

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

## Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ДЛЯ АНАЛІЗУ ВИПАДКОВОСТІ У КОЛЕСІ ФОРТУНИ  
DEVELOPMENT OF SOFTWARE TO ANALYZE  
RANDOMNESS IN THE WHEEL OF FORTUNE

Виконав: студент 4 курсу, групи КНТ-111  
Спеціальності 121 Інженерія програмного  
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЛАЗАРЧУК Д.Е.

(ПРИЗВИЩЕ та ініціали)

Керівник ГОФМАН Є.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет КНТ  
Кафедра програмних засобів  
Ступінь вищої освіти бакалавр  
Спеціальність 121 Інженерія програмного забезпечення  
(код і найменування)  
Освітня програма (спеціалізація) Інженерія програмного забезпечення  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
С.О. СУББОТІН  
«      »                                  2025 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

ЛАЗАРЧУКА Данііла Едуардовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для  
аналізу випадковості у колесі фортуни. Development of Software to Analyze  
Randomness in the Wheel of Fortune  
керівник проєкту (роботи) к.т.н., ГОФМАН Євгеній Олександрович,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “28” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне  
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРИЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | ГОФМАН Є.О., доцент                       |                |                           |
| Нормоконтроль       | КАЛІНІНА М. В., асистент                  |                |                           |

7. Дата видачі завдання «14» квітня 2025 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту (роботи) | Примітка     |
|-------|----------------------------------------------------------------------|-----------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                               | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                               | Розділ 1     |
| 3     | Вибір мови програмування та інших технологій розробки.               | 2 тиждень                               | Розділ 2     |
| 4     | Розробка структури програми.                                         | 2 тиждень                               | Розділ 3     |
| 5     | Розробка програми.                                                   | 3-4 тижні                               | Розділи 3, 4 |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 5 тиждень                               | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 6 тиждень                               | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 7 тиждень                               |              |
| 9     | Захист роботи.                                                       | 8 тиждень                               |              |

Студент(ка)

\_\_\_\_\_ Данііл ЛАЗАРЧУК  
( підпис ) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_ Євгеній ГОФМАН  
( підпис ) (Імя ПРИЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:  
91 с., 6 табл., 26 рис., 3 дод., 11 джерел.

PYTHON, PYQT6, PYCHARM, QT DESIGNER, БАЗА ДАНИХ, SQLITE, CSS, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, КОЛЕСО ФОРТУНИ, АНАЛІЗ ДАНИХ.

Об'єкт дослідження – процес розроблення програмного забезпечення для аналізу випадковості випадання тих або інших категорій у колесі фортуни.

Предмет дослідження – методи та програмні засоби, що застосовуються для обчислення ймовірності випадання різних категорій у колесі фортуни, а також їх аналіз на основі реальних даних.

Метою роботи є підвищення ефективності ставок у грі за допомогою програмного забезпечення, що автоматично аналізує ймовірності різних категорій результатів обертання колеса фортуни на основі вже зафіксованих даних.

Матеріали, методи та технічні засоби: мова програмування Python, бібліотека для створення користувацького інтерфейсу PyQt6, PyCharm, Qt Designer, SQLite, CSS.

Результати. Розроблено ПЗ з користувацьким інтерфейсом, призначене для аналізу ймовірностей на основі результатів обертань колеса фортуни. Програма дозволяє вводити числа, що вже випали під час обертання колеса фортуни та автоматично обчислює ймовірності випадання тих або інших категорій.

Практична цінність роботи полягає в ефективному аналізі результатів обертання колеса фортуни на основі реальних даних. Це дозволяє підвищити шанси на успішний результат у грі.

Висновки. Реалізовано ПЗ для аналізу ймовірностей, засноване на фактичних результатах обертання колеса фортуни. Додаток забезпечений зручним інтерфейсом для ведення даних та автоматичного розрахунку ймовірностей випадіння категорій.

Галузь використання – платформи онлайн-ігор.

## **ABSTRACT**

Explanatory note to the diploma qualifying work of the bachelor: 91 pages, 6 tables, 26 figures, 3 appendixes, 11 sources.

PYTHON, PYQT6, PYCHARM, QT DESIGNER, DATABASE, SQLITE, CSS, GRAPHICAL INTERFACE, WHEEL OF FORTUNE, DATA ANALYSIS.

The object of research is the process of developing software for analyzing the probability of occurrence of specific sectors of the wheel of fortune.

The subject of the research is methods and software tools used to calculate the probability of different categories appearing of the wheel of fortune, as well as their analysis based on real data.

The aim of research is to improve the effectiveness of gameplay by using software that automatically analyzes the probabilities of different outcome categories of the wheel of fortune based on recorded data.

Materials, methods, and technical tools: object-oriented programming, the Python programming language, the PyQt6 library for graphical user interface development, the PyCharm and Qt Designer development environments, and the CSS styling language.

Results. Software with a graphical interface was developed to analyze probabilities based on the results of the wheel of fortune spins. The program allows users to input numbers that have already appeared during real spins and automatically calculates the probabilities of various result categories.

The practical value of the work lies in the effective analysis of the wheel of fortune outcomes and the optimization of decisions based on real data. This allows for better-informed conclusions in the context of random outcome analysis.

Conclusions. The developed software for probability analysis is based on actual results from the wheel of fortune spins. The application is equipped with a user-

friendly interface for data management and automatic calculation of the probability of drawn numbers or categories.

Scope of use – online gaming platforms.

## ЗМІСТ

|                                                                           |    |
|---------------------------------------------------------------------------|----|
|                                                                           | С. |
| Перелік скорочень та умовних позначок .....                               | 10 |
| Вступ .....                                                               | 11 |
| 1 Аналіз предметної області .....                                         | 13 |
| 1.1 Огляд предметної області .....                                        | 13 |
| 1.2 Огляд існуючих програмних рішень .....                                | 15 |
| 1.2.1 Сайт «Casinobot.net» .....                                          | 15 |
| 1.2.2 Сайт «Roulette Predictor» .....                                     | 16 |
| 1.2.3 Мобільний додаток «BP Roulette Analyzer» .....                      | 17 |
| 1.3 Потреба програмних рішень .....                                       | 18 |
| 1.4 Порівняння існуючих аналогів .....                                    | 19 |
| 1.5 Постановка завдання .....                                             | 20 |
| 1.5.1 Функціональність системи .....                                      | 20 |
| 1.5.2 Вимоги до інтерфейсу .....                                          | 21 |
| 1.6 Висновки за розділом 1 .....                                          | 21 |
| 2 Матеріали і методи .....                                                | 22 |
| 2.1 Вибір мови програмування .....                                        | 22 |
| 2.2 Вибір середовища розробки .....                                       | 24 |
| 2.3 Висновки за розділом 2 .....                                          | 25 |
| 3 Основні рішення щодо реалізації компонентів системи .....               | 26 |
| 3.1 Алгоритм роботи системи .....                                         | 26 |
| 3.2 Проєктування користувацького інтерфейсу .....                         | 31 |
| 3.3 Висновки за розділом 3 .....                                          | 33 |
| 4 Експлуатація, тестування та експериментальне дослідження програми ..... | 34 |
| 4.1 Призначення й умови застосування програми .....                       | 34 |
| 4.2 Вимоги до інтерфейсу .....                                            | 34 |
| 4.3 Умови виконання програми .....                                        | 35 |
| 4.4 Результати тестування .....                                           | 36 |
| 4.5 Висновки за отриманими результатами .....                             | 38 |

|                                    |    |
|------------------------------------|----|
| Висновки .....                     | 39 |
| Перелік джерел посилання .....     | 41 |
| Додаток А Технічне завдання .....  | 43 |
| Додаток Б Текст програми .....     | 47 |
| Додаток В Слайди презентації ..... | 83 |

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

- IDE – Integrated Development Environment;  
UML – Unified Modeling Language;  
ПЗ – програмне забезпечення;  
БД – база даних;  
СКБД – система керування базами даних.

## ВСТУП

Колесо фортуни – це пристрій, результат обертання якого є випадковим і не залежить від попередніх значень. У багатьох ігрових системах результати обертання фіксуються для подальшого аналізу. З часом накопичується значний обсяг таких результатів, і виникає потреба у швидкому та точному аналізі цих даних.

Аналіз імовірності результатів обертань колеса фортуни може бути корисним для спостереження, щоб оцінити, наскільки рівномірно розподіляються проценти ймовірності по тим або іншим категоріям на ігровому столі. Здійснювати такі розрахунки вручну незручно, повільно та, через людський фактор, неточно. Особливо це стосується ситуацій, коли кількість зафіксованих результатів перевищує десятки або сотні.

Отже актуальною задачею є розробка програмного забезпечення для аналізу ймовірностей на основі реальних результатів обертань колеса фортуни, тому що існує потреба в інструменті, який дозволяє швидко та точно аналізувати результати випадкових подій без складних математичних розрахунків з боку користувача. Аналізування таких обсягів числових послідовностей вручну є неефективним та більше за все неможливим, тому актуальним стає створення простого у використанні програмного забезпечення, яке автоматизує обчислення ймовірностей та забезпечує зручний інтерфейс.

Для виконання такої задачі необхідно виконати наступні завдання:

- провести аналіз предметної області та дослідити специфіку гри з колесом фортуни;
- визначити функціональні вимоги до програмного забезпечення та обґрунтувати вибір інструментів розробки;
- спроектувати структуру програми, реалізувати логіку збирання та збереження введених даних через СУБД SQLite;
- реалізувати алгоритми обчислення ймовірностей появи визначних секторів та категорій на основні статистичних підрахунків;

- створити користувацький інтерфейс користувача з використанням бібліотеки PyQt6 та середовищем розробки Qt Designer, забезпечити зручність ведення даних;
- виконати тестування програми, перевірити її стабільність, правильність розрахунків та зручність користування;
- проаналізувати результати роботи додатку.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Перший розділ пояснювальної записки буде присвячено аналізу предметної області. Буде розглянуто огляд основних характеристик рулетки, описано математичні принципи розрахунку ймовірностей та розглянуто особливості використання стратегії Мартінгейла, що дозволить автоматизувати аналіз результатів гри та оптимізувати процес прийняття рішень для користувачів.

### 1.1 Огляд предметної області

Основою предметної області є гра "колесо фортуни", що реалізована у вигляді європейської (класичної) рулетки. Європейська рулетка вважається однією з найпоширеніших азартних настільних ігор у казино. Її популярність пояснюється простими правилами та можливістю грати як у симуляторі так і з реальним дилером [1].

Конструкція гри доволі проста: на обертовому колесі розміщено 36 секторів, пофарбованих у червоний і чорний кольори, кожен з яких має свій номер, а також один зелений сектор — «зеро». Під час гри кулька запускається у зворотному напрямку до обертання колеса та випадково зупиняється на одному з секторів [2].

Ставки можна зробити на наступні категорії:

- червоне/чорне;
- парне/непарне;
- 1 до 18/19 до 36;
- 1-а дюжина/2-а дюжина/3-а дюжина;
- 1-а колонка/2-а колонка/3-а колонка.

Звісно ж зробити ставку також можна і на один сектор з числом, але ймовірність випадання є значно нижчою, адже кількість можливих варіантів

досить велика. На полі 37 секторів – від 0 до 36, де кожен має теоретичну ймовірність випадіння:  $P = \frac{1}{37} = 0,027 = 2,7\%$ .

З цього маємо, що наступні категорії мають ймовірність випадіння:

- червоне/чорне:  $18/37 \approx 48,64\%$ ;
- парне/непарне:  $18/37 \approx 48,64\%$ ;
- 1 до 18/19 до 36:  $18/37 \approx 48,64\%$ ;
- 1-а дюжина/2-а дюжина/3-я дюжина:  $12/37 \approx 32,43\%$ ;
- 1-а колонка/2-а колонка/3-я колонка:  $12/37 \approx 32,43\%$ .

Для аналізу поточної гри буде використовуватись накопичення історії результатів обертань, обчислення частоти випадіння відповідних категорій ставок. Фактична ймовірність  $P(i)$  обчислюється за формулою (1.1):

$$P(i) = \frac{N_i}{N}, \quad (1.1)$$

де  $N_i$  – кількість випадіннь певного результату;

$N$  – загальна кількість обертань.

У програмному застосунку буде реалізовано аналіз імовірностей випадіння певних категорій та підрахунок ставок згідно зі стратегії Мартінгейла.

Стратегія Мартінгейла – це метод управління ставками, спрямований на компенсацію програшів. Її суть полягає в подвоєнні ставки після кожного програного обертання рулетки. Теоретично, при виграші гравець повертає всі попередні втрати та отримує прибуток, еквівалентний початковій ставці [3].

Тобто принцип стратегії Мартінгейла такий:

- гравець обирає початкову ставку;
- якщо ставка програє, гравець подвоює розмір наступної ставки і продовжує ставити на той самий результат, поки не виграє;

– при виграші отримує прибуток, який компенсує всі попередні програші та приносить чистий прибуток. Після цього цикл починається заново з початкової ставки, але вже на іншу категорію.

У нашому випадку в ПЗ розрахунок ймовірності наприклад для категорії червоне/чорне буде за формулою (1.2) [4]:

$$P = 1 - \left(1 - \frac{18}{37}\right)^n = 1 - \left(\frac{19}{37}\right)^n, \quad (1.2)$$

де  $n$  – кількість спроб.

## **1.2 Огляд існуючих програмних рішень**

### **1.2.1 Сайт «Casinobot.net»**

Одним із відомих програмних рішень є сервіс «Аналізатор рулетки» на сайті «Casinobot.net» (рис. 1.1). Цей ресурс спеціалізується на тематиці онлайн-казино та азартних ігор, надаючи користувачам рейтинги, огляди, поради від експертів і гравців, а також безкоштовне програмне забезпечення для підвищення шансів на виграш [5].

Зазначений інструмент повністю автоматизує збір та обробку статистики гри в рулетку на будь-якому онлайн-казино. На основі введених даних програма генерує прогнози щодо результатів ставок [6].

Уся статистична інформація аналізується відповідно до теорії ймовірностей. На її основі формується математична модель гри, яка дозволяє передбачити результат наступного обертання колеса. Використання такого аналізу в поєднанні зі стратегічними підходами до гри значно підвищує ефективність ставок [6].



Рисунок 1.1 – Приклад застосунку «Аналізатор рулетки» платформи «Casinobot.net» [6]

1.2.2 Сайт «Roulette Predictor»

«Roulette Predictor» надає користувачам онлайн-інструмент для аналізу результатів гри в рулетку (рис. 1.2).

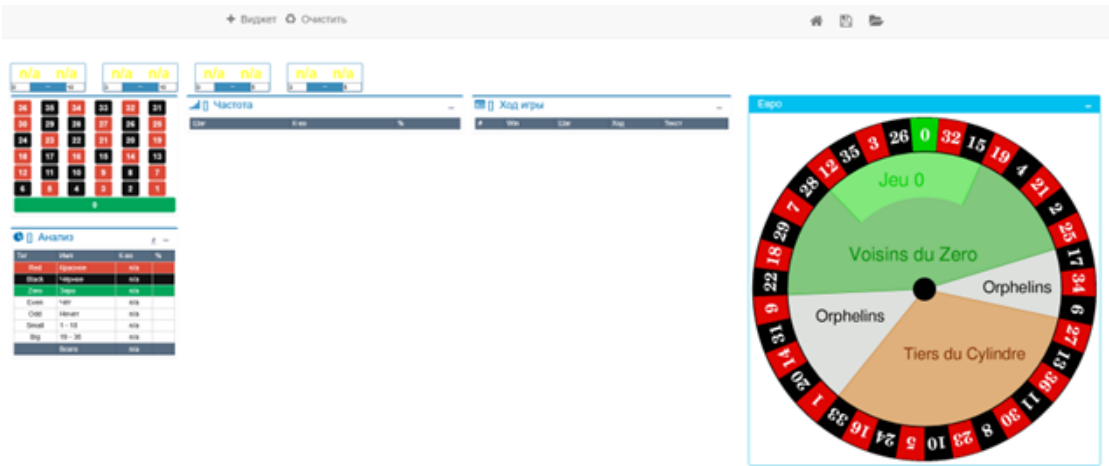


Рисунок 1.2 – Приклад застосунку «Roulette Predictor» [7]

Цей сервіс дозволяє вводити числа, які випадали під час попередніх обертань колеса, та отримувати статистичну обробку даних, що може виявити певні закономірності у грі.

Проект має відкритий вихідний код, розміщений у репозиторії на платформі GitHub. Це означає, що будь-хто може ознайомитися з принципами роботи інструменту, перевірити використані алгоритми або адаптувати код під власні потреби.

1.2.3 Мобільний додаток «BP Roulette Analyzer»

Мобільний застосунок «BP Roulette Analyzer» для Android (рис. 1.3).

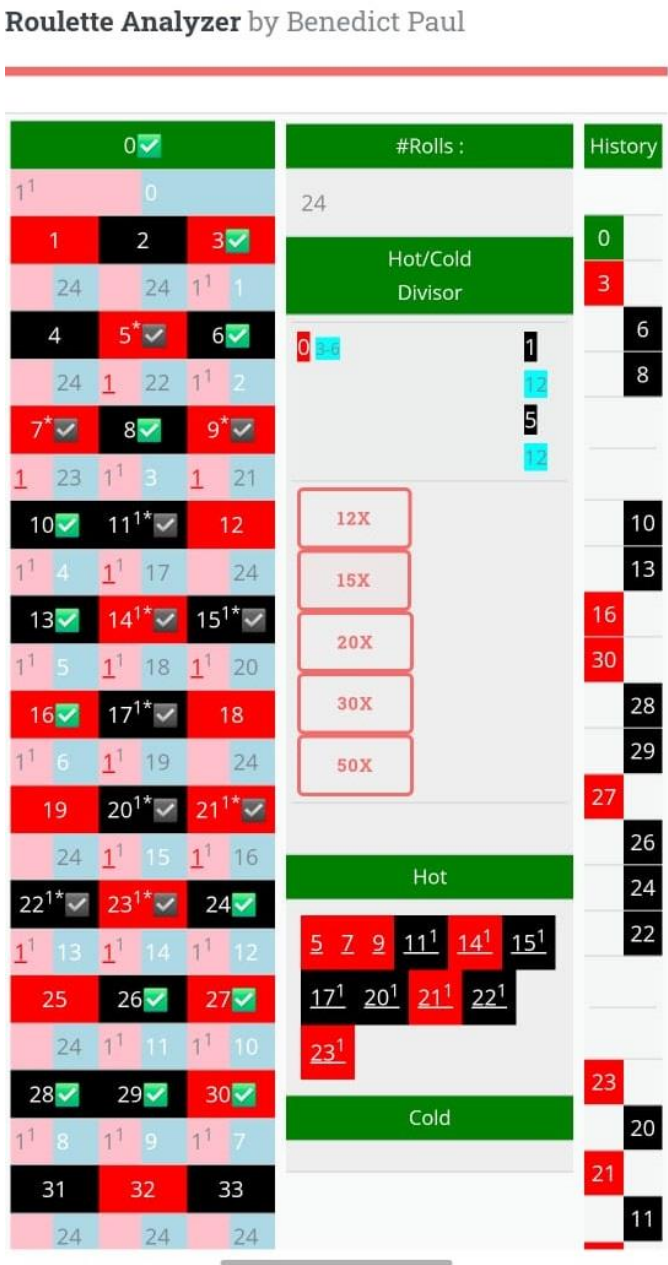


Рисунок 1.3 – Приклад застосунку «BP Roulette Analyzer» [8]

Цей застосунок дає змогу фіксувати кожен результат гри та формувати звіти, які рідко зустрічаються в аналогічних програмах. Наприклад, він відображає, скільки разів випадало гаряче число, а також скільки разів кожне число не з'являлося [8].

Окрім цього, додаток показує, коли гаряче число перейшло в категорію холодного — цю інформацію гравці використовують для прийняття рішень щодо ставок [8].

Інтерфейс програми є простим і зручним для користувача, однак функціональні можливості залишаються обмеженими. Крім того, додаток є платним, що може стати перешкодою для деяких користувачів.

### **1.3 Потреба програмних рішень**

Наявні мобільні або настільні додатки переважно платні, мають складний інтерфейс або орієнтовані на іншу функціональність, наприклад, на симуляцію гри, а не на аналіз вже існуючих результатів. Це створює потребу у створенні спеціалізованого програмного забезпечення, яке дозволить користувачеві вносити дані про фактичні результати обертань колеса фортуни та отримувати інформацію щодо ймовірностей.

Таке ПЗ має забезпечити:

- зручне введення даних;
- обчислення поточних ймовірностей випадіння кожної категорії;
- обчислення статистики гри;
- реалізація стратегії для визначення найперспективнішої ставки;
- візуальне представлення історії введених чисел;
- функції збереження та завантаження гри;
- інтуїтивно зрозумілий інтерфейс, не перевантажений зайвими функціями.

Розробка цього ПЗ дає змогу не лише спростити процес аналізу ймовірностей у грі, але й забезпечити більш гнучке управління ставками

завдяки автоматичному підрахунку найбільш імовірних результатів на основі вже наявних даних. Це мінімізує вплив людського фактору та помилок при ручних розрахунках.

Таким чином, створена система дозволяє кожному користувачеві, навіть без математичної чи технічної підготовки, з легкістю аналізувати ймовірності в рулетці та приймати обґрунтовані рішення щодо ставок.

#### 1.4 Порівняння існуючих аналогів

Виконаємо порівняння існуючих аналогів за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). Серед критеріїв виділимо наступні:

- збереження результатів: можливість зберігати історію обертань рулетки для подальшого аналізу
- зручність: простий та інтуїтивний інтерфейс, зручний для користувача;
- функціональність: наявність корисних функцій для аналізу;
- рахунок ставок: підрахунок рекомендованої ставки на категорію на основі попередніх даних;
- безкоштовність: можливість користуватись програмним забезпеченням без оплати.

Таблиця 1.1 – Порівняння аналогів

| Критерій порівняння    | Аналізатор рулетки (Casinobot.net) | Roulette Predictor | BP Roulette Analyzer |
|------------------------|------------------------------------|--------------------|----------------------|
| Збереження результатів | -                                  | +                  | +                    |
| Зручність              | +                                  | -                  | -                    |
| Функціональність       | +                                  | +                  | +                    |
| Рахунок ставок         | -                                  | -                  | -                    |
| Безкоштовність         | +                                  | +                  | -                    |

Як можемо бачити з таблиці 1.1, у кожного з перелічених сервісів є ряд переваг та недоліків. Серед спільного – це функціональність. Кожен з сервісів виконує свої поставленні функції, однак кожен з них не має функції підрахунку ставок, що є важливим фактором зручності для користувача. Всі аналоги потребують вдосконалення зручності, доступності та гнучких налаштувань.

## **1.5 Постановка завдання**

Програма має надати користувачеві можливість вводити результати гри, переглядати статистику за певними критеріями, а також отримувати прогнозовану ймовірність випадіння окремих секторів.

Отже актуальною задачею залишається розробка програмного забезпечення для аналізу випадковості у грі колесі фортуни, що автоматизує обробку результатів та надає користувачеві корисні аналітичні дані.

### **1.5.1 Функціональність системи**

Система має реалізовувати наступні функціональні вимоги:

- введення результатів гри;
- автоматичну класифікацію кожного веденого числа за категоріями;
- обчислення ймовірностей появи наступного значення для кожної категорії з урахуванням серій повторів;
- визначення категорії з найвищою ймовірністю для наступного обертання;
- автоматичний підрахунок ставок згідно зі стратегією Мартінгейла на основі послідовності втрат;
- збереження історії введених чисел у БД;
- завантаження історії збереженої гри з БД;
- скидання всіх даних і повернення до початкового стану гри.

### 1.5.2 Вимоги до інтерфейсу

Інтерфейс ПЗ повинен бути інтуїтивно зрозумілим та функціональним. Система повинна відповідати таким вимогам:

- зрозуміле розміщення елементів: кнопки, таблиці та поля вводу мають бути логічно структуровані та згруповані за призначенням;
- мінімалістичний дизайн: відсутність зайвих елементів, сучасне темне оформлення для зниження втомлюваності очей;
- миттєве оновлення інформації: результати, ймовірності та ставки оновлюються автоматично одразу після введення числа;
- наявність інтерактивних елементів управління: кнопки для додавання числа, прокрутки історії, запуску нової гри, збереження та завантаження даних;
- інформативні таблиці: таблиці статистики та ймовірностей мають відображати актуальні дані у зручному форматі;
- поле введення з перевіркою: введення лише допустимих значень (від 0 до 36), з повідомленням про помилки.

### 1.6 Висновки за розділом 1

У першому розділі проведено детальний аналіз предметної області (колеса фортуни). Розглянуто основні правила гри, структуру ставок та теоретичні ймовірності випадіння різних категорій. Також описано методику обчислення фактичних ймовірностей на основі історії обертань. Особливу увагу приділено стратегії Мартінгейла, яка буде інтегрована в програмний застосунок для автоматизованого підрахунку ставок. Після цього було проаналізовано існуючі програмні рішення, де ми порівняли можливості різних платформ та додатків, проаналізовано всі переваги та недоліки, зроблено певні висновки. Було визначено потребу програмного рішення та поставлено завдання зі своїми функціональними вимогами та вимогами до інтерфейсу.

## 2 МАТЕРІАЛИ І МЕТОДИ

У цьому розділі наведено обґрунтування вибору основних інструментів для розробки програмного забезпечення. Проведено порівняння популярних мов програмування, на основі чого обрано найбільш оптимальну мову для реалізації проєкту. Далі здійснено аналіз середовищ розробки, в результаті чого обрано середовище, що найкраще відповідає вимогам зручності та ефективності.

### 2.1 Вибір мови програмування

Для розробки програмного забезпечення для аналізу випадковості у колесі фортуни було вибрано мову програмування Python.

Python – це високорівнева мова програмування загального призначення, що відзначається своєю простотою, зрозумілим синтаксисом та широкими можливостями застосування. Завдяки легкості у вивченні, великій кількості готових бібліотек і підтримці різних напрямів розробки, Python здобула популярність серед як початківців, так і досвідчених програмістів [9].

Однією з ключових переваг Python є її чистий і логічний синтаксис, який значно спрощує написання та супровід коду. Мова чудово підходить для вирішення задач у різних галузях: від веб-розробки та обробки даних до штучного інтелекту й автоматизації. Кросплатформність і динамічна типізація також надають Python гнучкість і зручність у використанні [9].

Python широко використовується в таких сферах, як включаючи веб-розробку, аналіз даних, машинне навчання, штучний інтелект, автоматизацію, наукові обчислення тощо. Його адаптивність та надійні бібліотеки роблять цю мову одним із найкращих інструментів для реалізації різноманітних програмних проєктів [8].

Вибір мови програмування представимо у вигляді табл. 2.1 виконавши порівняння за наступними критеріями:

- простота синтаксису;
- швидкість розробки;
- динамічна типізація;
- підтримка різних типів моделей програмування;
- кросплатформність.

Ці критерії були обрані тому, що яскраво демонструють відмінності між переліченими мовами програмування. В Python нема необхідності використовувати крапку з комою та фігурні дужки в програмі, на відміну від Java та C++, які покажуть синтаксичну помилку. Для написання однієї і тієї ж програми на Python потрібно менше рядків, ніж на Java та C++. Python є динамічно типізованою мовою, що означає, що потрібно лише присвоїти значення змінній під час виконання програми та інтерпретатор сам визначить тип даних, на відміну від Java та C++, де потрібно явно вказувати тип даних. Python підтримує різні типи моделей програмування, як імперативне, об'єктно-орієнтоване та процедурне програмування, на відміну від Java, яка повністю базується на моделях програмування на основі об'єктів і класів [10].

Таблиця 2.1 – Порівняння мов програмування

| Критерій порівняння                          | Python | Java | C++ |
|----------------------------------------------|--------|------|-----|
| Простота синтаксису                          | +      | -    | -   |
| Швидкість розробки                           | +      | -    | -   |
| Динамічна типізація                          | +      | -    | -   |
| Підтримка різних типів моделей програмування | +      | -    | +   |
| Кросплатформність                            | +      | +    | +   |

На основі проведеного порівняння за обраними критеріями можна зробити висновок, що саме мова програмування Python є оптимальним вибором для реалізації ПЗ. Python має простий та зрозумілий синтаксис, потребує менше

рядків коду, є динамічно типізованою мовою програмування та підтримує різні типи моделей програмування. Завдяки високій популярності та активній спільноті, розробник завжди має доступ до великої кількості документації, прикладів та готових рішень. Окрім того, Python є кросплатформною мовою, що дозволяє запускати програму на різних операційних системах без необхідності адаптації коду.

## 2.2 Вибір середовища розробки

Для ефективної реалізації програмного забезпечення важливо обрати найбільш зручне IDE, яке відповідатиме вимогам проєкту за функціональністю, стабільністю та зручністю. Серед найпопулярніших середовищ для мови програмування Python є PyCharm, Visual Studio Code та Spyder.

PyCharm — інтегроване середовище розробки від JetBrains, яке є одним із найпопулярніших інструментів для програмістів Python завдяки широкому набору функцій, що значно спрощують створення програмного забезпечення [11].

Visual Studio Code — легкий і універсальний текстовий редактор від Microsoft, який здобув популярність серед розробників Python через свою гнучкість, підтримку численних розширень та інтегровані інструменти [11].

Spyder — IDE, розроблене спеціально для аналітиків даних та інженерів, з підтримкою операційних систем Linux, macOS і Windows [11].

Вибір середовища розробки для Python представимо у вигляді табл. 2.2 виконавши порівняння за наступними критеріями:

- автозаповнення коду;
- підтримка Git;
- плагіни та розширення;
- швидкість роботи;
- безкоштовність.

Таблиця 2.2 – Порівняння середовищ

| Критерій порівняння   | PyCharm Community Edition | Visual Studio Code | Spyder |
|-----------------------|---------------------------|--------------------|--------|
| Автозаповнення коду   | +                         | +                  | +      |
| Підтримка Git         | +                         | +                  | -      |
| Плагіни та розширення | +                         | +                  | -      |
| Швидкість роботи      | +                         | -                  | +      |
| Безкоштовність        | +                         | +                  | +      |

На основі проведеного порівняння за обраними критеріями можна зробити висновок, що саме IDE PyCharm є оптимальним вибором для написання коду. У порівнянні з ним, Spyder не має підтримки Git та плагінів і розширень для більш зручної розробки. Visual Studio Code навпаки, потрапляє майже за всіма критеріями, але у порівнянні з PyCharm менш швидкий.

### 2.3 Висновки за розділом 2

У другому розділі пояснювальної записки було здійснено вибір мови програмування для розробки ПЗ, проведено порівняльний аналіз мов програмування, серед яких за основними критеріями (простота синтаксису, швидкість розробки, динамічна типізація, підтримка різних моделей програмування та кросплатформність) було обрано Python як найоптимальнішу для виконання поставленого завдання.

Далі було проведено порівняння середовищ розробки, де враховувались такі характеристики як автозаповнення коду, підтримка Git, плагіни та розширення, швидкість роботи та безкоштовність. У результаті порівняння середовищ PyCharm Community Edition, Visual Studio Code та Spyder було обрано PyCharm, який забезпечує комфортну розробку на всіх етапах створення програмного продукту.

## 3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

### 3.1 Алгоритм роботи системи

В ході проєктування програмного забезпечення для аналізу випадковості у колесі фортуни було розроблено:

- функціональну схему розробленої програми (рис. 3.1), яка окреслює робочий процес системи;
- структурну схему системи (рис. 3.2), яка ілюструє організацію компонентів системи;

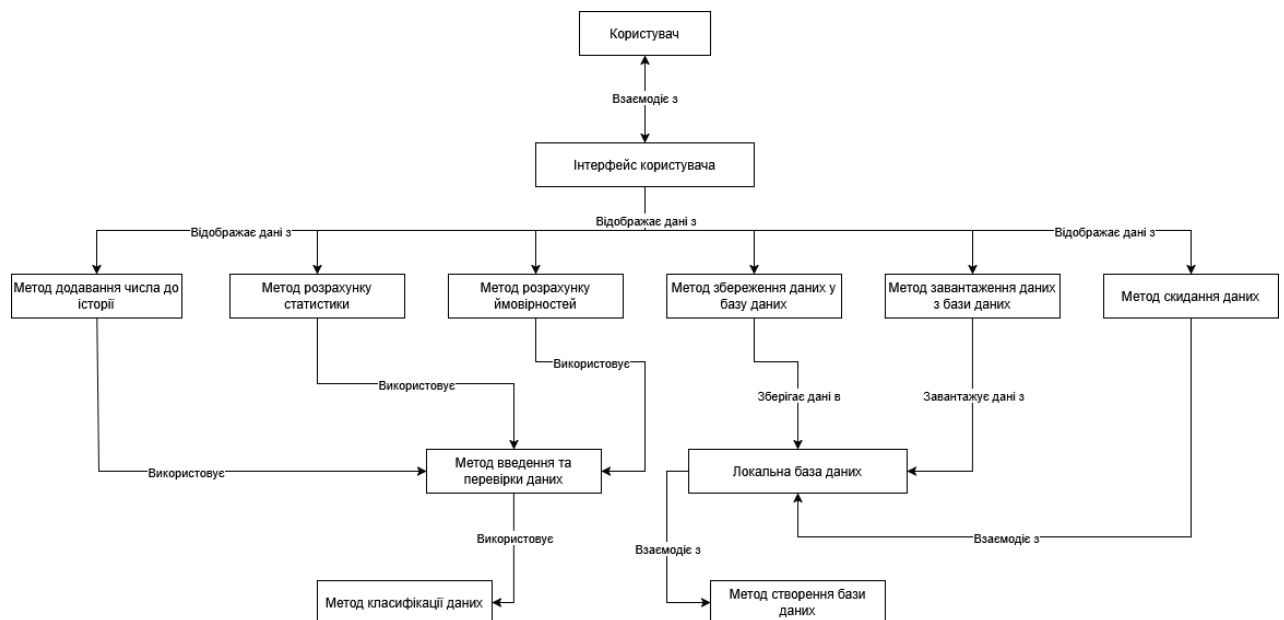


Рисунок 3.1 – Функціональна схема розробленої програми

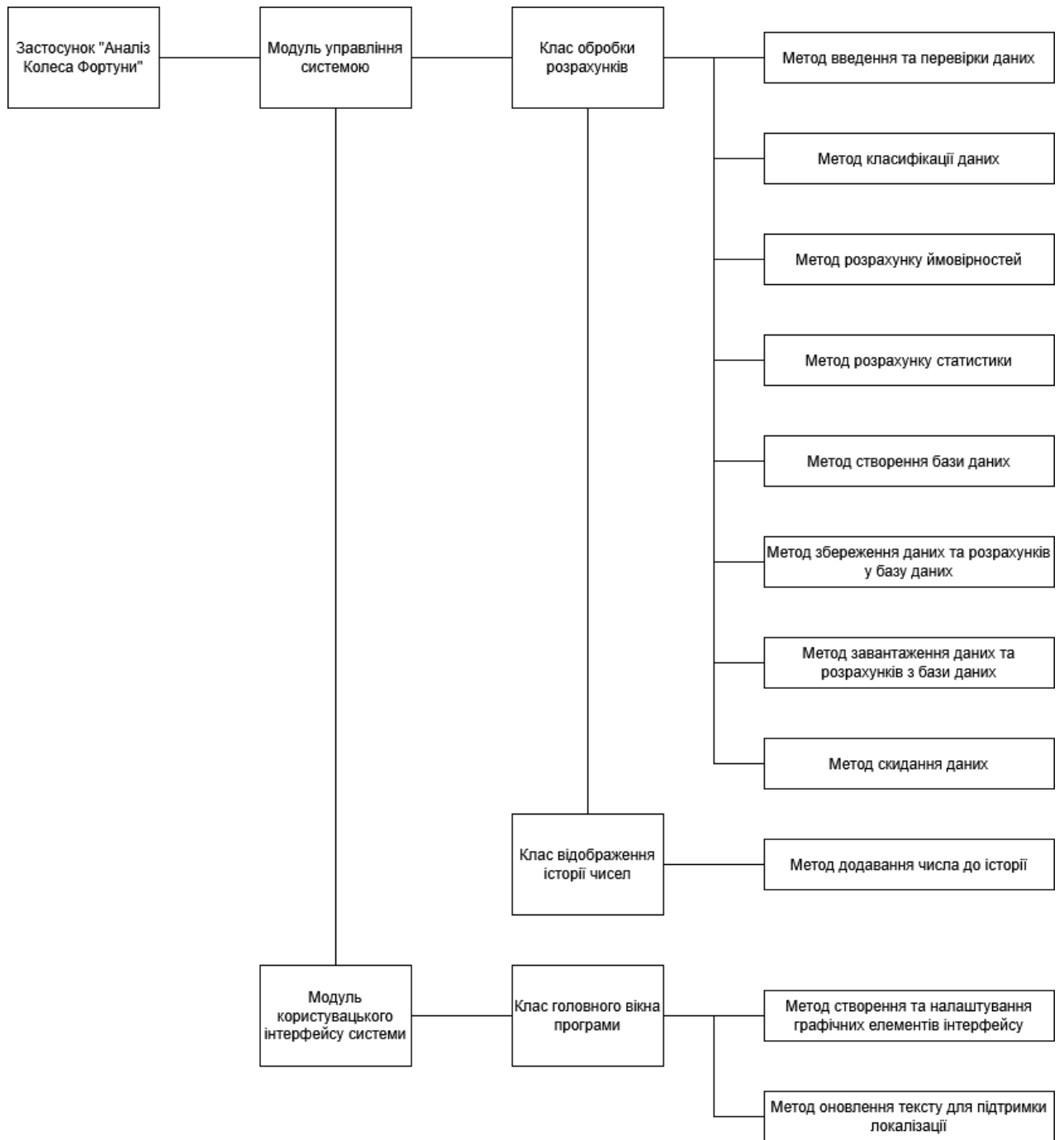


Рисунок 3.2 – Структурна схема системи

Також в ході проєктування було створено діаграму класів (рис. 3.3), яка відображає зв'язки між компонентами програми за допомогою нотацій мови UML.

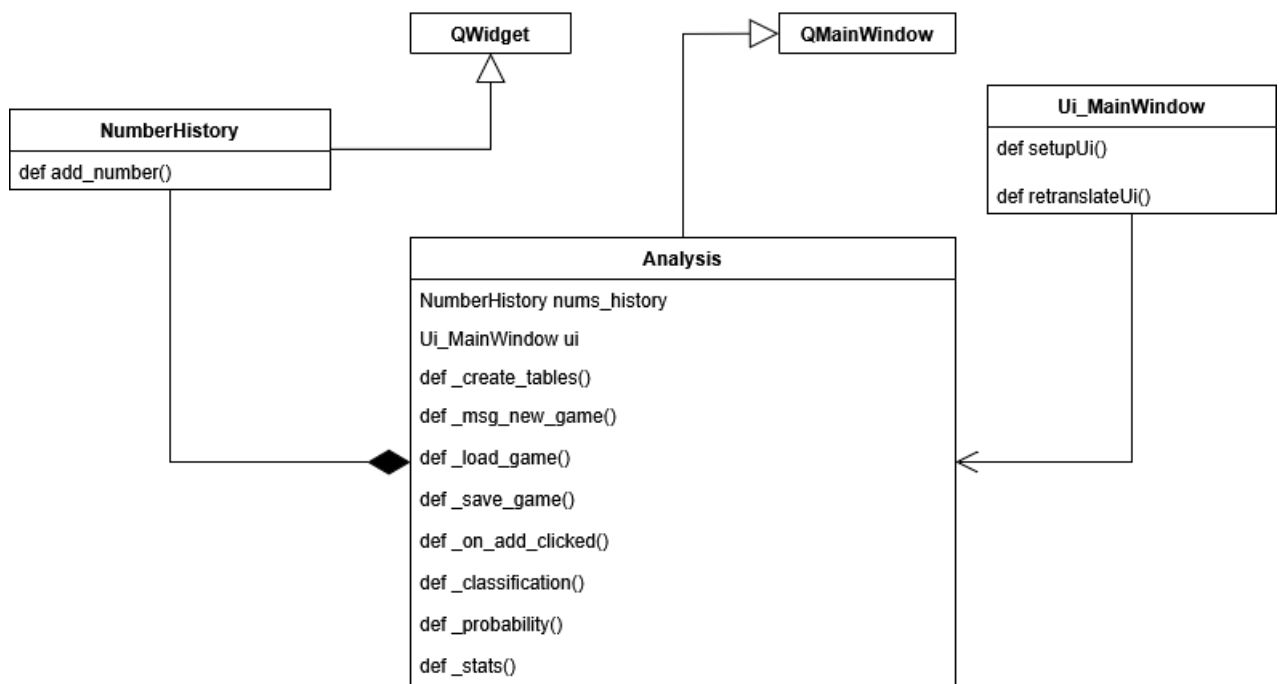


Рисунок 3.3 – Діаграма класів

Діаграма класів відображає три основні класи, пов'язані спеціальними відношеннями: **Analysis**, **NumberHistory** та **Ui\_MainWindow**. Клас **Analysis** є центральним і пов'язаний з класом **NumberHistory** через зв'язок «Композиція», оскільки екземпляри **NumberHistory** не можуть існувати окремо від **Analysis**. Клас **Analysis** успадковує клас **QMainWindow** з бібліотеки **PyQt6**, що забезпечує функціонал головного вікна програми. Клас **NumberHistory** успадковує клас **QWidget**, що дозволяє створювати віджети для відображення історії чисел. Клас **Ui\_MainWindow** пов'язаний з **Analysis** через спрямовану бінарну асоціацію, оскільки він відповідає за завантаження та відображення користувацького інтерфейсу, який використовує методи класу **Analysis** для оновлення даних.

Клас **Analysis** є ключовим компонентом програми, що реалізує основну логіку аналізу ймовірностей та обробки введених даних. Він містить методи для обчислення ймовірностей, збереження та завантаження даних, а також оновлення статистики. Перелік основних методів класу наведено у таблиці 3.1.

Таблиця 3.1 – Методи класу Analysis

| Метод                              | Опис                                                                                                                                             |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>def _create_tables()</code>  | Створює бази даних SQLite.                                                                                                                       |
| <code>def _msg_new_game()</code>   | Відображає діалогове вікно для підтвердження початку нової гри та скидає всі дані.                                                               |
| <code>def _load_numbers()</code>   | Завантажує збережену історію чисел та розрахунки з бази даних.                                                                                   |
| <code>def _save_numbers()</code>   | Зберігає поточний стан гри у базу даних                                                                                                          |
| <code>def _on_add_clicked()</code> | Перевіряє введене число викликає методи для обчислення статистики та ймовірностей.                                                               |
| <code>def _classification()</code> | Створює словник, який зберігає в собі всі можливі числа та категорії до яких вони належать, для подальшого визначення ведених даних користувачем |
| <code>def _probability()</code>    | Створює словник, який зберігає в собі всі можливі числа та категорії до яких вони належать, для подальшого визначення ведених даних користувачем |
| <code>def _stats()</code>          | Обчислює статистику частоти випадання категорій у відсотках і кількості.                                                                         |

Клас Analysis створює таблиці для відображення ймовірностей (`table_prob`) та статистики (`table_stats`), а також керує взаємодією з користувачем через користувацький інтерфейс.

Клас `NumberHistory` відповідає за динамічне відображення історії введених користувачем чисел. Кожне число відображається із відповідним кольором до його категорії. Перелік методів класу наведено у таблиці 3.2.

Таблиця 3.2 – Методи класу `NumberHistory`

| Метод                         | Опис                                                       |
|-------------------------------|------------------------------------------------------------|
| <code>def add_number()</code> | Додає число до історії із відповідним кольором і позицією. |

Клас використовує горизонтальну прокрутку для відображення великої кількості чисел, забезпечуючи автоматичне прокручування до останнього доданого числа.

Клас `Ui_MainWindow` відповідає за створення та налаштування користувацького інтерфейсу програми. Він відтворює всі елементи керування, включаючи текстові поля, кнопки та таблиці, а також задає стилізацію через CSS. Методи класу наведено у таблиці 3.3.

Таблиця 3.3 – Методи класу `Ui_MainWindow`

| Метод                            | Опис                                                           |
|----------------------------------|----------------------------------------------------------------|
| <code>def setupUi()</code>       | Створює та налаштовує графічні елементи інтерфейсу             |
| <code>def retranslateUi()</code> | Оновлює текстові елементи інтерфейсу для підтримки локалізації |

Клас `Ui_MainWindow` забезпечує відображення результатів обчислень, виконаних класом `Analysis`, у таблицях `table_prob` і `table_stats`, а також обробляє взаємодію користувача з елементами керування.

### 3.2 Проектування користувацького інтерфейсу

Для забезпечення зручної взаємодії користувача з програмним забезпеченням розроблено єдину форму головного вікна, яка містить усі необхідні елементи керування (рис. 3.4). Інтерфейс створено з використанням бібліотеки PyQt6 та середовища Qt Designer, що забезпечує мінімалістичний і інтуїтивно зрозумілий дизайн.

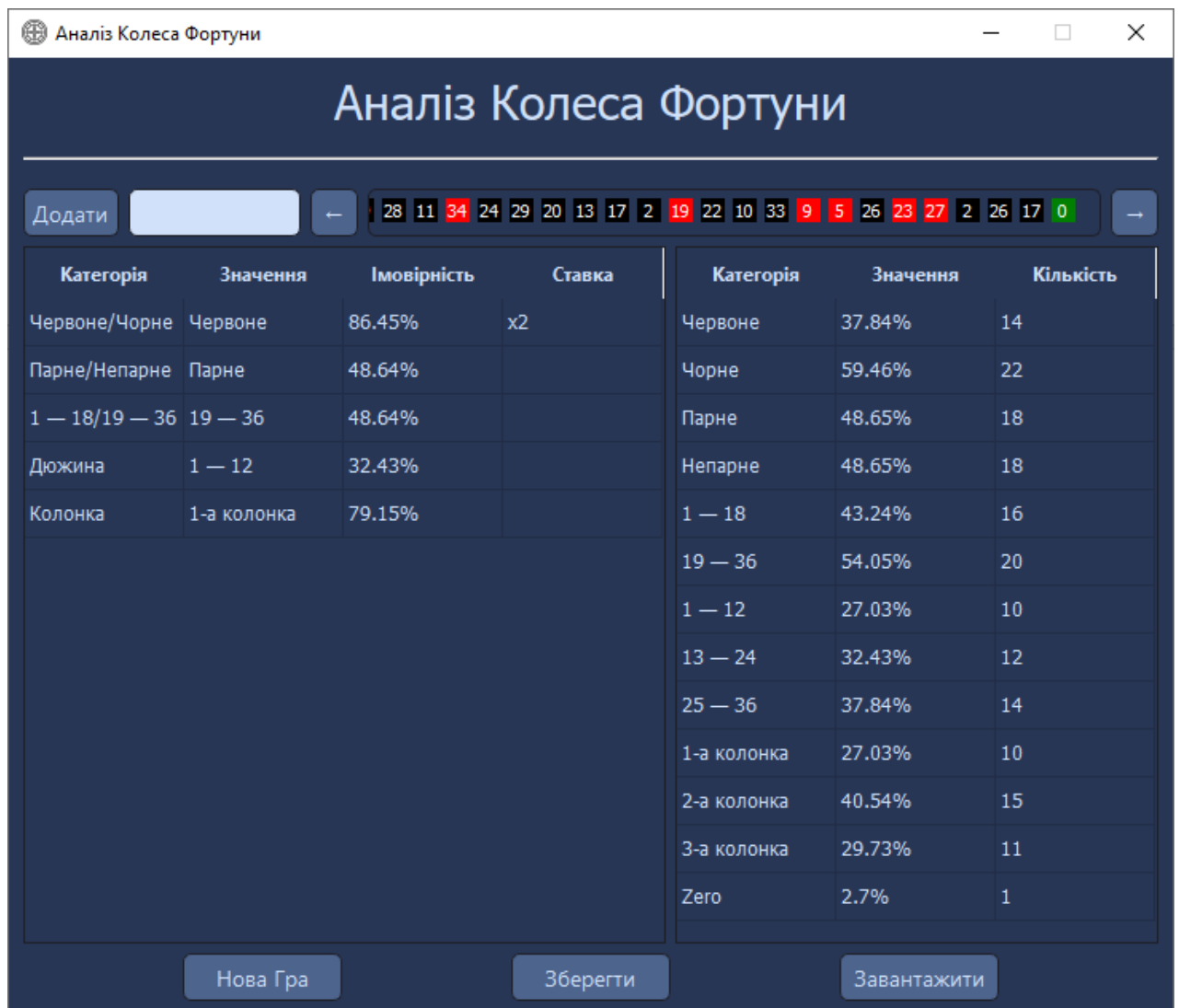


Рисунок 3.4 – Головне вікно програми

Головне вікно програми оформлено в темно-синій кольоровій гамі з контрастними текстовими елементами для забезпечення читабельності. Основні компоненти інтерфейсу включають:

Заголовок програми (QLabel): У верхній частині вікна розташовано надпис «Аналіз Колеса Фортуни». Цей елемент чітко вказує на призначення програми.

Поле введення чисел (QLineEdit): Текстове поле для введення результатів обертань колеса фортуни. Користувач може активувати введення числа натисканням клавіші Enter завдяки налаштуванню ярлика.

Кнопка «Додати» (QPushButton): Розташована поруч із полем введення, дозволяє додати введене число до історії обертань. При натисканні викликається метод, який перевіряє коректність введеного числа (ціле число в межах 0–36) і оновлює статистику та ймовірності. У разі помилки (наприклад, некоректне число) відображається попередження через QMessageBox.

Область історії чисел (QScrollArea): Динамічна область для відображення введених чисел, реалізована через клас NumberHistory. Кожне число відображається як QLabel із фоновим кольором, що відповідає його категорії кольору (зелений для 0, червоний для червоних чисел, чорний для чорних). Область підтримує горизонтальну прокрутку, керовану кнопками «←» і «→» (QPushButton), які змінюють значення горизонтальної смуги прокрутки на 50 пікселів.

Таблиця ймовірностей (QTableWidget): Таблиця з 5 рядками і 4 стовпцями відображає прогнозовані ймовірності для категорій: «Червоне/Чорне», «Парне/Непарне», «1–18/19–36», «Дюжина» та «Колонка». Стовпці включають назву категорії, значення (наприклад, «Червоне» чи «1-а колонка»), ймовірність у відсотках та рекомендовану ставку (наприклад, «x2»).

Таблиця статистики (QTableWidget): Таблиця з 13 рядками і 3 стовпцями відображає статистику введених чисел за категоріями: «Червоне», «Чорне», «Парне», «Непарне», «1–18», «19–36», «1–12», «13–24», «25–36», «1-а колонка», «2-а колонка», «3-я колонка» та «Zero». Для кожної категорії вказується відсоток і кількість випадань.

Кнопки керування грою:

- «Нова Гра» (QPushButton): викликає метод, який запитує підтвердження через QMessageBox і скидає всі дані, створюючи нову гру;
- «Зберегти» (QPushButton): викликає метод, який зберігає історію чисел у файл;
- «Завантажити» (QPushButton): викликає метод, який зчитує числа з файлу, оновлює історію та перераховує статистику і ймовірності.

### **3.3 Висновки за розділом 3**

У третьому розділі пояснювальної записки було описано процес проєктування програмного забезпечення для аналізу випадковості у колесі фортуни. Було наведено функціональну схему розробленої програми та структурну схеми системи. Було розроблено та описано класи Analysis, NumberHistory та Ui\_MainWindow, що є основою програми, а також визначено залежності та зв'язки між ними за допомогою діаграми класів UML. Наведено детальний опис методів кожного класу та їх призначення.

Розроблено користувацький інтерфейс із використанням бібліотеки PyQt6 та середовища Qt Designer, що забезпечує зручну взаємодію з програмою та описано призначення всіх елементів керування. Програмний застосунок здатний обробляти введені користувачем результати обертань колеса фортуни, зберігати їх у базі даних та завантажувати для подальшого аналізу.

## **4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ**

### **4.1 Призначення й умови застосування програми**

Програмний застосунок призначений для аналізу випадковості колеса фортуни шляхом обробки введених користувачем результатів обертань. Програма автоматично класифікує кожне ведене число за категорією, обчислює статистику за кожною категорією у вигляді кількості випадіннь та відсоток від загальної кількості введених чисел, ПЗ обчислює ймовірності появи наступного значення для кожної категорії з урахуванням серії повторів, визначає категорії з найвищою ймовірністю для наступного обертання, автоматично займається підрахунком ставок згідно зі стратегією Мартінгейла, може зберігати та завантажувати історію гри.

Умови застосування:

- операційна система: Windows 10/11, Linux (Ubuntu 20.04 або новіший) або macOS (10.15 або новіший);
- процесор: із частотою від 1,5 ГГц;
- оперативна пам'ять: 4 Гб;
- диск: 100 Мб вільного місця на диску

Програма підходить для користувачів без спеціальної підготовки завдяки інтуїтивному інтерфейсу та мінімальним вимогам до апаратного забезпечення.

### **4.2 Вимоги до інтерфейсу**

Інтерфейс програми розроблено зручним та інтуїтивно зрозумілим. Основні вимоги до інтерфейсу:

- зрозуміле розміщення елементів: усі елементи керування розташовані логічно, з чітким поділом функціональних зон.

- мінімалістичний дизайн: темно-синя кольорова гама з контрастними текстовими елементами забезпечує читабельність і зменшує втому очей. Використано стилізацію CSS для оформлення віджетів.
- миттєве оновлення інформації: при введенні нового числа, таблиці ймовірностей і статистики оновлюються автоматично, забезпечуючи актуальність даних.
- інтерактивність: підтримка введення чисел через клавішу Enter, горизонтальна прокрутка історії чисел кнопками «←» і «→», діалогові вікна для підтвердження дій.
- обробка помилок: у разі введення некоректних даних (наприклад, числа поза діапазоном 0–36) відображаються повідомлення через QMessageBox із чітким поясненням помилки.

Інтерфейс реалізовано за допомогою бібліотеки PyQt6 і середовища Qt Designer, що забезпечує кросплатформну сумісність і гнучкість налаштувань.

### **4.3 Умови виконання програми**

Для успішного виконання програми необхідно виконати наступні умови:  
встановлення Python: наявність Python 3.8 або новішого;

- встановлення бібліотеки PyQt6: виконання команди `pip install PyQt6` у командному рядку Windows або терміналі IDE;
- наявність вихідних файлів: файли ПЗ мають бути розташовані в одній директорії.
- запуск програми: виконання запуску `main.py` у командному рядку Windows або через IDE.
- введення даних: користувач вводить цілі числа від 0 до 36 через текстове поле, некоректні дані (літери, дробові числа, числа поза діапазоном) відхиляються з відповідним повідомленням.

#### 4.4 Результати тестування

Нижче наведено результати тестування програми у вигляді тест-кейсів, розроблених на основі ключових сценаріїв використання (рис. 4.1 – 4.3).

| Назва                                                                         |                                                | Введення валідного числа                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                         |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| <i>Перевірка коректності додавання до історії та оновлення таблиць числа:</i> |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| 14                                                                            |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| Передумови                                                                    |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| Запустити програму                                                            |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| №                                                                             | Кроки                                          | Очікуваний результат                                                                                                                                                                                                                                                                                                                                                          |
| 1                                                                             | У текстове поле ввести валідне число (14)      | Число буде відображено у текстовому полі                                                                                                                                                                                                                                                                                                                                      |
| 2                                                                             | Натиснути на кнопку "Додати" або клавішу Enter | Валідне число (14) додано до історії чисел у вигляді червоної мітки;<br>Таблиця ймовірностей оновлена, показує ймовірність 48.64% на категорії "Чорне", "Непарне" та "19 – 36" та 32.43% на категорії "1 – 12" і "1-а колонка";<br>Таблиця статистики оновлена, відображає значення 100% та кількість 1 для категорій "Червоне", "Парне", "1 – 18", "13 – 24", "2-а колонка". |
|                                                                               |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
|                                                                               |                                                |                                                                                                                                                                                                                                                                                                                                                                               |

Рисунок 4.1 – Тест-кейс №1: Введення валідного числа

| Назва                                                                         |                                                | Введення невалідного числа                                                                                                                    |
|-------------------------------------------------------------------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                         |                                                |                                                                                                                                               |
| <i>Перевірка коректності додавання до історії та оновлення таблиць числа:</i> |                                                |                                                                                                                                               |
| -5                                                                            |                                                |                                                                                                                                               |
| Передумови                                                                    |                                                |                                                                                                                                               |
| Запустити програму                                                            |                                                |                                                                                                                                               |
| №                                                                             | Кроки                                          | Очікуваний результат                                                                                                                          |
| 1                                                                             | У текстове поле ввести валідне число           | Число буде відображено у текстовому полі                                                                                                      |
| 2                                                                             | Натиснути на кнопку "Додати" або клавішу Enter | Буде відображено повідомлення "Помилка. Введіть число від 0 до 36!";<br>Текстове поле буде очищено;<br>Історія чисел і таблиці не змінюються. |
|                                                                               |                                                |                                                                                                                                               |
|                                                                               |                                                |                                                                                                                                               |

Рисунок 4.2 – Тест-кейс №2: Введення невалідного числа

| Назва                                                                           |                                                                                              | Збереження, скидання та завантаження бази даних гри                                                               |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                           |                                                                                              |                                                                                                                   |
| Перевірка функції збереження, скидання та завантаження бази даних гри з даними: |                                                                                              |                                                                                                                   |
| - 0                                                                             |                                                                                              |                                                                                                                   |
| - 12                                                                            |                                                                                              |                                                                                                                   |
| - 26                                                                            |                                                                                              |                                                                                                                   |
| Передумови                                                                      |                                                                                              |                                                                                                                   |
| Запустити програму                                                              |                                                                                              |                                                                                                                   |
| №                                                                               | Кроки                                                                                        | Очікуваний результат                                                                                              |
| 1                                                                               | У текстове поле ввести валідне число (0)                                                     | Число буде відображено у текстовому полі                                                                          |
| 2                                                                               | Натиснути на кнопку "Додати" або клавішу Enter                                               | Валідне число (0) додано до історії чисел у вигляді зеленої мітки, додано до таблиць ймовірностей та статистики   |
| 3                                                                               | У текстове поле ввести валідне число (12)                                                    | Число буде відображено у текстовому полі                                                                          |
| 4                                                                               | Натиснути на кнопку "Додати" або клавішу Enter                                               | Валідне число (12) додано до історії чисел у вигляді червоної мітки, додано до таблиць ймовірностей та статистики |
| 5                                                                               | У текстове поле ввести валідне число (26)                                                    | Число буде відображено у текстовому полі                                                                          |
| 6                                                                               | Натиснути на кнопку "Додати" або клавішу Enter                                               | Валідне число (26) додано до історії чисел у вигляді чорної мітки, додано до таблиць ймовірностей та статистики   |
| 5                                                                               | Натиснути кнопку "Зберегти"                                                                  | З'явиться вікно підтвердження дії.                                                                                |
| 6                                                                               | У вікні "Увага. Ви дійсно хочете зберегти гру?" натиснути на кнопку кнопку "Yes"             | Гра буде збережена у базі даних.                                                                                  |
| 7                                                                               | Натиснути на кнопку "Завантажити"                                                            | З'явиться вікно підтвердження дії.                                                                                |
| 8                                                                               | У вікні "Увага. Ви дійсно хочете завантажити збережений файл гри?" натиснути на кнопку "Yes" | Числа "0", "12" та "26" відновлено в історії чисел, таблицях ймовірностей та статистики.                          |
| 9                                                                               | Натиснути на кнопку "Нова гра"                                                               | З'явиться вікно підтвердження дії.                                                                                |
| 10                                                                              | У вікні "Увага. Ви дійсно хочете почати нову гру?" натиснути на кнопку "Yes"                 | Історія чисел та таблиць очищені.                                                                                 |

Рисунок 4.3 – Тест-кейс №3: Збереження, скидання та завантаження бази даних

Продовження результатів тестування у вигляді тест-кейсів (рис. 4.4).

| Назва                                               |                                                                              | Завантаження пустої бази даних гри                                                                                                    |
|-----------------------------------------------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Тема:                                               |                                                                              |                                                                                                                                       |
| <i>Перевірка завантаження пустої бази даних гри</i> |                                                                              |                                                                                                                                       |
| Передумови                                          |                                                                              |                                                                                                                                       |
| Запустити програму                                  |                                                                              |                                                                                                                                       |
| №                                                   | Кроки                                                                        | Очікуваний результат                                                                                                                  |
| 1                                                   | Натиснути на кнопку "Нова гра"                                               | З'явиться вікно підтвердження дії                                                                                                     |
| 2                                                   | У вікні "Увага. Ви дійсно хочете почати нову гру?" натиснути на кнопку "Yes" | Буде відображено помилку з повідомленням: "Помилка. Файл гри містить некоректні дані! Файл не містить дійсних чисел у діапазоні 0-36" |
| 3                                                   | Натиснути на кнопку "Завантажити"                                            | З'явиться вікно підтвердження дії                                                                                                     |
| 4                                                   | У вікні "Увага. Ви дійсно хочете почати нову гру?" натиснути на кнопку "Yes" | Буде відображено помилку з повідомленням: "Помилка при завантаженні гри: Немає збереженої історії чисел."                             |

Рисунок 4.4 – Тест-кейс №4: Завантаження пустої бази даних гри

#### 4.5 Висновки за отриманими результатами

Результати тестування підтверджують, що розроблене ПЗ успішно виконує всі заплановані функції, зокрема введення та обробку чисел, класифікацію за категоріями, обчислення статистичних показників і ймовірностей, визначення найбільш ймовірних категорій для наступного обертання, а також реалізацію стратегії підрахунку ставок за методом Мартінгейла. Інтерфейс програми виявився зручним, інтуїтивно зрозумілим і приємним у використанні, що робить програму доступною для широкого кола користувачів. Під час тестування програма продемонструвала стабільну роботу, правильну обробку помилок та коректне збереження і завантаження БД. Програма повністю відповідає технічному завданню, показала надійність у роботі та може ефективно використовуватись для аналізу випадковості колеса фортуни.

## ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було розроблено програмне забезпечення для аналізу випадковості колеса фортуни. Було проведено огляд предметної області, вивчено особливості обчислення ймовірностей для різних категорій ставок, проаналізовано існуючі програмні рішення та визначено їхні переваги й недоліки. Сформульовано вимоги до функціональності та інтерфейсу майбутнього застосунку.

Для цього було виконано ряд завдань:

- проведено аналіз предметної області та математичних основ розрахунку ймовірностей;
- здійснено огляд і порівняння аналогічних програмних рішень;
- визначено технічні вимоги, обґрунтовано вибір мови програмування, бібліотек та середовища розробки;
- реалізовано архітектуру застосунку, алгоритми обробки даних та користувацький інтерфейс.

Розроблений застосунок був реалізований мовою Python з використанням бібліотеки PyQt6 у середовищі PyCharm. Для створення графічного інтерфейсу використовувалось середовище Qt Designer та стилізація CSS.

Застосунок було протестовано на основі ключових сценаріїв: введення коректних і некоректних даних, збереження та завантаження бази даних гри, відображення ймовірностей і ставок. Результати тестування підтвердили стабільність, коректність обчислень і зручність у користуванні.

Застосунок реалізує такі функціональні вимоги:

- введення чисел з автоматичною класифікацією за категоріями;
- обчислення статистики та ймовірностей випадіння для кожної категорії;
- визначення найімовірнішої категорії для наступного обертання;
- підрахунок ставок за стратегією Мартінгейла;

- збереження та завантаження бази даних гри;
- миттєве оновлення інтерфейсу й повідомлення про помилки.

В результаті розробки реалізовано зручний і функціональний програмний засіб для аналізу випадковості колеса фортуни, який може бути корисним як для досліджень, так і для практичного використання в онлайн-іграх.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Європейська рулетка / [Електрон. ресурс]. – Режим доступу: <https://www.prointernet.in.ua/ruletka/yevropejska/>.
2. Американська, європейська та французька рулетка: їх особливості та формат гри / [Електрон. ресурс]. – Режим доступу: <https://news.liga.net/ua/all/pr/amerykanska-ievropeiska-ta-frantsuzka-ruletka-ikh-osoblyvosti-ta-format-hry>.
3. What is the Martingale system in roulette? / [Electronic resource]. – Access mode: <https://www.casino.org/roulette/strategy/martingale/>.
4. Calculating the probability to win with martingale in roulette / [Electronic resource]. – Access mode: <https://math.stackexchange.com/questions/1407772/calculating-the-probability-to-win-with-martingale-in-roulette>.
5. Що ви знайдете на сайті casinobot.net / [Електрон. ресурс]. – Режим доступу: <https://casinobot.net/>.
6. Онлайн аналізатор рулетки / [Електрон. ресурс]. – Режим доступу: <https://casinobot.net/tools/roulette-analyzer/>.
7. Roulette Predictor / [Електрон. ресурс]. – Режим доступу: <https://roulette.gsm-center.com.ua/>.
8. BP Roulette Analyzer / [Електрон. ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.bpaulenterprise.roulettefull&hl=en>.
9. Why Choose Python Over Other Programming Languages / [Electronic resource]. – Access Mode: <https://www.linkedin.com/pulse/why-choose-python-over-other-programming-languages-superiorcodelabs-oh69c>.
10. Comparison of Python with Other Programming Languages / [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/comparison-of-python-with-other-programming-languages/>.

11. Топ-5 кращих IDE для Python розробки у 2024 / [Електрон. ресурс].  
– Режим доступу: <https://university.sigma.software/best-python-ide-and-code-editors/>.

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

У роботі розглядається галузь програмного забезпечення для аналізу випадковості у колесі фортуни. Програмне забезпечення має забезпечити автоматизоване обчислення ймовірностей, ведення статистики та підрахунок ставок за заданими стратегіями.

### **A.1 Підстава для розробки**

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Розробка програмного забезпечення для аналізу випадковості у колесі фортуни», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

### **A.2 Основні вимоги до програми, що розробляється**

#### **A.2.1 Вимоги до функціональних характеристик**

До загальних функціональних вимог слід віднести:

- введення даних, що відповідають результатам обертань;
- автоматичну класифікацію кожного числа за категоріями;
- обчислення статистики: кількість випадіннь та відсоткове співвідношення для кожної категорії;
- розрахунок ймовірностей наступного випадіння певних категорій з урахуванням серій повторів;
- визначення найбільш ймовірної категорії для наступного обертання;
- підрахунок рекомендованих ставок відповідно до стратегії Мартінгейла;
- збереження та завантаження бази даних гри;
- скидання даних та запуск нової гри.

### **A.2.2 Вимоги до інтерфейсу програми**

До вимог до інтерфейсу слід віднести:

- інтерфейс повинен бути інтуїтивно зрозумілим та зручним у використанні;
- усі елементи управління мають бути логічно структурованими;
- застосунок повинен підтримувати миттєве оновлення статистики та ймовірностей;
- необхідна підтримка введення даних з клавіатури, а також елементів керування для прокрутки історії;
- при введенні некоректних даних мають виводитись повідомлення про помилки;

### **A.2.3 Вимоги до надійності**

До вимог до надійності слід віднести:

- програма має стабільно працювати при тривалому введенні даних;
- має виконуватись перевірка на допустимі значення (0–36);
- передбачена обробка помилок при збереженні та завантаженні даних;
- має бути відсутня можливість аварійного завершення програми через дії користувача.

### **A.2.4 Вимоги до складу та параметрів технічних засобів**

Мінімальні технічні характеристики для запуску програмного забезпечення:

- операційна система: Windows 10/11, Linux (Ubuntu 20.04 або новіше), macOS 10.15 або новіше;

- процесор: з тактовою частотою від 1,5 ГГц;
- оперативна пам'ять: від 4 Гб;
- місце на диску: не менше 100 Мб вільного простору.

**ДОДАТОК Б**  
**Текст програми**

## Б.1 Текст файлу main.py

```

import sys
import sqlite3
from PyQt6.QtWidgets import QApplication, QMainWindow,
QMessageBox, QTableWidgetItem, QLabel, QWidget, QTableWidgetItem
from PyQt6.QtCore import Qt
from PyQt6.QtGui import QIcon
from design import Ui_MainWindow
from math import floor

class NumberHistory(QWidget):
    def __init__(self, scroll_area):
        super().__init__()
        self.scroll_area = scroll_area
        self.setFixedHeight(40)
        self.setStyleSheet("background-color: #121212;")
        self.setMinimumWidth(0)
        self.x_offset = 5 # Положення наступного числа по осі X

    def add_number(self, number):
        label = QLabel(str(number['number']), self)
        label.setAlignment(Qt.AlignmentFlag.AlignCenter)
        label.setFixedSize(15, 15)
        label.move(self.x_offset, 5)
        label.setStyleSheet(f"""QLabel {{
                                background-color: {'green' if
number['number'] == 0 else 'red' if number['color'] == "red" else
'black'}};
                                font-size: 8pt;
                                color: white;
                                text-align: center;}}""")
        label.show()

        self.x_offset += 20 # 30 ширина + 5 відступ

```

```

self.setMinimumWidth(self.x_offset + 10)

self.scroll_area.horizontalScrollBar().setValue(self.scroll_area.
horizontalScrollBar().maximum())

class Analysis(QMainWindow):
    def __init__(self):
        super(Analysis, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.setWindowIcon(QIcon("icon.ico"))

        self.conn = sqlite3.connect("roulette_game.db")
        self.cursor = self.conn.cursor()
        self._create_tables()

        self.two_divider = 18 / 37
        self.three_divider = 12 / 37

        self.color_prob = self.two_divider
        self.current_color = ""
        self.color_streak = 1

        self.parity_prob = self.two_divider
        self.current_parity = ""
        self.parity_streak = 1

        self.half_prob = self.two_divider
        self.current_half = ""
        self.half_streak = 1

        self.dozen_prob = self.three_divider
        self.current_dozen = ""
        self.dozen_streak = 1
        self.dozen_dict = {"1 st 12": 0, "2 nd 12": 0, "3 rd 12":

```

0}

```

self.column_prob = self.three_divider
self.current_column = ""
self.column_streak = 1
self.column_dict = {"1st": 0, "2nd": 0, "3rd": 0}

self.total = 0
self.red_count = 0
self.black_count = 0
self.even_count = 0
self.odd_count = 0
self.half_1 = 0
self.half_2 = 0
self.dozen_1 = 0
self.dozen_2 = 0
self.dozen_3 = 0
self.column_1 = 0
self.column_2 = 0
self.column_3 = 0
self.zero_count = 0
self.max_prob = -1
self.last_prob = -1
self.bet_streak = 1

self.history_numbers = []
self.nums = []
self.nums = self._classification(self.nums)

self.ui.btn_add.setShortcut("Return")
self.ui.btn_add.setFocus()
self.ui.btn_add.clicked.connect(self._on_add_clicked)

self.ui.new_game.clicked.connect(self._msg_new_game)
self.ui.btn_left.clicked.connect(lambda:
self.ui.scroll_nums.horizontalScrollBar().setValue(

```

```

        self.ui.scroll_nums.horizontalScrollBar().value() -
50))

        self.ui.btn_right.clicked.connect(lambda:
self.ui.scroll_nums.horizontalScrollBar().setValue(
        self.ui.scroll_nums.horizontalScrollBar().value() +
50))

        self.ui.save_game.clicked.connect(self._save_game)
        self.ui.load_game.clicked.connect(self._load_game)

        self.ui.table_prob.setRowCount(5)
        self.ui.table_prob.verticalHeader().setVisible(False)

self.ui.table_prob.setEditTriggers(QTableWidget.EditTrigger.NoEdi
tTriggers)

        self.ui.table_prob.setFocusPolicy(Qt.FocusPolicy.NoFocus)

self.ui.table_prob.setSelectionMode(QTableWidget.SelectionMode.No
Selection)

        self.ui.table_prob.setItem(0, 0,
QTableWidgetItem("Червоне/Чорне"))
        self.ui.table_prob.setItem(1, 0,
QTableWidgetItem("Парне/Непарне"))
        self.ui.table_prob.setItem(2, 0, QTableWidgetItem("1 -
18/19 - 36"))
        self.ui.table_prob.setItem(3, 0,
QTableWidgetItem("Дюжина"))
        self.ui.table_prob.setItem(4, 0,
QTableWidgetItem("Колонка"))

        self.ui.table_stats.setRowCount(13)
        self.ui.table_stats.verticalHeader().setVisible(False)

self.ui.table_stats.setEditTriggers(QTableWidget.EditTrigger.NoEd
itTriggers)

```

```

self.ui.table_stats.setFocusPolicy(Qt.FocusPolicy.NoFocus)

self.ui.table_stats.setSelectionMode(QTableWidget.SelectionMode.NoSelection)

        self.ui.table_stats.setItem(0, 0,
QTableWidgetItem("Червоне"))
        self.ui.table_stats.setItem(1, 0,
QTableWidgetItem("Чорне"))
        self.ui.table_stats.setItem(2, 0,
QTableWidgetItem("Парне"))
        self.ui.table_stats.setItem(3, 0,
QTableWidgetItem("Непарне"))
        self.ui.table_stats.setItem(4, 0, QTableWidgetItem("1 -
18"))
        self.ui.table_stats.setItem(5, 0, QTableWidgetItem("19 -
36"))
        self.ui.table_stats.setItem(6, 0, QTableWidgetItem("1 -
12"))
        self.ui.table_stats.setItem(7, 0, QTableWidgetItem("13 -
24"))
        self.ui.table_stats.setItem(8, 0, QTableWidgetItem("25 -
36"))
        self.ui.table_stats.setItem(9, 0, QTableWidgetItem("1-a
колонка"))
        self.ui.table_stats.setItem(10, 0, QTableWidgetItem("2-a
колонка"))
        self.ui.table_stats.setItem(11, 0, QTableWidgetItem("3-a
колонка"))
        self.ui.table_stats.setItem(12, 0,
QTableWidgetItem("Zero"))

        self.nums_history = NumberHistory(self.ui.scroll_nums)
        self.ui.scroll_nums.setWidget(self.nums_history)

    def _create_tables(self):
        # Створює таблиці в БД

```

```

self.cursor.execute("""
    CREATE TABLE IF NOT EXISTS game_history (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        number INTEGER NOT NULL,
        timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
    )
""")

self.cursor.execute("""
    CREATE TABLE IF NOT EXISTS game_stats (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        red_percent REAL,
        red_count INTEGER,
        black_percent REAL,
        black_count INTEGER,
        even_percent REAL,
        even_count INTEGER,
        odd_percent REAL,
        odd_count INTEGER,
        half1_percent REAL,
        half_1 INTEGER,
        half2_percent REAL,
        half_2 INTEGER,
        dozen1_percent REAL,
        dozen_1 INTEGER,
        dozen2_percent REAL,
        dozen_2 INTEGER,
        dozen3_percent REAL,
        dozen_3 INTEGER,
        column1_percent REAL,
        column_1 INTEGER,
        column2_percent REAL,
        column_2 INTEGER,
        column3_percent REAL,
        column_3 INTEGER,
        zero_percent REAL,
        zero_count INTEGER,

```

```

        total INTEGER
    )
    """)
self.cursor.execute("""
    CREATE TABLE IF NOT EXISTS game_probabilities (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        color_prob REAL,
        current_color TEXT,
        parity_prob REAL,
        current_parity TEXT,
        half_prob REAL,
        current_half TEXT,
        dozen_prob REAL,
        current_dozen TEXT,
        column_prob REAL,
        current_column TEXT
    )
    """)
self.conn.commit()

def _msg_new_game(self):
    # Очищає дані гри та БД
    msgbox_clear = QMessageBox()
    msgbox_clear.setIcon(QMessageBox.Icon.Information)
    msgbox_clear.setWindowTitle("Увага")
    msgbox_clear.setText("Ви дійсно хочете почати нову гру?")

    msgbox_clear.setStandardButtons(QMessageBox.StandardButton.Yes |
    QMessageBox.StandardButton.No)

    result = msgbox_clear.exec()
    if result == QMessageBox.StandardButton.Yes:
        self.cursor.execute("DELETE FROM game_history")
        self.cursor.execute("DELETE FROM game_stats")
        self.cursor.execute("DELETE FROM game_probabilities")
        self.cursor.execute("DELETE FROM sqlite_sequence

```

```

WHERE name='game_history'")
        self.conn.commit()
        self.close()
        self.__init__()
        self.show()

    def _save_game(self):
        # Зберігати у БД
        msgbox_save_game = QMessageBox()
        msgbox_save_game.setIcon(QMessageBox.Icon.Information)
        msgbox_save_game.setWindowTitle("Увага")
        msgbox_save_game.setText("Ви дійсно хочете зберегти
гру?")

        msgbox_save_game.setStandardButtons(QMessageBox.StandardButton.Yes
s | QMessageBox.StandardButton.No)
        result = msgbox_save_game.exec()
        if result == QMessageBox.StandardButton.Yes:
            try:
                # Очищення попередніх даних
                self.cursor.execute("DELETE FROM game_history")
                self.cursor.execute("DELETE FROM game_stats")
                self.cursor.execute("DELETE FROM
game_probabilities")
                self.cursor.execute("DELETE FROM sqlite_sequence
WHERE name='game_history'")

                # Збереження історії чисел
                for num in self.history_numbers:
                    self.cursor.execute("INSERT INTO game_history
(number) VALUES (?)", (num,))

                # Збереження статистики
                self.cursor.execute("""
                    INSERT INTO game_stats (
                        red_percent, red_count, black_percent,

```

```

black_count, even_percent, even_count, odd_percent, odd_count,
        half1_percent, half_1, half2_percent,
half_2, dozen1_percent, dozen_1, dozen2_percent,
        dozen_2, dozen3_percent, dozen_3,
column1_percent, column_1, column2_percent, column_2,
        column3_percent, column_3, zero_percent,
zero_count, total
        ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
        """, (
            self.red_percent, self.red_count,
self.black_percent, self.black_count, self.even_percent,
self.even_count, self.odd_percent, self.odd_count,
            self.half1_percent, self.half_1,
self.half2_percent, self.half_2, self.dozen1_percent,
self.dozen_1, self.dozen2_percent, self.dozen_2,
self.dozen3_percent, self.dozen_3,
            self.column1_percent, self.column_1,
self.column2_percent, self.column_2, self.column3_percent,
self.column_3, self.zero_percent, self.zero_count, self.total
        ))

# Збереження ймовірностей
self.cursor.execute("""
        INSERT INTO game_probabilities (
            color_prob, current_color, parity_prob,
current_parity,
            half_prob, current_half, dozen_prob,
current_dozen,
            column_prob, current_column
        ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
        """, (
            self.color_prob, self.current_color,
self.parity_prob, self.current_parity,
self.half_prob, self.current_half,
self.dozen_prob, self.current_dozen,

```

```

        self.column_prob, self.current_column
    ))

    self.conn.commit()
    QMessageBox.information(self, "Успіх", "Гру
успішно збережено!")

    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка
при збереженні гри: {str(e)}")

def _load_game(self):
    # Завантажує БД
    msgbox_load_game = QMessageBox()
    msgbox_load_game.setIcon(QMessageBox.Icon.Information)
    msgbox_load_game.setWindowTitle("Увага")
    msgbox_load_game.setText("Ви дійсно хочете завантажити
збережений стан гри?")

msgbox_load_game.setStandardButtons(QMessageBox.StandardButton.Yes
| QMessageBox.StandardButton.No)
    result = msgbox_load_game.exec()

    if result == QMessageBox.StandardButton.Yes:
        try:
            # Завантаження історії чисел
            self.cursor.execute("SELECT number FROM
game_history")

            numbers = [row[0] for row in
self.cursor.fetchall()]

            if not numbers:
                raise ValueError("Немає збереженої історії
чисел.")

            # Завантаження статистики
            self.cursor.execute("SELECT * FROM game_stats")

```

```

stats = self.cursor.fetchone()
if stats:
    self.red_percent = stats[1]
    self.red_count = stats[2]
    self.black_percent = stats[3]
    self.black_count = stats[4]
    self.even_percent = stats[5]
    self.even_count = stats[6]
    self.odd_percent = stats[7]
    self.odd_count = stats[8]
    self.half1_percent = stats[9]
    self.half_1 = stats[10]
    self.half2_percent = stats[11]
    self.half_2 = stats[12]
    self.dozen1_percent = stats[13]
    self.dozen_1 = stats[14]
    self.dozen2_percent = stats[15]
    self.dozen_2 = stats[16]
    self.dozen3_percent = stats[17]
    self.dozen_3 = stats[18]
    self.column1_percent = stats[19]
    self.column_1 = stats[20]
    self.column2_percent = stats[21]
    self.column_2 = stats[22]
    self.column3_percent = stats[23]
    self.column_3 = stats[24]
    self.zero_precent = stats[25]
    self.zero_count = stats[26]
    self.total = stats[27]

# Завантаження ймовірностей
self.cursor.execute("SELECT * FROM
game_probabilities")
probs = self.cursor.fetchone()
if probs:
    self.color_prob = probs[1]

```

```

        self.current_color = probs[2]
        self.parity_prob = probs[3]
        self.current_parity = probs[4]
        self.half_prob = probs[5]
        self.current_half = probs[6]
        self.dozen_prob = probs[7]
        self.current_dozen = probs[8]
        self.column_prob = probs[9]
        self.current_column = probs[10]

    # Оновлення інтерфейсу
    self.close()
    self.__init__()
    self.show()

    for num in numbers:
        self._probability(self.nums, num)
        self._stats(self.nums, num)

    QMessageBox.information(self, "Успіх", "Гру
успішно завантажено!")
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка
при завантаженні гри: {str(e)}")

def _on_add_clicked(self):
    text = self.ui.line_add.text()
    try:
        self._probability(self.nums, text)
        self._stats(self.nums, text)
    except ValueError:
        msgbox_value_error = QMessageBox()
        msgbox_value_error.setIcon(QMessageBox.Icon.Warning)
        msgbox_value_error.setWindowTitle("Помилка")
        msgbox_value_error.setText("Невірний тип даних!
Введіть ціле число від 0 до 36!")

```

```
msgbox_value_error.setStandardButtons(QMessageBox.StandardButton.
Ok)
```

```
msgbox_value_error.exec()
self.ui.line_add.clear()
```

```
def _classification(self, classified_dict):
    red_nums = (1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23,
25, 27, 30, 32, 34, 36)
    for i in range(37):
        if i == 0:
            color = "none"
            parity = "none"
            half = "none"
            dozen = "none"
            column = "none"
        else:
            color = "red" if i in red_nums else "black"
            parity = "even" if i % 2 == 0 else "odd"
            half = "1 to 18" if 1 <= i <= 18 else "19 to 36"
            if 1 <= i <= 12:
                dozen = "1 st 12"
            elif 13 <= i <= 24:
                dozen = "2 nd 12"
            elif 25 <= i <= 36:
                dozen = "3 rd 12"
            else:
                dozen = "none"
            if i % 3 == 1:
                column = "1st"
            elif i % 3 == 2:
                column = "2nd"
            elif i % 3 == 0:
                column = "3rd"
            else:
                column = "none"
```

```

        classified_dict.append({
            "number": i,
            "color": color,
            "parity": parity,
            "dozen": dozen,
            "column": column,
            "1 to 18/19 to 36": half
        })

    classified_dict = {item["number"]: item for item in
classified_dict}
    return classified_dict

def _probability(self, classified_dict, user_input):
    user_input = int(user_input)
    if user_input in classified_dict:
        info = classified_dict[user_input]
        self.nums_history.add_number(info)
        self.history_numbers.append(user_input)

    if not user_input == 0:
        # Імовірність кольору
        if info['color'] == self.current_color:
            self.color_streak = 1
            self.current_color = "red" if
self.current_color == "black" else "black"
            self.color_prob = self.two_divider
        else:
            if self.current_color == "":
                self.current_color = "red" if
info['color'] == "black" else "black"
            else:
                self.color_streak += 1
                self.color_prob = 1 - (19 / 37) **
self.color_streak

```

```

# Імовірність парності
if info['parity'] == self.current_parity:
    self.parity_streak = 1
    self.current_parity = "even" if
self.current_parity == "odd" else "odd"
    self.parity_prob = self.two_divider
else:
    if self.current_parity == "":
        self.current_parity = "even" if
info['parity'] == "odd" else "odd"
    else:
        self.parity_streak += 1
        self.parity_prob = 1 - (19 / 37) **
self.parity_streak

# Імовірність половини
if info['1 to 18/19 to 36'] == self.current_half:
    self.half_streak = 1
    self.current_half = "1 to 18" if
self.current_half == "19 to 36" else "19 to 36"
    self.half_prob = self.two_divider
else:
    if self.current_half == "":
        self.current_half = "1 to 18" if info['1
to 18/19 to 36'] == "19 to 36" else "19 to 36"
    else:
        self.half_streak += 1
        self.half_prob = 1 - (19 / 37) **
self.half_streak

# Імовірність дюжини
for i in self.dozen_dict:
    if i == info['dozen']:
        self.dozen_dict[info['dozen']] += 1

```

```

if info['dozen'] == self.current_dozen:
    self.dozen_streak = 1
    self.dozen_prob = self.three_divider
    next_dozen = []
    for i in self.dozen_dict:
        if not i == info['dozen']:
            next_dozen.append(self.dozen_dict[i])
    min_dozen = min(next_dozen)
    next_dozen.clear()
    for i in self.dozen_dict:
        if not i == info['dozen']:
            if self.dozen_dict[i] == min_dozen:
                self.current_dozen = i
    else:
        if self.current_dozen == "":
            for i in self.dozen_dict:
                if i == info['dozen']:
                    self.current_dozen = "2 nd 12" if
i == "1 st 12" else "1 st 12"
        else:
            self.dozen_streak += 1
            self.dozen_prob = 1 - (25 / 37) **
self.dozen_streak

```

```

# Імовірність колонки
for i in self.column_dict:
    if i == info['column']:
        self.column_dict[info['column']] += 1

if info['column'] == self.current_column:
    self.column_streak = 1
    self.column_prob = self.three_divider
    next_column = []
    for i in self.column_dict:
        if not i == info['column']:

```

```

next_column.append(self.column_dict[i])
    min_column = min(next_column)
    next_column.clear()
    for i in self.column_dict:
        if not i == info['column']:
            if self.column_dict[i] == min_column:
                self.current_column = i
    else:
        if self.current_column == "":
            for i in self.column_dict:
                if i == info['column']:
                    self.current_column = "2nd" if i
== "1st" else "1st"
        else:
            self.column_streak += 1
            self.column_prob = 1 - (25 / 37) **
self.column_streak

rounded_color = floor(self.color_prob * 10000) /
100
rounded_parity = floor(self.parity_prob * 10000)
/ 100
rounded_half = floor(self.half_prob * 10000) /
100
rounded_dozen = floor(self.dozen_prob * 10000) /
100
rounded_column = floor(self.column_prob * 10000)
/ 100

self.ui.table_prob.setItem(self.max_prob, 3,
QTableWidgetItem(""))
best_prob = []

self.ui.table_prob.setItem(0, 1,
QTableWidgetItem("Червоне")) if self.current_color == "red" else
self.ui.table_prob.setItem(0, 1, QTableWidgetItem("Чорне"))

```

```

        self.ui.table_prob.setItem(0, 2,
QTableWidgetItem(str(rounded_color) + "%"))
        best_prob.append(rounded_color)

        self.ui.table_prob.setItem(1, 1,
QTableWidgetItem("Парне")) if self.current_parity == "even" else
self.ui.table_prob.setItem(1, 1, QTableWidgetItem("Непарне"))
        self.ui.table_prob.setItem(1, 2,
QTableWidgetItem(str(rounded_parity) + "%"))
        best_prob.append(rounded_parity)

        self.ui.table_prob.setItem(2, 1,
QTableWidgetItem("1 - 18")) if self.current_half == "1 to 18"
else self.ui.table_prob.setItem(2, 1, QTableWidgetItem("19 -
36"))

        self.ui.table_prob.setItem(2, 2,
QTableWidgetItem(str(rounded_half) + "%"))
        best_prob.append(rounded_half)

        if self.current_dozen == "1 st 12":
            self.ui.table_prob.setItem(3, 1,
QTableWidgetItem("1 - 12"))
        elif self.current_dozen == "2 nd 12":
            self.ui.table_prob.setItem(3, 1,
QTableWidgetItem("13 - 24"))
        elif self.current_dozen == "3 rd 12":
            self.ui.table_prob.setItem(3, 1,
QTableWidgetItem("25 - 36"))
        self.ui.table_prob.setItem(3, 2,
QTableWidgetItem(str(rounded_dozen) + "%"))
        best_prob.append(rounded_dozen)

        if self.current_column == "1st":
            self.ui.table_prob.setItem(4, 1,
QTableWidgetItem("1-а колонка"))
        elif self.current_column == "2nd":

```

```

        self.ui.table_prob.setItem(4, 1,
QTableWidgetItem("2-а колонка"))
        elif self.current_column == "3rd":
            self.ui.table_prob.setItem(4, 1,
QTableWidgetItem("3-а колонка"))
            self.ui.table_prob.setItem(4, 2,
QTableWidgetItem(str(rounded_column) + "%"))
            best_prob.append(rounded_column)

max_value = max(best_prob)
if max_value > 70.0:
    self.max_prob = best_prob.index(max_value)
    if self.max_prob == self.last_prob:
        self.bet_streak *= 2
    else:
        self.bet_streak = 1
    self.last_prob = self.max_prob
    self.ui.table_prob.setItem(self.max_prob, 3,
QTableWidgetItem(f"x{self.bet_streak}"))
else:
    msgbox_amount_error = QMessageBox()
    msgbox_amount_error.setIcon(QMessageBox.Icon.Warning)
    msgbox_amount_error.setWindowTitle("Помилка")
    msgbox_amount_error.setText("Введіть число від 0 до
36!")

msgbox_amount_error.setStandardButtons(QMessageBox.StandardButton
.Ok)

    msgbox_amount_error.exec()
    self.ui.line_add.clear()

def _stats(self, classified_dict, user_input):
    user_input = int(user_input)
    if user_input in classified_dict:
        if user_input == 0:
            self.zero_count += 1

```

```

else:
    info = classified_dict[user_input]

    if info['color'] == "red":
        self.red_count += 1
    else:
        self.black_count += 1

    if info['parity'] == "even":
        self.even_count += 1
    else:
        self.odd_count += 1

    if info['1 to 18/19 to 36'] == "1 to 18":
        self.half_1 += 1
    else:
        self.half_2 += 1

    if info['dozen'] == "1 st 12":
        self.dozen_1 += 1
    elif info['dozen'] == "2 nd 12":
        self.dozen_2 += 1
    elif info['dozen'] == "3 rd 12":
        self.dozen_3 += 1

    if info['column'] == "1st":
        self.column_1 += 1
    if info['column'] == "2nd":
        self.column_2 += 1
    elif info['column'] == "3rd":
        self.column_3 += 1

self.total += 1

self.red_percent = round(self.red_count / self.total
* 100, 2)

```

```

        self.black_percent = round(self.black_count /
self.total * 100, 2)
        self.even_percent = round(self.even_count /
self.total * 100, 2)
        self.odd_percent = round(self.odd_count / self.total
* 100, 2)
        self.half1_percent = round(self.half_1 / self.total *
100, 2)
        self.half2_percent = round(self.half_2 / self.total *
100, 2)
        self.dozen1_percent = round(self.dozen_1 / self.total
* 100, 2)
        self.dozen2_percent = round(self.dozen_2 / self.total
* 100, 2)
        self.dozen3_percent = round(self.dozen_3 / self.total
* 100, 2)
        self.column1_percent = round(self.column_1 /
self.total * 100, 2)
        self.column2_percent = round(self.column_2 /
self.total * 100, 2)
        self.column3_percent = round(self.column_3 /
self.total * 100, 2)
        self.zero_precent = round(self.zero_count /
self.total * 100, 2)

        self.ui.table_stats.setItem(0, 1,
QTableWidgetItem(str(self.red_percent) + "%"))
        self.ui.table_stats.setItem(0, 2,
QTableWidgetItem(str(self.red_count)))
        self.ui.table_stats.setItem(1, 1,
QTableWidgetItem(str(self.black_percent) + "%"))
        self.ui.table_stats.setItem(1, 2,
QTableWidgetItem(str(self.black_count)))
        self.ui.table_stats.setItem(2, 1,
QTableWidgetItem(str(self.even_percent) + "%"))
        self.ui.table_stats.setItem(2, 2,

```

```

QTableWidgetItem(str(self.even_count)))
        self.ui.table_stats.setItem(3, 1,
QTableWidgetItem(str(self.odd_percent) + "%"))
        self.ui.table_stats.setItem(3, 2,
QTableWidgetItem(str(self.odd_count)))
        self.ui.table_stats.setItem(4, 1,
QTableWidgetItem(str(self.half1_percent) + "%"))
        self.ui.table_stats.setItem(4, 2,
QTableWidgetItem(str(self.half_1)))
        self.ui.table_stats.setItem(5, 1,
QTableWidgetItem(str(self.half2_percent) + "%"))
        self.ui.table_stats.setItem(5, 2,
QTableWidgetItem(str(self.half_2)))
        self.ui.table_stats.setItem(6, 1,
QTableWidgetItem(str(self.dozen1_percent) + "%"))
        self.ui.table_stats.setItem(6, 2,
QTableWidgetItem(str(self.dozen_1)))
        self.ui.table_stats.setItem(7, 1,
QTableWidgetItem(str(self.dozen2_percent) + "%"))
        self.ui.table_stats.setItem(7, 2,
QTableWidgetItem(str(self.dozen_2)))
        self.ui.table_stats.setItem(8, 1,
QTableWidgetItem(str(self.dozen3_percent) + "%"))
        self.ui.table_stats.setItem(8, 2,
QTableWidgetItem(str(self.dozen_3)))
        self.ui.table_stats.setItem(9, 1,
QTableWidgetItem(str(self.column1_percent) + "%"))
        self.ui.table_stats.setItem(9, 2,
QTableWidgetItem(str(self.column_1)))
        self.ui.table_stats.setItem(10, 1,
QTableWidgetItem(str(self.column2_percent) + "%"))
        self.ui.table_stats.setItem(10, 2,
QTableWidgetItem(str(self.column_2)))
        self.ui.table_stats.setItem(11, 1,
QTableWidgetItem(str(self.column3_percent) + "%"))
        self.ui.table_stats.setItem(11, 2,

```

```

QTableWidgetItem(str(self.column_3)))
        self.ui.table_stats.setItem(12, 1,
QTableWidgetItem(str(self.zero_precent) + "%"))
        self.ui.table_stats.setItem(12, 2,
QTableWidgetItem(str(self.zero_count)))

    def closeEvent(self, event):
        """Закриває з'єднання з базою даних при закритті
програми."""
        self.conn.close()
        super().closeEvent(event)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = Analysis()
    window.show()
    sys.exit(app.exec())

```

## Б.2 Текст файлу design.py

```

from PyQt6 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(730, 600)
        MainWindow.setMinimumSize(QtCore.QSize(730, 600))
        MainWindow.setMaximumSize(QtCore.QSize(730, 600))
        MainWindow.setStyleSheet("QMainWindow {\n"
"    background-color: #283655;\n"
"    color: #c4dfe6;\n"
"    font-family: \"Bahnschrift SemiBold\";\n"
"}\n"
"QLabel {\n"
"    color: #c4dfe6;\n"

```

```

"    font-family: \"Verdana\";\n"
"    font-size: 16pt;\n"
"}\n"
"QPushButton{\n"
"    color: #c4dfe6;\n"
"    border: 2px solid #1e1f26;\n"
"    border-radius: 5px;\n"
"    font-size: 14px;\n"
"    background-color: #4d648d;\n"
"}\n"
"QPushButton:hover{\n"
"    background-color: #d0e1f9;\n"
"}\n"
"QPushButton:pressed{\n"
"    background-color: #d0e1f9;\n"
"    border: 5px solid #1e1f26;\n"
"}\n"
"QLineEdit{\n"
"    color: #c4dfe6;\n"
"    background-color: #d0e1f9;\n"
"    font-size: 20;\n"
"    border: 2px solid #1e1f26;\n"
"    border-radius: 3px;\n"
"}\n"
"QTableWidget {\n"
"    background-color: #121212;\n"
"    color: #c4dfe6;\n"
"    font-family: \"Bahnschrift SemiBold\";\n"
"}\n"
"QHeaderView::section {\n"
"    background-color: #673AB7;\n"
"    color: #c4dfe6;\n"
"    padding: 4px;\n"
"    border: 1px solid #5E35B1;\n"
"    font-weight: bold;\n"
"}\n"

```

```
"Line{\n"
"    color: #c4dfe6;\n"
"}")
```

```
MainWindow.setLocale(QQtCore.QLocale(QQtCore.QLocale.Language.Ukrai
nian, QtCore.QLocale.Country.Ukraine))
```

```
    self.centralwidget = QtWidgets.QWidget(parent=MainWindow)
    self.centralwidget.setStyleSheet("QMainWindow {\n"
"    background-color: #283655;\n"
"    color: #d0e1f9;\n"
"    font-family: \"Bahnschrift SemiBold\";\n"
"}\n"
"QLabel {\n"
"    color: #d0e1f9;\n"
"    font-family: \"ahnschrift SemiBold\";\n"
"    font-size: 16pt;\n"
"}\n"
"QPushButton{\n"
"    color: #d0e1f9;\n"
"    border: 1px solid #1e1f26;\n"
"    border-radius: 5px;\n"
"    font-size: 14px;\n"
"    background-color: #4d648d;\n"
"}\n"
"QPushButton:hover{\n"
"    background-color: #d0e1f9;\n"
"}\n"
"QPushButton:pressed{\n"
"    background-color: #d0e1f9;\n"
"    border: 3px solid #1e1f26;\n"
"}\n"
"QLineEdit{\n"
"    color: #1e1f26;\n"
"    background-color: #d0e1f9;\n"
"    font-size: 20;\n"
"    border: 1px solid #1e1f26;\n"
"}
```

```

"    border-radius: 5px;\n"
"}\n"
"QWidget {\n"
"    background-color: #283655;\n"
"    border: 1px solid #1e1f26;\n"
"    color: #d0e1f9;\n"
"    font-family: \"Bahnschrift SemiBold\";\n"
"}\n"
"QHeaderView::section {\n"
"    background-color: #283655;\n"
"    color: #d0e1f9;\n"
"    font-weight: bold;\n"
"}\n"
"Line{\n"
"    color: #c4dfe6;\n"
"}")

        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayout =
QtWidgets.QVBoxLayout(self.centralwidget)
        self.verticalLayout.setObjectName("verticalLayout")
        self.lbl_naming =
QtWidgets.QLabel(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Preferred,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.lbl_naming.sizePolicy().hasHeightForWidth())

        self.lbl_naming.setSizePolicy(sizePolicy)
        self.lbl_naming.setMinimumSize(QtCore.QSize(0, 0))
        font = QtGui.QFont()
        font.setFamily("Bahnschrift SemiBold")
        font.setPointSize(-1)
        self.lbl_naming.setFont(font)

```

```

        self.lbl_naming.setStyleSheet("QLabel{\n"
"    font-family: \"Bahnschrift SemiBold\";\n"
"    font-size: 30px;\n"
"}")

        self.lbl_naming.setObjectName("lbl_naming")
        self.verticalLayout.addWidget(self.lbl_naming, 0,
QtCore.Qt.AlignmentFlag.AlignHCenter|QtCore.Qt.AlignmentFlag.AlignVCenter)

        self.line = QtWidgets.QFrame(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.line.sizePolicy().hasHeightForWidth())

        self.line.setSizePolicy(sizePolicy)
        self.line.setMinimumSize(QtCore.QSize(0, 25))
        self.line.setMaximumSize(QtCore.QSize(712, 3))
        self.line.setStyleSheet("Line{\n"
"    margin: 10px;\n"
"}")

        self.line setFrameShape(QtWidgets.QFrame.Shape.HLine)
        self.line setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)
        self.line.setObjectName("line")
        self.verticalLayout.addWidget(self.line, 0,
QtCore.Qt.AlignmentFlag.AlignVCenter)

        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.btn_add =
QtWidgets.QPushButton(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)

```

```

        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn_add.sizePolicy().hasHeightF
orWidth())
        self.btn_add.setSizePolicy(sizePolicy)
        self.btn_add.setMinimumSize(QtCore.QSize(60, 30))

self.btn_add.setCursor(QtGui.QCursor(QtCore.Qt.CursorShape.Pointi
ngHandCursor))
        self.btn_add.setObjectName("btn_add")
        self.horizontalLayout.addWidget(self.btn_add)
        self.line_add =
QtWidgets.QLineEdit(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Maximum)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.line_add.sizePolicy().hasHeight
ForWidth())
        self.line_add.setSizePolicy(sizePolicy)
        self.line_add.setMinimumSize(QtCore.QSize(0, 30))
        self.line_add.setObjectName("line_add")
        self.horizontalLayout.addWidget(self.line_add)
        self.btn_left =
QtWidgets.QPushButton(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn_left.sizePolicy().hasHeight
ForWidth())
        self.btn_left.setSizePolicy(sizePolicy)

```

```

        self.btn_left.setMinimumSize(QtCore.QSize(30, 30))

self.btn_left.setCursor(QtGui.QCursor(QtCore.Qt.CursorShape.PointingHandCursor))

        self.btn_left.setObjectName("btn_left")
        self.horizontalLayout.addWidget(self.btn_left)
        self.scroll_nums =
QtWidgets.QScrollArea(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Expanding,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.scroll_nums.sizePolicy().hasHeightForWidth())

        self.scroll_nums.setSizePolicy(sizePolicy)
        self.scroll_nums.setMinimumSize(QtCore.QSize(0, 0))
        self.scroll_nums.setMaximumSize(QtCore.QSize(16777215,
30))

        self.scroll_nums.setStyleSheet("QScrollArea {\n"
"    background-color: #283655;\n"
"    border: 1px solid #1e1f26;\n"
"    border-radius: 5px;\n"
"}")

self.scroll_nums.setVerticalScrollBarPolicy(QtCore.Qt.ScrollBarPolicy.ScrollBarAlwaysOff)

self.scroll_nums.setHorizontalScrollBarPolicy(QtCore.Qt.ScrollBarPolicy.ScrollBarAlwaysOff)

        self.scroll_nums.setWidgetResizable(True)
        self.scroll_nums.setObjectName("scroll_nums")
        self.scroll_content = QtWidgets.QWidget()
        self.scroll_content.setGeometry(QtCore.QRect(0, 0, 16,
16))

```

```

        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.scroll_content.sizePolicy().has
HeightForWidth())
        self.scroll_content.setSizePolicy(sizePolicy)
        self.scroll_content.setStyleSheet("QWidget {\n"
"    background-color: #121212;\n"
"}")

        self.scroll_content.setObjectName("scroll_content")
        self.scroll_nums.setWidget(self.scroll_content)
        self.horizontalLayout.addWidget(self.scroll_nums, 0,
QtCore.Qt.AlignmentFlag.AlignVCenter)
        self.btn_right =
QtWidgets.QPushButton(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.btn_right.sizePolicy().hasHeigh
tForWidth())
        self.btn_right.setSizePolicy(sizePolicy)
        self.btn_right.setMinimumSize(QtCore.QSize(30, 30))

self.btn_right.setCursor(QtGui.QCursor(QtCore.Qt.CursorShape.Poin
tingHandCursor))
        self.btn_right.setObjectName("btn_right")
        self.horizontalLayout.addWidget(self.btn_right)
        self.verticalLayout.addLayout(self.horizontalLayout)
        self.horizontalLayout_3 = QtWidgets.QHBoxLayout()

```

```

self.horizontalLayout_3.setObjectName("horizontalLayout_3")
    self.table_prob =
QtWidgets.QTableWidget(parent=self.centralwidget)
    sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Expanding,
QtWidgets.QSizePolicy.Policy.Expanding)
    sizePolicy.setHorizontalStretch(0)
    sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.table_prob.sizePolicy().hasHeightForWidth())
    self.table_prob.setSizePolicy(sizePolicy)
    self.table_prob.setStyleSheet("QHeaderView::section {\n"
"    border: none;\n"
"    background-color: #283655;\n"
"    gridline-color: #1e1f26;\n"
"    color: #d0e1f9;\n"
"    font-weight: bold;\n"
"}")

self.table_prob.setHorizontalScrollBarPolicy(QtCore.Qt.ScrollBarPolicy.ScrollBarAlwaysOff)
    self.table_prob.setObjectName("table_prob")
    self.table_prob.setColumnCount(4)
    self.table_prob.setRowCount(0)
    item = QtWidgets.QTableWidgetItem()
    self.table_prob.setHorizontalHeaderItem(0, item)
    item = QtWidgets.QTableWidgetItem()
    self.table_prob.setHorizontalHeaderItem(1, item)
    item = QtWidgets.QTableWidgetItem()
    self.table_prob.setHorizontalHeaderItem(2, item)
    item = QtWidgets.QTableWidgetItem()
    self.table_prob.setHorizontalHeaderItem(3, item)
    self.horizontalLayout_3.addWidget(self.table_prob)
    self.table_stats =
QtWidgets.QTableWidget(parent=self.centralwidget)

```

```

        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Expanding)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.table_stats.sizePolicy().hasHeightForWidth())

        self.table_stats.setSizePolicy(sizePolicy)
        self.table_stats.setMinimumSize(QtCore.QSize(303, 0))
        self.table_stats.setStyleSheet("QHeaderView::section {\n"
"    border: none;\n"
"    background-color: #283655;\n"
"    gridline-color: #1e1f26;\n"
"    color: #d0e1f9;\n"
"    font-weight: bold;\n"
"}")

self.table_stats.setHorizontalScrollBarPolicy(QtCore.Qt.ScrollBarPolicy.ScrollBarAlwaysOff)

        self.table_stats.setObjectName("table_stats")
        self.table_stats.setColumnCount(3)
        self.table_stats.setRowCount(0)
        item = QtWidgets.QTableWidgetItem()
        self.table_stats.setHorizontalHeaderItem(0, item)
        item = QtWidgets.QTableWidgetItem()
        self.table_stats.setHorizontalHeaderItem(1, item)
        item = QtWidgets.QTableWidgetItem()
        self.table_stats.setHorizontalHeaderItem(2, item)
        self.horizontalLayout_3.addWidget(self.table_stats)
        self.verticalLayout.addLayout(self.horizontalLayout_3)
        self.horizontalLayout_4 = QtWidgets.QHBoxLayout()

self.horizontalLayout_4.setObjectName("horizontalLayout_4")
        self.new_game =
QtWidgets.QPushButton(parent=self.centralwidget)

```

```

        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.new_game.sizePolicy().hasHeight
ForWidth())
        self.new_game.setSizePolicy(sizePolicy)
        self.new_game.setMinimumSize(QtCore.QSize(100, 30))

self.new_game.setCursor(QtGui.QCursor(QtCore.Qt.CursorShape.Point
ingHandCursor))
        self.new_game.setCheckable(True)
        self.new_game.setObjectName("new_game")
        self.horizontalLayout_4.addWidget(self.new_game)
        self.save_game =
QtWidgets.QPushButton(parent=self.centralwidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.save_game.sizePolicy().hasHeigh
tForWidth())
        self.save_game.setSizePolicy(sizePolicy)
        self.save_game.setMinimumSize(QtCore.QSize(100, 30))

self.save_game.setCursor(QtGui.QCursor(QtCore.Qt.CursorShape.Poin
tingHandCursor))
        self.save_game.setCheckable(True)
        self.save_game.setObjectName("save_game")
        self.horizontalLayout_4.addWidget(self.save_game)
        self.load_game =
QtWidgets.QPushButton(parent=self.centralwidget)

```

```

        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Policy.Fixed,
QtWidgets.QSizePolicy.Policy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.load_game.sizePolicy().hasHeight
tForWidth())
        self.load_game.setSizePolicy(sizePolicy)
        self.load_game.setMinimumSize(QtCore.QSize(100, 30))

self.load_game.setCursor(QtGui.QCursor(QtCore.Qt.CursorShape.PointingHandCursor))
        self.load_game.setCheckable(True)
        self.load_game.setObjectName("load_game")
        self.horizontalLayout_4.addWidget(self.load_game)
        self.verticalLayout.addLayout(self.horizontalLayout_4)
MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow",
"Аналіз Колеса Фортуни"))
        self.lbl_naming.setText(_translate("MainWindow", "Аналіз
Колеса Фортуни"))
        self.btn_add.setText(_translate("MainWindow", "Додати"))
        self.btn_left.setText(_translate("MainWindow", "←"))
        self.btn_right.setText(_translate("MainWindow", "→"))
        item = self.table_prob.horizontalHeaderItem(0)
        item.setText(_translate("MainWindow", "Категорія"))
        item = self.table_prob.horizontalHeaderItem(1)
        item.setText(_translate("MainWindow", "Значення"))
        item = self.table_prob.horizontalHeaderItem(2)

```

```

        item.setText(_translate("MainWindow", "Імовірність"))
        item = self.table_prob.horizontalHeaderItem(3)
        item.setText(_translate("MainWindow", "Ставка"))
        item = self.table_stats.horizontalHeaderItem(0)
        item.setText(_translate("MainWindow", "Категорія"))
        item = self.table_stats.horizontalHeaderItem(1)
        item.setText(_translate("MainWindow", "Значення"))
        item = self.table_stats.horizontalHeaderItem(2)
        item.setText(_translate("MainWindow", "Кількість"))
        self.new_game.setText(_translate("MainWindow", "Нова
Гра"))
        self.save_game.setText(_translate("MainWindow",
"Зберегти"))
        self.load_game.setText(_translate("MainWindow",
"Завантажити"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec())

```

**ДОДАТОК В**  
**Слайди презентації**

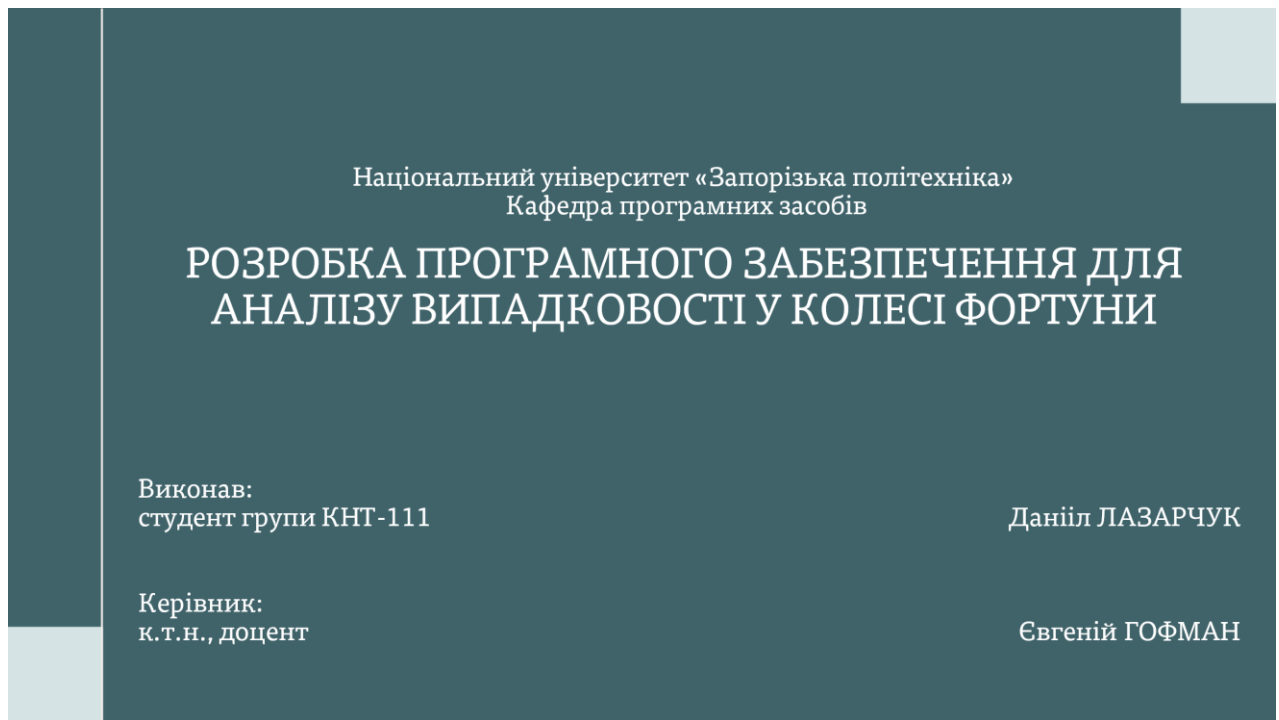


Рисунок Д.1 – Слайд 1

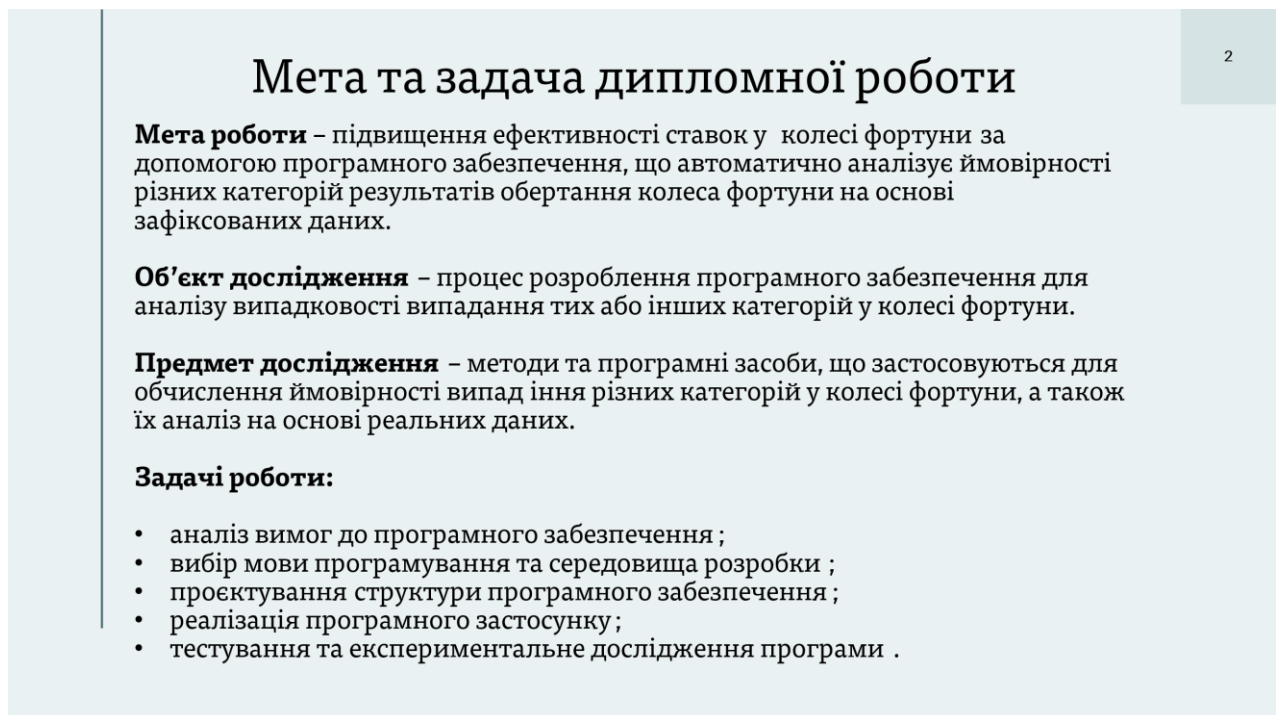


Рисунок Д.2 – Слайд 2

## Аналіз вимог до роботи

### Функціональні вимоги:

- введення результатів гри;
- автоматичну класифікацію кожного веденого числа за категоріями;
- обчислення ймовірностей появи наступного значення для кожної категорії з урахуванням серій повторів;
- визначення категорії з найвищою ймовірністю для наступного обертання;
- автоматичний підрахунок ставок згідно зі стратегією Мартінгейла на основі послідовності втрат;
- збереження та завантаження історії введених чисел у файл.

### Вимоги до інтерфейсу:

- зрозуміле розміщення елементів;
- мінімалістичний дизайн;
- миттєве оновлення інформації;
- наявність інтерактивних елементів управління;
- інформативні таблиці.

Рисунок Д.3 – Слайд 3

## Основні формули алгоритмів:

$$P(i) = \frac{N_i}{N}$$

де  $N_i$  - кількість випадінь певного результату, а  $N$  - загальна кількість обертань

$$P = 1 - \left(\frac{a}{37}\right)^n$$

де  $a$  - кількість небажаного результату певної категорії, а  $n$  - кількість невдалих спроб поспіль

Рисунок Д.4 – Слайд 4

# Актуальність роботи

Приклади існуючих програмних рішень:

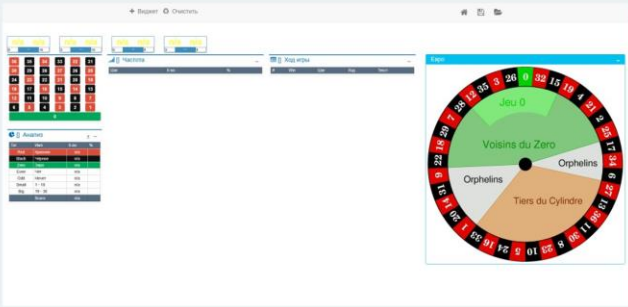


Рисунок Д.5 – Слайд 5

# Порівняння мов програмування

| Критерій порівняння                          | Python | Java | C++ |
|----------------------------------------------|--------|------|-----|
| Простота синтаксису                          | +      | -    | -   |
| Швидкість розробки                           | +      | -    | -   |
| Динамічна типізація                          | +      | -    | -   |
| Підтримка різних типів моделей програмування | +      | -    | +   |
| Кросплатформність                            | +      | +    | +   |

Рисунок Д.6 – Слайд 6

# Порівняння середовищ розробки

7

| Критерій порівняння   | PyCharm Community Edition | Visual Studio Code | Spyder |
|-----------------------|---------------------------|--------------------|--------|
| Автозаповнення коду   | +                         | +                  | +      |
| Підтримка Git         | +                         | +                  | -      |
| Плагіни та розширення | +                         | +                  | -      |
| Швидкість роботи      | +                         | -                  | +      |
| Безкоштовність        | +                         | +                  | +      |

Рисунок Д.7 – Слайд 7

# Структурна схема системи

8

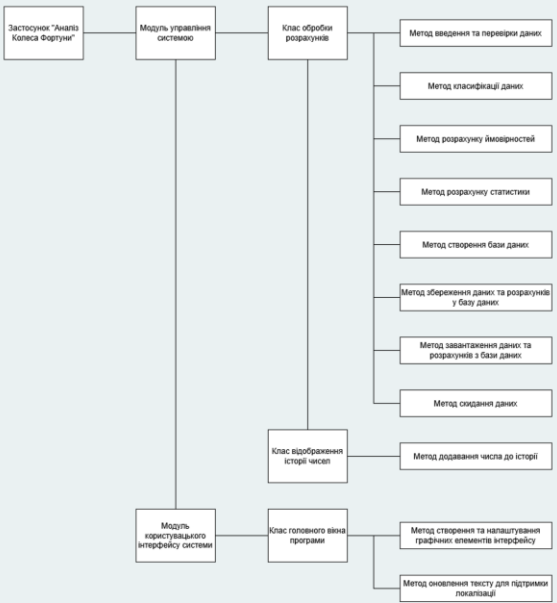


Рисунок Д.8 – Слайд 8

# Функціональна схема програми

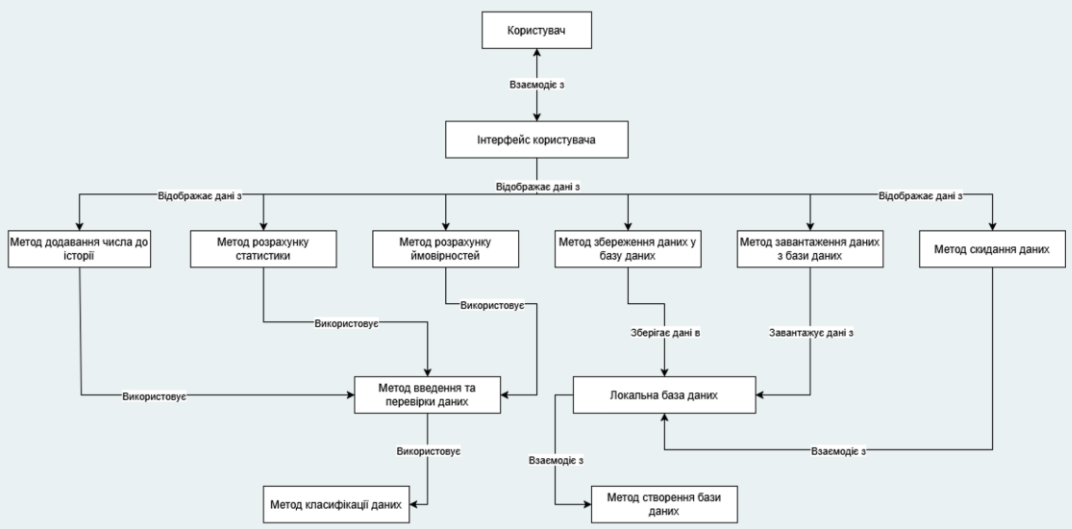


Рисунок Д.9 – Слайд 9

# Діаграма класів

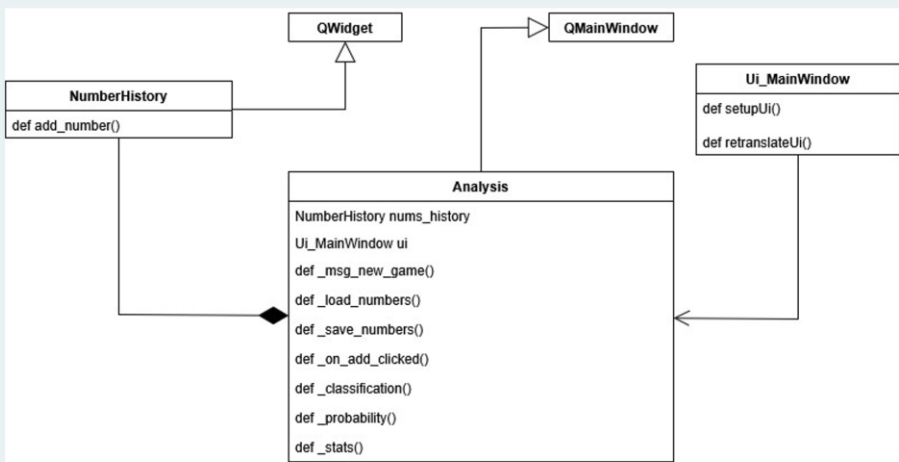


Рисунок Д.10 – Слайд 10

# Приклад роботи системи

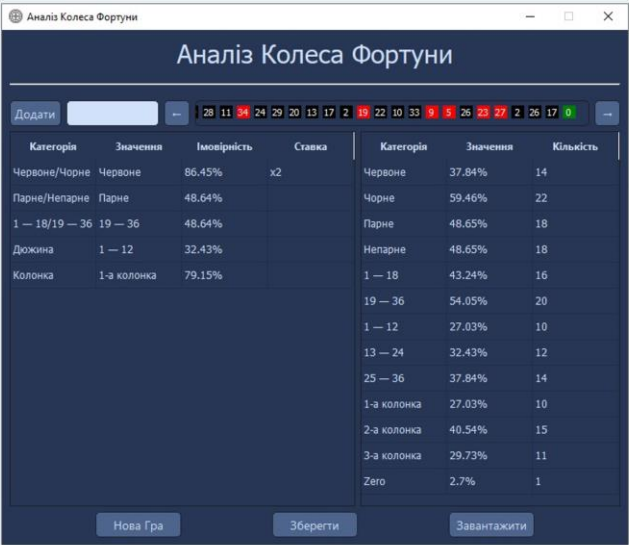


Рисунок Д.11 – Слайд 11

# Тест-кейси програмного застосунку

| Назва                                                                  |                                                | Введення валідного числа                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                  |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| Перевірка коректності додавання до історії та оновлення таблиць чисел: |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| 14                                                                     |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| Передумови                                                             |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| Запустити програму                                                     |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
| №                                                                      | Кроки                                          | Очікуваний результат                                                                                                                                                                                                                                                                                                                                                          |
| 1                                                                      | У текстове поле ввести валідне число (14)      | Число буде відображено у текстовому полі                                                                                                                                                                                                                                                                                                                                      |
| 2                                                                      | Натиснути на кнопку "Додати" або клавішу Enter | Валідне число (14) додано до історії чисел у вигляді червоної мітки;<br>Таблиця ймовірностей оновлена, показує ймовірність 48.64% на категорії "Чорне", "Непарне" та "19 – 36" та 32.43% на категорії "1 – 12" і "1-а колонка";<br>Таблиця статистики оновлена, відображає значення 100% та кількість 1 для категорій "Червоне", "Парне", "1 – 18", "13 – 24", "2-а колонка". |
|                                                                        |                                                |                                                                                                                                                                                                                                                                                                                                                                               |
|                                                                        |                                                |                                                                                                                                                                                                                                                                                                                                                                               |

| Назва                                                                  |                                                | Введення невалідного числа                                                                                                                    |
|------------------------------------------------------------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                  |                                                |                                                                                                                                               |
| Перевірка коректності додавання до історії та оновлення таблиць чисел: |                                                |                                                                                                                                               |
| -5                                                                     |                                                |                                                                                                                                               |
| Передумови                                                             |                                                |                                                                                                                                               |
| Запустити програму                                                     |                                                |                                                                                                                                               |
| №                                                                      | Кроки                                          | Очікуваний результат                                                                                                                          |
| 1                                                                      | У текстове поле ввести валідне число           | Число буде відображено у текстовому полі                                                                                                      |
| 2                                                                      | Натиснути на кнопку "Додати" або клавішу Enter | Буде відображено повідомлення "Помилка. Введіть число від 0 до 36!";<br>Текстове поле буде очищено;<br>Історія чисел і таблиці не змінюються. |
|                                                                        |                                                |                                                                                                                                               |
|                                                                        |                                                |                                                                                                                                               |

Рисунок Д.12 – Слайд 12

# Тест-кейси програмного застосунку

| Назва                                                                      |                                                                                              | Збереження, скидання та завантаження файлу гри                                                                    |
|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                      |                                                                                              |                                                                                                                   |
| Перевірка функції збереження, скидання та завантаження файлу гри з даними: |                                                                                              |                                                                                                                   |
| - 0                                                                        |                                                                                              |                                                                                                                   |
| - 12                                                                       |                                                                                              |                                                                                                                   |
| - 26                                                                       |                                                                                              |                                                                                                                   |
| Передумови                                                                 |                                                                                              |                                                                                                                   |
| Запустити програму                                                         |                                                                                              |                                                                                                                   |
| №                                                                          | Кроки                                                                                        | Очікуваний результат                                                                                              |
| 1                                                                          | У текстове поле ввести валидне число (0)                                                     | Число буде відображено у текстовому полі                                                                          |
| 2                                                                          | Натиснути на кнопку "Додати" або клавішу Enter                                               | Валидне число (0) додано до історії чисел у вигляді зеленої мітки, додано до таблиць ймовірностей та статистики   |
| 3                                                                          | У текстове поле ввести валидне число (12)                                                    | Число буде відображено у текстовому полі                                                                          |
| 4                                                                          | Натиснути на кнопку "Додати" або клавішу Enter                                               | Валидне число (12) додано до історії чисел у вигляді червоної мітки, додано до таблиць ймовірностей та статистики |
| 5                                                                          | У текстове поле ввести валидне число (26)                                                    | Число буде відображено у текстовому полі                                                                          |
| 6                                                                          | Натиснути на кнопку "Додати" або клавішу Enter                                               | Валидне число (26) додано до історії чисел у вигляді жовтої мітки, додано до таблиць ймовірностей та статистики   |
| 7                                                                          | Натиснути кнопку "Зберегти"                                                                  | З'явиться вікно підтвердження дії.                                                                                |
| 8                                                                          | У вікні "Увага. Ви дійсно хочете зберегти гру?" натиснути на кнопку "Yes"                    | Гра буде збережена у файлі saved_games.txt                                                                        |
| 9                                                                          | Натиснути на кнопку "Нова гра"                                                               | З'явиться вікно підтвердження дії.                                                                                |
| 10                                                                         | У вікні "Увага. Ви дійсно хочете почати нову гру?" натиснути на кнопку "Yes"                 | Історія чисел та таблиць очищені.                                                                                 |
| 11                                                                         | Натиснути на кнопку "Завантажити"                                                            | З'явиться вікно підтвердження дії.                                                                                |
| 12                                                                         | У вікні "Увага. Ви дійсно хочете завантажити збережений файл гри?" натиснути на кнопку "Yes" | Числа "0", "12" та "26" відновлено в історії чисел, таблицях ймовірностей та статистики.                          |

| Назва                                                                                 |                                                                                              | Завантаження невалидних даних з файлу збереженої гри                                                                                  |
|---------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Тема:                                                                                 |                                                                                              |                                                                                                                                       |
| Перевірка завантаження даних з файлу збереженої гри "saved_game.txt" вручну з даними: |                                                                                              |                                                                                                                                       |
| - seventeen                                                                           |                                                                                              |                                                                                                                                       |
| Передумови                                                                            |                                                                                              |                                                                                                                                       |
| 1. Відкрити файл "saved_game.txt"                                                     |                                                                                              |                                                                                                                                       |
| 2. Очистити файл від попередніх даних (якщо є)                                        |                                                                                              |                                                                                                                                       |
| 3. Додати відповідні дані                                                             |                                                                                              |                                                                                                                                       |
| 4. Зберегти дані у текстовому файлі                                                   |                                                                                              |                                                                                                                                       |
| №                                                                                     | Кроки                                                                                        | Очікуваний результат                                                                                                                  |
| 1                                                                                     | Натиснути на кнопку "Завантажити"                                                            | З'явиться вікно підтвердження дії                                                                                                     |
| 2                                                                                     | У вікні "Увага. Ви дійсно хочете завантажити збережений файл гри?" натиснути на кнопку "Yes" | Буде відображено помилку з повідомленням: "Помилка. Файл гри містить некоректні дані! Файл не містить дійсних чисел у діапазоні 0-36" |

Рисунок Д.13 – Слайд 13

# Висновки

Результати тестування підтверджують, що розроблене ПЗ успішно виконує всі заплановані функції, зокрема введення та обробку чисел, класифікацію за категоріями, обчислення статистичних показників і ймовірностей, визначення найбільш ймовірних категорій для наступного обертання, а також реалізацію стратегії підрахунку ставок за методом Мартінгейла. Інтерфейс програми виявився зручним, інтуїтивно зрозумілим і приємним у використанні, що робить програму доступною для широкого кола користувачів. Під час тестування програма продемонструвала стабільну роботу, правильну обробку помилок та коректне збереження і завантаження даних. Програма повністю відповідає технічному завданню, показала надійність у роботі та може ефективно використовуватись для аналізу випадковості колеса фортуни.

Рисунок Д.14 – Слайд 14



Рисунок Д.15 – Слайд 15