових кидків	37
81	Відсоток реалізації штрафних кидків	88

Аналізувати

Рисунок В.6 – Результати роботи програми аналізу статистики баскетбольних матчів

Додаток Г (довідниковий)

Інструкція користувача

Після запуску програми відкривається вікно початкової активності (рисунок Г.1).

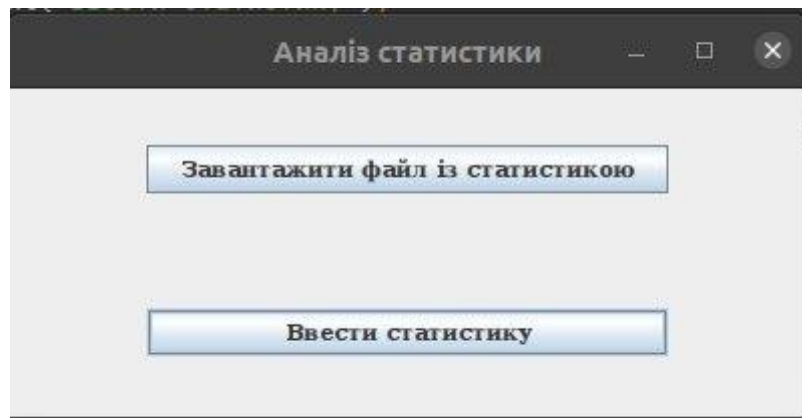


Рисунок Г.1 – Початкова активність

Аби почати роботу програми, необхідно обрати одну із запропонованих опцій. При виборі опції «Завантажити файл із статистикою» відкривається діалогове вікно відкриття файлів (рисунок Г.2). Користувач має обрати файл для завантаження, що містить інформацію про статистику матчів, які необхідно аналізувати (рисунок Г.3).

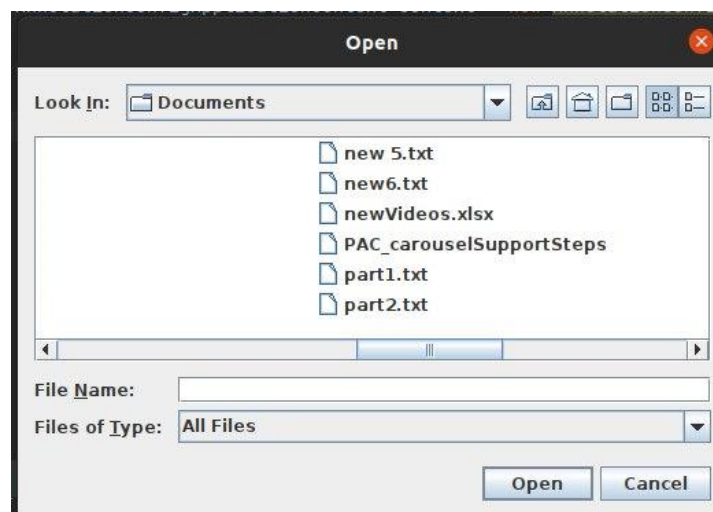


Рисунок Г.2 – Діалогове вікно для відкриття документів

	Команда 1	Команда 2
Кількість підборів за гру	45	46
Кількість передач за гру	27	24
Кількість перехоплень за гру	9	7
Кількість блокшитів за гру	5	6
Кількість втрат за гру	12	13
Відсоток реалізації кидків з гри	48	47
Відсоток реалізації 3-х очкових кидків	36	36
Відсоток реалізації штрафних кидків	79	82

Рисунок Г.3 – Файл зі статистикою

При виборі опції «Ввести статистику» користувачу пропонується вікно для введення статистики для команд з відповідними полями для введення (рисунок Г.4). Після введення статистики користувачем необхідно натиснути кнопку «Аналізувати» (рисунок Г.5).

Введення статистики		
<input type="text"/>	Кількість підборів за гру	<input type="text"/>
<input type="text"/>	Кількість передач за гру	<input type="text"/>
<input type="text"/>	Кількість перехоплень за гру	<input type="text"/>
<input type="text"/>	Кількість блокшитів за гру	<input type="text"/>
<input type="text"/>	Кількість втрат за гру	<input type="text"/>
<input type="text"/>	Відсоток реалізації кидків з гри	<input type="text"/>
<input type="text"/>	Відсоток реалізації 3-х очкових кидків	<input type="text"/>
<input type="text"/>	Відсоток реалізації штрафних кидків	<input type="text"/>
<input type="button" value="Аналізувати"/>		

Рисунок Г.4 – Введення статистики матчу

Value 1	Statistic	Value 2
44	Кількість підборів за гру	48
25	Кількість передач за гру	27
7	Кількість перехоплень за гру	7
3	Кількість блокшитів за гру	5
14	Кількість втрат за гру	11
47	Відсоток реалізації кидків з гри	47
38	Відсоток реалізації 3-х очкових кидків	37
81	Відсоток реалізації штрафних кидків	88

Аналізувати

Рисунок Г.5 – Введення статистики матчу

Після натиснення кнопки «Аналізувати» відбувається виведення результату аналізу через діалогове вікно у вигляді прогнозованого результату баскетбольного матчу.

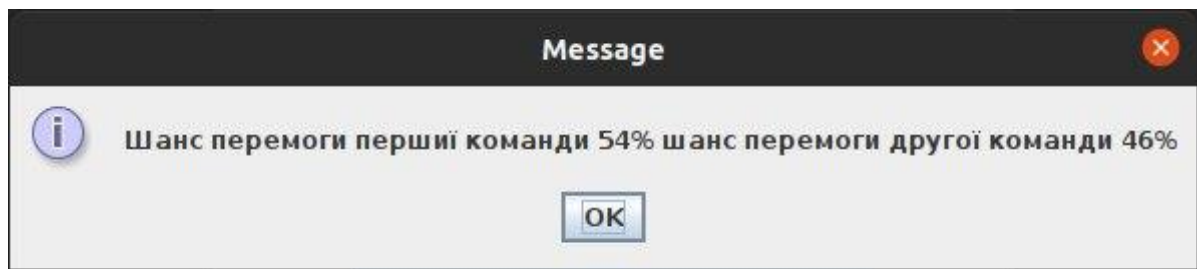


Рисунок Г.5 – Виведення результату