

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЛЕСІ УКРАЇНКИ**

Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

**ПАДУКА ДАНИЛО АНДРІЙОВИЧ**  
**АНАЛІЗ АЛГОРИТМІВ СТРАТЕГІЙ ВИГРАШУ В ОНЛАЙН-КАЗИНО**  
Спеціальність: 122 Комп'ютерні науки  
Освітньо-професійна програма: Комп'ютерні науки та інформаційні технології  
Робота на здобуття першого рівня «бакалавр»

Науковий керівник:  
**МАМЧИЧ ТЕТЯНА ІВАНІВНА,**  
кандидат фізико-математичних наук, доцент  
кафедри комп'ютерних наук та кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ

Протокол № \_\_\_\_\_  
засідання кафедри комп'ютерних наук  
та кібербезпеки  
від \_\_\_\_\_ 2024 р.  
Завідувач кафедри

(\_\_\_\_\_) Гришанович Т. О.

**ЛУЦЬК – 2024**

## **ЗМІСТ**

### **ВСТУП**3

### **РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АЛГОРИТМІВ СТРАТЕГІЙ ВИГРАШУ В ОНЛАЙН-КАЗИНО**6

#### **1.1 Історія та розвиток азартних ігор**6

#### **1.2 Основні принципи роботи онлайн-казино**7

#### **1.3 Огляд існуючих алгоритмів та стратегій виграшу**8

#### **1.4 Класифікація стратегій гри в казино**11

##### **1.4.1 Алгоритм Мартингейла**14

##### **1.4.2 Алгоритм Лабушера**16

##### **1.4.3 Алгоритм Д'Аламбера**18

### **РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ**20

#### **2.1 Постановка задачі, призначення та вимоги до розробки програмного продукту**20

#### **2.2 Опис проекту**21

#### **2.3 Вибір моделі розробки програмного продукту**22

#### **2.4 Обґрунтування вибору інструментальних засобів розробки програмного продукту**23

#### **2.5 Програмна реалізація алгоритмів**24

#### **2.6 Тестування та налагодження програмної розробки**31

#### **2.7 Аналіз результатів**32

#### **2.8 Рекомендації по впровадженню та використанню програмної розробки**37

### **ВИСНОВКИ**40

### **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**41

### **ДОДАТКИ**43

## ВСТУП

**Актуальність теми.** У сучасному світі онлайн-казино стали однією з найпопулярніших форм розваг, доступних через Інтернет. Вони пропонують гравцям можливість грати в різноманітні азартні ігри, такі як рулетка, покер, блекджек та інші, без необхідності відвідувати фізичні казино. Цей феномен є цікавим як з точки зору соціальної поведінки, так і з економічної перспективи, адже онлайн-казино генерують мільярди доларів доходу щорічно. Зростаюча популярність онлайн-казино обумовлює необхідність дослідження ефективних стратегій гри, що дозволяють гравцям збільшити свої шанси на виграш.

У сфері азартних ігор математичний аналіз відіграє ключову роль. Математичні моделі та статистичні методи дозволяють оцінити ймовірності виграшу та програшу, а також аналізувати стратегії, які можуть підвищити ефективність гри.

Розуміння того, як різні стратегії можуть вплинути на поведінку гравців і їхні фінансові результати, може допомогти у створенні більш безпечного і відповідального ігрового середовища. Це включає розробку програм підтримки для гравців, які можуть мати проблеми з азартними іграми, а також впровадження механізмів контролю та обмеження ставок.

Вивчення стратегій виграшу в онлайн-казино може мати значний вплив на економіку та суспільство. Це допоможе гравцям краще розуміти ризики та потенційні виграші. Соціальний аспект включає питання відповідальної гри та запобігання ігровій залежності.

**Мета роботи.** Метою даної дипломної роботи є аналіз алгоритмів стратегій виграшу в онлайн-казино та розробка програмного продукту для оцінки їх ефективності та можливостей застосування в реальних умовах.

**Завдання дослідження:**

- Проаналізувати літературні джерела та дослідження, присвячені стратегіям гри в казино, математичним моделям та алгоритмам. Це включає аналіз праць з теорії ймовірностей, статистики, математичного моделювання та інших суміжних дисциплін;
- На основі теоретичних досліджень розробити комп'ютерний продукт для симуляції гри. Це включає написання програмного коду для реалізації алгоритмів;
- Провести серію симуляцій для тестування розроблених алгоритмів. Аналіз результатів симуляцій дозволить оцінити ефективність кожної стратегії та порівняти їх між собою;
- На основі проведених досліджень зробити висновки щодо доцільності використання різних стратегій.

**Об'єкт дослідження** – Алгоритми стратегій виграшу в онлайн-казино.

**Предмет дослідження** – Процес розробки та аналізу алгоритмів стратегій виграшу.

**Матеріал дослідження.****Теоретичні аспекти дослідження**

В основі роботи лежить теоретичне дослідження математичних моделей та стратегій, що використовуються в азартних іграх. Було вивчено літературні джерела, наукові статті та праці, присвячені теорії ймовірності, математичному аналізу та їх застосуванню в азартних іграх. Основні аспекти включають:

- Основи теорії ймовірності та її застосування в азартних іграх;
- Математичні моделі алгоритмів стратегій виграшу, таких як Мартингейл, Лабушер та Д'Аламбер;

- Мова програмування C++ для розробки програмного забезпечення;
- Статистичні дані для аналізу результатів тестування стратегій;
- Інструменти для симуляції ігрових процесів та збору результатів.

### **Практичні аспекти дослідження**

Розроблено та реалізовано наступні компоненти:

- Симуляція результатів рулетки;
- Алгоритми стратегій виграшу (Мартингейл, Лабушер, Д'Аламбер);
- Модуль для збору та аналізу результатів ігрових сесій;

# РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АЛГОРИТМІВ СТРАТЕГІЙ ВИГРАШУ В ОНЛАЙН-КАЗИНО

## 1.1 Історія та розвиток азартних ігор

Азартні ігри мають коріння в давніх часах і є невід'ємною частиною культури різних цивілізацій. Перші докази азартних ігор можна знайти в давньому Китаї, Індії, Єгипті та Римі, де вони використовувалися як розважальні заняття та для вирішення спорів.

У середньовіччі азартні ігри стали популярними серед верхівки та знаті, але часто супроводжувалися законодавчими обмеженнями та заборонами. Наприклад, у 15 столітті у Франції було видано декрет, який забороняв гру у карти та кубики.

У 17-18 століттях у Європі з'явилися перші справжні казино, такі як Казино ді Венеція, де гості могли грати в рулетку, фарби та інші ігри. Вони стали місцями розваг та соціальних зустрічей.

У 19 столітті азартні ігри знову зазнали обмежень у багатьох країнах, але це не призвело до повного зникнення інтересу до них. У США популярність азартних ігор різко зросла в епоху Золотої лихоманки, коли багаті люди грали у покер та інші ігри.

У 20 столітті азартні ігри стали глобальною індустрією з мільярдами доларів обороту. Лас-Вегас, який став символом азартного світу, з'явився як місто, де всі мріють випробувати удачу.

### Розвиток онлайн-казино

З появою Інтернету в 1990-х роках азартні ігри отримали нове життя. Перші онлайн-казино з'явилися в 1994 році та невдовзі набули популярності серед ігрових ентузіастів, які тепер могли грати зручно з власного дому.

Спочатку онлайн-казино пропонували обмежений вибір ігор і базувалися на простих графічних інтерфейсах. Проте з розвитком технологій та програмного забезпечення вони стали надавати вражаючу графіку, звукові ефекти та різноманітність ігор.

Мобільні технології революціонізували індустрію азартних ігор, перетворивши телефони та планшети в персональні казино. Гравці можуть зараз грати в будь-який час та з будь-якого місця, що робить гральний процес ще більш доступним та зручним.

Легалізація онлайн-гемблінгу в різних країнах світу відкрила шлях для розвитку інтернет-казино та створила умови для стабільного росту цього сектора. В той же час регулювання цієї галузі стало важливим аспектом для забезпечення чесності та безпеки гравців.

Сучасні онлайн-казино пропонують широкий спектр ігор, включаючи класичні азартні ігри, віртуальні ігрові автомати, спортивні ставки та навіть живі дилерські ігри, що надають аутентичний досвід гри у реальному часі.

Загалом, розвиток онлайн-казино демонструє надзвичайні технологічні досягнення і змінює спосіб, яким люди розуміють та долучаються до світу азартних ігор. Цей сектор продовжує швидко розвиватися, пропонуючи нові можливості та інновації для гравців по всьому світу.

## **1.2 Основні принципи роботи онлайн-казино**

Онлайн-казино - це електронна платформа, яка надає можливість гравцям грати в азартні ігри через Інтернет. Робота таких платформ базується на декількох ключових принципах, які забезпечують їх функціонування та взаємодію з гравцями.

**Ліцензування та регулювання:** Основою для роботи будь-якого онлайн-казино є наявність відповідної ліцензії, яка видається відповідними регуляторними органами. Ліцензія гарантує, що казино відповідає стандартам справедливості та безпеки гравців.

**Програмне забезпечення:** Основою технічного забезпечення онлайн-казино є спеціалізоване програмне забезпечення, розроблене провідними розробниками ігор. Це ПЗ забезпечує роботу всіх ігор, а також управління грошовими потоками та особистими даними гравців.

**Реєстрація та створення облікового запису:** Для гравців необхідно створити обліковий запис, зазвичай з використанням особистої інформації та деталей платіжної системи. Це дає можливість казино забезпечити індивідуальний підхід до кожного гравця.

**Гральні ігри:** Онлайн-казино пропонують широкий вибір азартних ігор, включаючи покер, рулетку, блекджек, ігрові автомати (слоти), баккара, крепс та інші. Кожна гра має свої правила та може мати різні варіанти ставок.

**Генератори випадкових чисел (RNG):** Всі гральні ігри в онлайн-казино використовують генератори випадкових чисел для забезпечення випадковості результатів. Це гарантує справедливість ігор і запобігає будь-яким маніпуляціям чи обманним діям.

**Фінансові транзакції:** Онлайн-казино підтримують різні методи оплати та виведення коштів, такі як кредитні картки, електронні гаманці, банківські перекази тощо. Всі фінансові операції здійснюються через захищені протоколи зв'язку для забезпечення конфіденційності та безпеки.

**Бонуси та промоакції:** Онлайн-казино пропонують різні бонуси та промоакції для залучення нових гравців та стимулювання активних. Це можуть бути вітальні бонуси за перший депозит, без депозитні бонуси, безкоштовні спіни у слотах тощо.

**Клієнтська підтримка:** Онлайн-казино надають клієнтську підтримку для вирішення будь-яких питань або проблем, з якими можуть зіткнутися гравці. Це може бути через онлайн-чат, електронну пошту або телефонну лінію підтримки.

**Мобільні додатки:** Більшість онлайн-казино мають мобільні додатки, які дозволяють гравцям грати на смартфонах і планшетах. Це робить доступ до ігор ще більш зручним та доступним.

**Забезпечення безпеки:** Онлайн-казино використовують різні технології для захисту особистих даних та фінансових транзакцій гравців. Це включає в себе шифрування даних, двофакторну аутентифікацію та інші заходи безпеки.

**Соціальні аспекти:** Онлайн-казино також враховують соціальні аспекти гри, зокрема відповідальну гральну поведінку. Багато платформ пропонують інструменти для самообмеження гравців та попередження проблем із азартністю.

**Регулярні аудити та перевірки:** Більшість онлайн-казино регулярно піддаються аудитам і перевіркам незалежних організацій, які переконуються у чесності і випадковості ігор.

### **1.3 Огляд існуючих алгоритмів та стратегій виграшу**

Стратегії виграшу в азартних іграх мають давню історію, яка сягає античних часів. Відомо, що люди завжди прагнули підвищити свої шанси на



виграш, розробляючи різні методиками для азартних ігор. У давньому Китаї та Римській імперії були спроби систематизувати підхід до гри в кості та інші азартні ігри. Проте більш науковий підхід до розробки стратегій з'явився в Європі у XVIII столітті.

У XVIII столітті у Франції з'явилися такі стратегії, як Мартингейл, яка стала популярною серед гравців у рулетку. Власники казино та математики почали вивчати ймовірності та розробляти системи для підвищення шансів на виграш. Джон Генрі Мартингейл, британський казино-власник, сприяв популяризації стратегії, яка базувалася на подвоєнні ставки після кожного програшу.

Перші документовані випадки використання стратегій виграшу відносяться до XIX століття, коли математики і гравці почали систематично вивчати азартні ігри та розробляти стратегії. Наприклад, Жан ле Ронд Д'Аламбер, французький математик, запропонував свою систему ставок, засновану на теорії ймовірності та більш консервативному підході порівняно з Мартингейлом. Інші системи, такі як система Лабушера, також виникли в цей період і стали популярними серед гравців.

Розглянемо деякі з існуючих алгоритмів та стратегій виграшу:

**Система Фібоначчі:** базується на числовій послідовності, яку вперше описав Леонардо Пізанський (Фібоначчі) у 13 столітті. У контексті азартних ігор, вона використовується для визначення ставок, де кожна нова ставка є сумою двох попередніх. Вперше система Фібоначчі була застосована в азартних іграх у Італії та Франції в середньовіччі. Гравець збільшує ставку після програшу та зменшує після перемоги, використовуючи числа Фібоначчі. Ця стратегія часто використовується в рулетці та інших іграх, де можна робити зовнішні ставки з двох варіантів (червоне/чорне, парне/непарне).

**Система Парлай:** виникла у XIX столітті та використовується для послідовного збільшення або зменшення ставок в залежності від результатів попередньої гри. Поширення цієї стратегії спостерігалось в Англії, Франції та інших європейських країнах у XIX столітті. Гравець використовує виграшну суму на наступну ставку, зберігаючи початкову ставку, якщо він виграє. При програші, ставка повертається до початкового рівня. Ця стратегія популярна серед гравців у блекджек і баккару.

**Система 1-3-2-6:** виникла як спроба структурувати ставки в азартних іграх, зокрема в баккара. Спочатку відомі випадки використання системи 1-3-2-

б з'явилися в казино у Франції в XVIII столітті. Гравець встановлює певну структуру ставок: перша ставка - 1 одиниця, друга - 3 одиниці, третя - 2 одиниці, четверта - 6 одиниць. При перемозі гравець повертається до першої ставки. Ця стратегія часто використовується в іграх, де є можливість зробити послідовні ставки з певними коефіцієнтами виграшу, наприклад, в баккара чи рулетці.

**Система Оскара Грінда:** отримала свою назву від американського математика Оскара Грінда, який розробив її в 1960-х роках. Ця система ставок призначалася для використання в рулетці. Популярність системи Оскара Грінда зросла в казино Лас-Вегаса в період зростання інтересу до азартних ігор у США. Заснований на принципі послідовного збільшення ставок після програшу та зменшення після виграшу, гравець використовує фіксовані рівні ставок, змінюючи їх в залежності від результатів попередніх спроб. Головним чином призначений для використання в рулетці, де гравець може робити ставки на червоне/чорне або інші подібні зовнішні ставки.

**Система Червоне/Чорне:** є однією з найпростіших та найпоширеніших серед азартних гравців. Її використання почалося ще в XVIII столітті, коли виникла популярність рулетки в Європі. Вперше систему Червоне/Чорне почали застосовувати в казино у Франції та Італії. Гравець робить ставку на червоне або чорне, яке має майже 50% ймовірність виграшу. Ця стратегія базується на принципі рівних шансів і відсутності залежності від попередніх результатів. Використовується в рулетці для зовнішніх ставок на один з двох можливих результатів (червоне або чорне).

**Система Паролі:** виникла у Франції в XVIII столітті і стала популярною серед азартних гравців, що шукали систему, яка дозволяє виграти в кратність від своєї початкової ставки. Поширення цієї системи відбулося в період популярності рулетки та інших азартних ігор у Франції та Європі. Гравець встановлює послідовність ставок, де кожна наступна ставка залежить від результату попередньої. При виграші, ставка збільшується в кратність від початкової ставки, а при програші повертається до початкового рівня. Використовується в різних іграх казино, де можна робити послідовні ставки на різні результати.

**Система Голландської рулетки:** отримала свою назву від Голландського художника, який її розробив у середині XX століття. Вона була спроектована для використання в рулетці та визначення стратегії ставок. Популярність

системи почала зростати серед гравців рулетки в Європі та США у другій половині ХХ століття. Гравець робить ставку на одне з чисел (або групу чисел), які є улюбленими чи особливими для нього, і утримується від зміни цієї стратегії протягом гри. Використовується в рулетці, особливо в залежності від особистих вподобань та переконань гравця.

**Система Колонок:** виникла в середині ХІХ століття та використовується в рулетці для розподілу ставок на колонки чисел на ставках на три різних ряди. Ця стратегія ставок стала популярною серед гравців у Франції та Європі у ХІХ столітті. Гравець робить ставки на одну з трьох колонок чисел (1-12, 13-24, 25-36) з однаковими коефіцієнтами виграшу, що дає йому шанси виграти приблизно  $1/3$  кожного оберту. Використовується в рулетці для ставок на визначені групи чисел, що дозволяє гравцеві розподілити свої ставки для покриття більшої частки виграшів.

**Система Лоуренса:** виникла в середині ХХ століття і була розроблена для використання в різних карточних іграх, таких як блекджек і покер. Популярність системи Лоуренса зросла серед азартних гравців у США та Європі у другій половині ХХ століття. Гравець використовує стратегію зменшення або збільшення ставок в залежності від результатів попередніх роздач або обертів, застосовуючи стратегічні зміни, щоб збільшити свої шанси на виграш. Застосовується в карточних іграх, де гравці можуть контролювати свої ставки і змінювати їх залежно від карти, яку вони отримали або їхнього поточного стану гри.

#### 1.4 Класифікація стратегій гри в казино

Азартні ігри завжди привертали увагу людей, які прагнуть випробувати свою удачу і виграти великі суми грошей. З часом гравці почали розробляти різні стратегії, спрямовані на підвищення своїх шансів на успіх. В цій статті розглянемо класифікацію стратегій гри в казино, основні підходи та принципи, що лежать в їх основі.

**Стратегії для різних видів ігор:**

## 1. Рулетка:

- **Мартингейл:** Одна з найвідоміших стратегій, яка передбачає подвоєння ставки після кожного програшу, щоб при виграші компенсувати всі попередні втрати і отримати прибуток, рівний початковій ставці.
- **Д'Аламбер:** Поступове збільшення ставки на одну одиницю після програшу і зменшення на одну одиницю після виграшу.
- **Фібоначчі:** Використання числової послідовності Фібоначчі для визначення розміру ставки. Після програшу ставка збільшується за наступним числом у послідовності, а після виграшу повертається на дві позиції назад.

## 2. Блекджек:

- **Стратегія базової гри:** Використання математично обґрунтованих рішень для кожної можливої комбінації карт.
- **Лічильник карт:** Відстеження карт, що вже були використані, щоб передбачити ймовірність отримання вигідних карт у наступних раундах.
- **Система Паролі:** Після кожного виграшу гравець подвоює свою ставку.

## 3. Покер:

- **Техаський холдем:** Включає різні стратегії, залежно від позиції гравця, розміру стеку, стилю гри опонентів тощо. Однією з ключових стратегій є агресивна гра з сильними руками і обережна з слабкими.
- **Омаха:** Вимагає уважного підходу до вибору стартових рук, а також розуміння ймовірностей покращення своєї руки після флопу.
- **Блеф:** Важливий компонент покерної стратегії, який включає обман опонентів щодо сили своєї руки.

## 4. Баккара:

- **Система 1-3-2-6:** Передбачає, що гравець збільшує ставки за певною схемою.

- **Система Паролі:** Подібна до системи, що використовується у блекджеку, де ставки подвоюються після кожного виграшу.

## 5. Слоти:

- **Управління банкролом:** Визначення фіксованого бюджету для гри та дотримання його незалежно від виграшів чи програшів.
- **Частота виплат:** Вибір ігор з високою частотою виплат або з прогресивними джекпотами для збільшення шансів на виграш.

## Основні підходи та принципи:

### 1. Управління банкролом:

- Один з ключових аспектів успішної гри в казино. Гравці повинні розподіляти свої ставки таким чином, щоб уникнути великих втрат і зберегти достатньо коштів для продовження гри.
- Встановлення лімітів на виграші та програші допомагає контролювати свої емоції і уникати необдуманих рішень.

### 2. Математичний підхід:

- Використання математичних моделей і теорії ймовірностей для розрахунку оптимальних ставок і прийняття рішень у грі.
- У блекджеку та покері цей підхід дозволяє гравцям зменшити перевагу казино і збільшити свої шанси на виграш.

### 3. Психологія гри:

- Розуміння поведінки своїх опонентів і використання психологічних прийомів для впливу на їхні рішення.
- Вміння контролювати свої емоції і залишатися спокійним у будь-яких ситуаціях.

#### 4. Стратегічне планування:

- Визначення чітких цілей і розробка плану гри на основі аналізу своїх сильних і слабких сторін.
- Постійне вдосконалення своїх навичок і адаптація стратегій залежно від змін у грі.

#### 5. Дисципліна:

- Дотримання вибраної стратегії і уникнення імпульсивних рішень, які можуть призвести до втрат.
- Встановлення чітких правил для себе і їхнє неухильне дотримання.

Стратегії гри в казино дуже різноманітні і залежать від конкретної гри та особистих вподобань гравця. Незважаючи на різноманіття підходів, основні принципи залишаються незмінними: математичний аналіз, управління банкролом, психологія гри і стратегічне планування. Вивчення і застосування цих принципів допоможе гравцям підвищити свої шанси на успіх і зменшити ризику під час гри в казино.

Більш детально розглянемо такі алгоритми (Мартингейла, Лабушера, Д'Аламбера) через їхню історичну значимість, популярність серед гравців та унікальні підходи до управління ставками, що робить їх цікавими для аналізу ефективності в грі в рулетку. Вивчення цих алгоритмів дозволяє глибше зрозуміти їх механізми, ефективність та обмеження, що є важливим для аналізу стратегій в азартних іграх.

##### 1.4.1 Алгоритм Мартингейла

Алгоритм Мартингейла є однією з найбільш відомих і часто використовуваних стратегій в азартних іграх, особливо в сферах з майже рівною ймовірністю виграшу або програшу, наприклад, у рулетці або блекджеку. Його принцип досить простий: після кожної програшної ставки гравець подвоює свою наступну ставку з метою компенсації попередніх збитків і отримання прибутку, який дорівнює початковій ставці. Назва стратегії походить від французького математика Поля П'єра Леві Мартингейла, який вивчав теорію ймовірностей у середині ХХ століття.

Основні переваги алгоритму Мартингейла включають його простоту та інтуїтивність, яка дозволяє гравцям легко розуміти логіку застосування. Вона здається привабливою через можливість швидко відновити втрати та здобути невеликий, але стабільний прибуток, особливо у короткостроковій перспективі.

Проте, стратегія Мартингейла не є без ризиковою. Основними недоліками є відсутність верхнього ліміту для ставок, що може призвести до значних втрат при тривалій серії програшів. Ця стратегія також не змінює ймовірність виграшу в окремій ставці, оскільки кожна ставка є незалежною. Головний ризик полягає в тому, що при великій кількості програшів гравець може вичерпати свій банкролл.

Варіації стратегії Мартингейла можуть включати більш складні системи, які враховують максимальні ліміти ставок або зміну розмірів ставок залежно від результатів. Деякі гравці використовують цю стратегію не лише в азартних іграх, але і в інших сферах життя, де керування ризиками та ставками відіграє важливу роль у досягненні успіху.

У сучасному гральному середовищі багато онлайн-казино мають політики обмеження або заборони на використання стратегії Мартингейла, що може вплинути на її ефективність і відповідальне гральне поведінка. Тим не менш, вона залишається однією з найвідоміших і досліджуваних стратегій серед гравців, що продовжують використовувати її в своїй грі.

**Принцип роботи:** Припустимо, що гравець починає зі ставки 10 одиниць. Якщо він програє, наступна ставка буде 20 одиниць. Якщо знову програє, наступна ставка буде 40 одиниць, і так далі. Коли гравець виграє, він отримає прибуток, рівний його початковій ставці (у цьому випадку 10 одиниць). Після виграшу гравець повертається до початкової ставки і починає процес знову.

**Приклад:**

- Ставка 1: 10 одиниць, результат програш;
- Ставка 2: 20 одиниць, результат програш;
- Ставка 3: 40 одиниць, результат програш;
- Ставка 4: 80 одиниць, результат виграш;
- Ставка 5: 10 одиниць, результат виграш.

Гравець виграє після четвертої ставки і отримує 10 одиницю прибутку, збільшуючи свій баланс.

**Плюси:**

1. **Простота і логіка:** Алгоритм легко розуміти і застосовувати, оскільки його принципи не потребують складних обчислень чи спеціальних знань.
2. **Швидкий виграш:** У короткострокових сесіях гравець може швидко виграти і зробити прибуток, який дорівнює його початковій ставці.
3. **Відновлення збитків:** При виграші гравець компенсує всі попередні програші, що робить стратегію привабливою для тих, хто прагне мінімізувати втрати.

### **Мінуси:**

1. **Ризик великих втрат:** При тривалих серіях програшів ставки швидко зростають, що може призвести до значних фінансових втрат.
2. **Обмеження казино:** Більшість казино встановлюють максимальні ставки, що обмежує можливість безкінечного подвоєння і може призвести до нестабільної роботи стратегії.
3. **Великий банк потрібен:** Для успішного використання алгоритму потрібен значний банк, щоб покрити тривалі серії програшів.
4. **Психологічний стрес:** Часті програші і необхідність постійно збільшувати ставки можуть викликати психологічний стрес у гравця.

### **1.4.2 Алгоритм Лабушера**

Алгоритм Лабушера є одним із відомих методів ставок, який використовується в азартних іграх, таких як рулетка або блекджек. Його назва походить від французького гравця Лабушера, який жив у XVIII столітті і використовував цей метод для гри в рулетку. Алгоритм Лабушера базується на послідовному змінненні ставок залежно від результатів попередніх спроб з метою здобуття прибутку і управління ризиками.

Існують різноманітні варіації алгоритму Лабушера, які включають зміну стратегії в залежності від результатів гри або використання різних початкових послідовностей чисел. Деякі гравці модифікують стратегію, щоб адаптувати її до конкретної азартної гри або особистих уподобань.

Сучасне інтернет-казино зазвичай не забороняє використання алгоритму Лабушера, однак існують правила, які обмежують максимальні ставки або змінюють правила гри, що можуть вплинути на ефективність стратегії. Гравцям слід завжди звертатися до правил конкретного казино перед застосуванням алгоритму Лабушера в своїй грі.



**Принцип роботи:** Алгоритм Лабушера базується на списку чисел, які гравець записує перед початком гри. Кожне число в списку представляє частину загального бажаного виграшу. Наприклад, гравець може записати наступний список чисел: 1, 2, 3, 4, 5. Сума всіх чисел у списку (в даному випадку 15) представляє загальний бажаний виграш. Гравець починає з суми першого і останнього чисел у списку. Якщо ставка виграє, ці числа викреслюються зі списку. Якщо ставка програє, сума додається в кінці списку. Гравець продовжує, поки всі числа не будуть викреслені.

### Приклад:

- Список чисел: 1, 2, 3, 4, 5
- Перша ставка:  $1 + 5 = 6$  (результат програш, додати 6 в кінець списку)
- Новий список чисел: 1, 2, 3, 4, 5, 6
- Друга ставка:  $1 + 6 = 7$  (результат виграш, викреслити 1 і 6)
- Новий список чисел: 2, 3, 4, 5

Гравець продовжує, доки всі числа не будуть викреслені.

### Плюси:

1. **Контроль над ставками:** Гравець може налаштувати список чисел відповідно до свого бажаного виграшу і ризиків.
2. **Гнучкість:** Алгоритм Лабушера дозволяє легко коригувати ставки залежно від виграшів і програшів.
3. **Менший ризик:** Порівняно з Мартингейлом, ця стратегія менш ризиківана, оскільки ставки не зростають так швидко.

### Мінуси:

1. **Складність:** Алгоритм складніший для розуміння і застосування, ніж інші стратегії.
2. **Ризик тривалих серій програшів:** Як і будь-яка інша стратегія, Лабушер не гарантує виграшу і може призвести до великих втрат при тривалих серіях програшів.
3. **Великий банк потрібен:** Хоча ставки не зростають так швидко, гравець все ще потребує значного банку для покриття можливих програшів.

Загалом, алгоритм Лабушера залишається однією з популярних і стратегічно обґрунтованих методик ставок в азартних іграх, яка вимагає розуміння принципів управління ризиками і готовності до варіацій в гральному процесі.

### 1.4.3 Алгоритм Д'Аламбера

Алгоритм Д'Аламбера є одним з відомих методів управління ставками в азартних іграх, таких як рулетка, блекджек чи баккара. Цей алгоритм названий на честь відомого французького математика та гравця Жан-Ле Рон Д'Аламбера, який жив у XVIII столітті.

Історія алгоритму Д'Аламбера сягає XVIII століття, коли його було розроблено як спробу математично обґрунтувати успіх у азартних іграх. Ця стратегія набула популярності завдяки своїй простоті та логічному підходу до управління ставками. Незважаючи на певні обмеження, алгоритм Д'Аламбера залишається однією з найпоширеніших систем серед гравців, які шукають методи мінімізації ризиків та збільшення шансів на виграш.

**Принцип роботи:** Алгоритм Д'Аламбера передбачає збільшення ставки на одну одиницю після кожного програшу і зменшення на одну одиницю після кожного виграшу. Основна ідея полягає в тому, що після програшу гравець збільшує свою ставку. Це робиться з метою балансування виграшів і втрат у тривалій гральній сесії, а також для максимізації можливого прибутку.

#### Приклад:

- Початкова ставка: 1 одиниця
- Перша ставка: 1 результат програш;
- Друга ставка: 2 результат програш;
- Третя ставка: 3 результат виграш;
- Четверта ставка: 2 результат виграш;
- П'ята ставка: 1 результат виграш.

В цьому прикладі гравець починає з 1 одиниці, після двох програшів збільшує ставки до 3 одиниць, виграє і зменшує ставки до 2 одиниць, а потім до 1 одиниці. В результаті він залишається з невеликим прибутком.

#### Плюси:

1. **Простота:** Алгоритм Д'Аламбера легкий для розуміння і застосування, не потребує складних обчислень.
2. **Менший ризик:** Порівняно з Мартингейлом, ставки збільшуються і зменшуються поступово, що зменшує ризик великих втрат.
3. **Відновлення збитків:** Система допомагає гравцям поступово відновлювати втрати, не піддаючись великим коливанням у ставках.

### **Мінуси:**

1. **Повільне відновлення:** При тривалих серіях програшів відновлення збитків може бути повільним, що може вимагати терпіння від гравця.
2. **Відсутність гарантій:** Як і будь-яка інша стратегія ставок, алгоритм Д'Аламбера не гарантує виграшу і може призвести до збитків у довгостроковій перспективі.
3. **Необхідність дисципліни:** Гравці повинні чітко дотримуватися стратегії і не піддаватися емоціям, що може бути складно під час гри.

Також варто враховувати, що багато сучасних інтернет-казино можуть мати обмеження на використання алгоритмів управління ставками, включаючи алгоритм Д'Аламбера. Гравцям рекомендується завжди дотримуватися правил конкретного казино та не використовувати заборонені стратегії.

Алгоритм Д'Аламбера підходить для гравців, які віддають перевагу більш консервативному підходу до управління ставками. Важливо розуміти, що жодна стратегія не може гарантувати стабільного прибутку в азартних іграх, і відповідальне управління банком є ключовим фактором для успішної гри. Гравці повинні ретельно зважувати всі плюси і мінуси цієї стратегії перед її застосуванням, щоб забезпечити максимальний успіх і мінімізувати можливі втрати.

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

### 2.1 Постановка задачі, призначення та вимоги до розробки програмного продукту

Основною задачею даної роботи є розробка програми для симуляції ставок у рулетці з використанням різних стратегій. Програма допоможе користувачам зрозуміти принципи роботи стратегій ставок через інтерактивну симуляцію. Програма має забезпечувати наочне відображення результатів кожної ставки та дозволити користувачам детально вивчити принципи роботи стратегій.

Програма для симуляції ставок у рулетці призначена для використання дослідниками, викладачами та студентами. Вона буде корисним інструментом у навчальному процесі, допомагаючи покращити розуміння теоретичних аспектів стратегій ставок. Програма також забезпечить візуальне представлення процесу ставок, що сприятиме кращому засвоєнню матеріалу, дозволить користувачам експериментувати з різними стратегіями та параметрами для глибшого розуміння їхньої роботи.

Основними вимогами до розробки є:

- Програма повинна підтримувати симуляцію основних стратегій ставок, таких як Мартингейл, Лабушер та Д'Аламбер.
- Користувачі повинні мати можливість налаштовувати параметри симуляції, такі як початковий баланс, початкова ставка, максимальна ставка та кількість ставок.
- Програма має підтримувати інтерактивне відображення результатів кожної ставки, включаючи номер ставки, розмір ставки та результат (виграш чи програш).
- Інтуїтивно зрозумілий і зручний інтерфейс, який забезпечить легкий доступ до всіх функцій програми.
- Програма повинна бути достатньо продуктивною для швидкої симуляції великої кількості ставок.
- Програма має бути розроблена з урахуванням можливості її запуску на різних операційних системах (Windows, macOS, Linux).

Для забезпечення виконання поставлених задач, програма повинна включати в себе наступні функції:

- Підтримка різних стратегій ставок.
- Можливість вибору стратегії для симуляції.

- Показ результату кожної ставки у реальному часі.
- Опції для налаштування параметрів симуляції, таких як початковий баланс, початкова ставка, максимальна ставка та кількість ставок.
- Виведення детальної історії ставок, включаючи номер ставки, розмір ставки та результат.
- Зручний та інтуїтивно зрозумілий інтерфейс для взаємодії з програмою.
- Підтримка роботи на різних операційних системах, таких як Windows, macOS та Linux.
- Забезпечення високої продуктивності та оптимальної швидкості виконання симуляції, незалежно від платформи.

## 2.2 Опис проекту

Проект спрямований на розробку програмного забезпечення для симуляції різних стратегій гри в рулетку. Основною метою програми є надання можливості користувачеві вибирати одну з трьох реалізованих стратегій (Мартингейл, Лабушера, Д'Аламбера), введення параметрів гри і отримання результатів симуляції. Ось детальний опис проекту:

### **Архітектура програмного забезпечення:**

- Програма реалізована на мові програмування C++ з використанням стандартних бібліотек.
- Вона складається з головної функції main(), що керує усіма аспектами введення даних від користувача, вибору стратегії та виведення результатів.
- Кожна стратегія гри (Мартингейл, Лабушера, Д'Аламбера) реалізована як окрема функція зі своєю логікою і обробкою ставок.

### **Функціональні можливості програми:**

- Користувач має можливість ввести початковий баланс, початкову ставку, максимальну ставку і кількість ставок для симуляції.
- Програма дозволяє вибирати одну з трьох стратегій гри: Мартингейл, Лабушера або Д'Аламбера, кожна з яких має свої унікальні принципи управління ставками.
- Після завершення симуляції програма виводить історію ставок для обраної стратегії, результати кожної ставки (виграш або програш) і оновлений баланс банкролу.

## 2.3 Вибір моделі розробки програмного продукту

Зважаючи на вибір мови програмування C++ для розробки програми симуляції гри в рулетку, важливо врахувати кілька ключових аспектів при виборі моделі розробки. Використання C++ є обґрунтованим з урахуванням його ефективності та потужності для реалізації складних алгоритмів. Програма реалізована за допомогою модульних функцій, що сприяє чіткості і легкості розширення коду, використовуючи стандартні бібліотеки `iostream` та `vector` для спрощення взаємодії з користувачем і маніпулювання даними.

Проект базується на ітеративному підході, що дозволяє поетапно розширювати функціонал та вдосконалювати програму, а також використовує модель інкрементальної розробки для ефективного тестування і відлагодження. Гнучкий підхід до розробки дозволяє програмі швидко адаптуватися до змін у вимогах або виявлених можливостях, забезпечуючи при цьому управління потенційними ризиками для стабільності і надійності програми.

Детальне документування коду та включення необхідних коментарів сприяють зручності підтримки та розумінню програмного продукту. Кожна функція програми, включаючи функцію для симуляції результату рулетки (`spin_wheel`), алгоритми стратегій гри (`martingale_strategy`, `labouchere_strategy`, `d_alembert_strategy`), а також функцію для виведення історії ставок (`print_bet_history`), має чітко визначені обов'язки і є самодостатньою для виконання конкретної задачі.

Структура програми використовує стандартні підходи до обробки даних, такі як використання векторів для зберігання і керування послідовностями ставок і історією гри. Це забезпечує ефективне управління пам'яттю і зручність в обробці великої кількості даних. Використання структур даних, таких як структура `Bet` для зберігання інформації про кожну ставку, дозволяє зберігати і зберігати усю необхідну інформацію компактно та ефективно.

Вибір ітераційної моделі також сприяє зниженню ризику введення помилок, оскільки кожен етап розробки піддається детальному тестуванню перед переходом до наступного.

Окрім того, гнучкість управління змінами, забезпечена ітераційною моделлю, дозволяє ефективно керувати ризиками, пов'язаними з розробкою програмного забезпечення. Процес ітераційної розробки дозволяє вчасно виявляти та виправляти помилки, забезпечуючи тим самим підвищену надійність і якість програмного продукту. Кожен етап розробки, включаючи планування, аналіз, дизайн, реалізацію і тестування, піддається детальному контролю і валідації перед переходом до наступного етапу.

## 2.4 Обґрунтування вибору інструментальних засобів розробки програмного продукту

Обґрунтування вибору Visual Studio Code як інструментального засобу для розробки програмного забезпечення полягає в кількох ключових аспектах, які сприяють покращенню продуктивності, ефективності та комфорту розробників. Початковою перевагою є безкоштовність та відкритий вихідний код VS Code, що робить його доступним для широкого кола користувачів без витрат на ліцензії. Цей редактор підтримує багато мов програмування, включаючи такі популярні, як C++, Python, JavaScript, Java та інші, що робить його універсальним інструментом для різних проектів.

Інтегрована підтримка Git є ще однією важливою особливістю VS Code, оскільки вона спрощує керування версіями коду, спільну роботу над проектами та відслідковування змін, що робить його відмінним інструментом для роботи в команді. Для полегшення роботи з Git доступні різноманітні розширення та плагіни, що дозволяють виконувати більш складні операції, такі як злиття гілок та вирішення конфліктів.

Розширюючи можливості VS Code, великий вибір розширень дозволяє налаштовувати редактор під індивідуальні потреби кожного розробника. Наявність розширень для відлагодження коду, автоматизації рутинних завдань, роботи з різними фреймворками та іншими інструментами значно підвищує продуктивність розробників і спрощує розробку складних програмних продуктів.

Окрім того, Visual Studio Code володіє інтерфейсом, який є дуже зручним та інтуїтивно зрозумілим, що сприяє швидкому освоєнню новими користувачами. Мінімалістичний дизайн і зосередження на основних функціях зробили VS Code одним з найпопулярніших редакторів серед розробників усього світу. Його низьке споживання ресурсів та висока швидкість роботи зробили його ідеальним вибором для використання на різних платформах і пристроях.

Вбудована підтримка Docker дозволяє зручно розгортати та тестувати програми в контейнерах, що особливо важливо для розробників, які працюють у середовищах з мікросервісною архітектурою. Інтегровані інструменти для автоматичного перевірки коду та підтримки розробки за допомогою розширень забезпечують високу якість і надійність коду, сприяючи швидкому виявленню та усуненню помилок.

Загальна підтримка спільноти користувачів VS Code дозволяє швидко вирішувати труднощі та обмінюватися досвідом, що сприяє постійному вдосконаленню робочих процесів і розширенню функціональних можливостей

редактора. Ці аспекти роблять Visual Studio Code відмінним інструментом для будь-якого проекту розробки програмного забезпечення, незалежно від його масштабів і складності.

## 2.5 Програмна реалізація алгоритмів

Ця програма моделює гру в рулетку і реалізує три різні стратегії управління ставками: Мартингейл, Лабушера і Д'Аламбера. Розглянемо кожну частину програми детальніше:

### Функція `spin_wheel()`:

Ця функція симулює результат обертання рулетки, де випадкове число визначається функцією `rand()` і конвертується в дійсне число від 0 до 1. Ймовірність виграшу чорного кольору встановлюється на 18/37.

```
// Функція для симуляції результату рулетки
string spin_wheel() {
    // Вірогідності виграшу чи програшу для чорного і червоного кольорів
    double black_win_probability = 18.0 / 37.0; // Чорний: 18 виграшних номерів з 37

    double random_number = (double)rand() / RAND_MAX;

    if (random_number < black_win_probability) {
        return "black"; // Виграш
    } else {
        return "red"; // Програш
    }
}
```

Рис 2.1 – Функція для симуляції результату рулетки

### Структура `Bet`:

Використовується для зберігання інформації про кожну ставку: номер ставки, її розмір і результат (win або lose).



```

struct Bet {
    int number;
    int amount;
    string result;
};

```

Рис 2.2 – Структура для зберігання інформації про кожну ставку

### Алгоритм Мартингейл (`martingale_strategy`):

Функція `martingale_strategy` реалізує алгоритм Мартингейла. Вона приймає чотири параметри: початковий баланс гравця (`bankroll`), початкова сума ставки (`initial_bet`), максимальна допустима ставка (`max_bet`) та кількість ставок, яку гравець планує зробити (`num_bets`). Функція повертає пару значень: поточний баланс після всіх ставок та історію всіх ставок у вигляді вектора структур `Bet`.

Алгоритм роботи функції наступний. Спочатку ініціалізується розмір ставки (`bet`) як початкова ставка (`initial_bet`). Створюється вектор для зберігання історії ставок (`bet_history`). Запускається цикл, який виконується до досягнення кількості ставок (`num_bets`). На кожній ітерації перевіряється, чи достатньо коштів для здійснення ставки. Якщо баланс менший за ставку, цикл переривається.

Далі ініціалізується структура для поточної ставки, зберігаючи її номер і суму. Виконується симуляція результату обертання рулетки за допомогою функції `spin_wheel`. Якщо результатом є "black" (виграш), баланс гравця збільшується на суму ставки, результат зберігається як "win", і розмір наступної ставки повертається до початкової ставки. Якщо результатом є "red" (програш), баланс зменшується на суму ставки, результат зберігається як "lose", і розмір наступної ставки подвоюється.

Якщо розмір ставки перевищує максимальну ставку, він обмежується значенням `max_bet`. Інформація про поточну ставку додається до історії ставок. Після завершення циклу функція повертає остаточний баланс гравця і історію всіх ставок.

Таким чином, функція реалізує стратегію Мартингейла, враховуючи обмеження на максимальну ставку та перевіряючи наявність достатньої кількості коштів у гравця для виконання наступної ставки.

```

// Алгоритм Мартингейл
pair<int, vector<Bet>> martingale_strategy(int bankroll, int initial_bet,
int max_bet, int num_bets) {
    int bet = initial_bet;
    vector<Bet> bet_history;
    for (int i = 0; i < num_bets; ++i) {
        if (bankroll < bet) break;
        Bet current_bet;
        current_bet.number = i + 1; // Нумерація ставок
        current_bet.amount = bet;
        if (spin_wheel() == "black") {
            bankroll += bet;
            current_bet.result = "win";
            bet = initial_bet;
        } else {
            bankroll -= bet;
            current_bet.result = "lose";
            bet *= 2;
            // Обмеження на максимальну ставку
            bet = min(bet, max_bet);
        }
        bet_history.push_back(current_bet);
    }
    return {bankroll, bet_history};
}

```

Рис 2.3 – Алгоритм Мартингейл

**Стратегія Лабушера (labouchere\_strategy):**

Функція `labouchere\_strategy` реалізує стратегію ставок Лабушера. Функція приймає п'ять параметрів: початковий баланс гравця (`bankroll`), поточну послідовність чисел для визначення розміру ставок (`sequence`), максимальну допустиму ставку (`max\_bet`), кількість ставок, яку планується зробити (`num\_bets`), і початкову послідовність чисел для відновлення після завершення послідовності (`initial\_sequence`). Якщо поточна послідовність чисел порожня, вона відновлюється до початкової. Розмір ставки визначається як сума першого та останнього елементів послідовності.

Функція ініціалізує вектор для зберігання історії ставок по такому самому принципу, як з алгоритмом Мартингейла.

Таким чином, `labouchere\_strategy` реалізує стратегію Лабушера, враховуючи обмеження на максимальну ставку та перевіряючи наявність

достатньої кількості коштів у гравця для виконання наступної ставки.

```
// Стратегія Лабушера
pair<int, vector<Bet>> labouchere_strategy(int bankroll, vector<int> sequence,
int max_bet, int num_bets, const vector<int>& initial_sequence) {
    vector<Bet> bet_history;
    for (int i = 0; i < num_bets; ++i) {
        if (sequence.empty()) {
            sequence = initial_sequence; // Відновлюємо початкову послідовність
        }

        int bet = sequence.front() + sequence.back(); // Розрахунок ставки
        if (bet > max_bet) bet = max_bet; // Обмеження на максимальну ставку

        if (bankroll < bet) break; // Перевірка, чи вистачає коштів

        Bet current_bet;
        current_bet.number = i + 1; // Нумерація ставок
        current_bet.amount = bet;

        if (spin_wheel() == "black") {
            bankroll += bet;
            current_bet.result = "win";
            // Виграш: видалення першого та останнього елементів послідовності
            if (sequence.size() > 1) {
                sequence.erase(sequence.begin());
                sequence.pop_back();
            } else {
                sequence.clear(); // Якщо залишився один елемент
            }
        } else {
            bankroll -= bet;
            current_bet.result = "lose";
            // Програш: додавання ставки в кінець послідовності
            sequence.push_back(bet);
        }
        bet_history.push_back(current_bet);
    }
    return {bankroll, bet_history};
}
```

Рис 2.4 – Стратегія Лабушера

### Стратегія Д'Аламбера (d\_alembert\_strategy):

Функція d\_alembert\_strategy реалізує стратегію Д'Аламбера для азартних ігор, таких як рулетка. Ця стратегія базується на поступовому збільшенні ставки після програшу та зменшенні ставки після виграшу, що має на меті забезпечити стабільний приріст капіталу при серії виграшів.

Функція приймає чотири параметри по аналогії з алгоритм Мартингейла повертає пару значень: поточний баланс після всіх ставок та історію всіх ставок у вигляді вектора структур `Bet`.

Після завершення циклу функція повертає остаточний баланс гравця і історію всіх ставок.

Таким чином, функція реалізує стратегію Д'Аламбера, яка передбачає поступове коригування розміру ставки залежно від результатів попередніх ставок, з метою мінімізувати ризики та забезпечити стабільний приріст капіталу.

```
// Стратегія Д'Аламбера
pair<int, vector<Bet>> d_alembert_strategy(int bankroll,
int initial_bet, int max_bet, int num_bets) {
    int bet = initial_bet;
    vector<Bet> bet_history;
    for (int i = 0; i < num_bets; ++i) {
        if (bankroll < bet) break;
        Bet current_bet;
        current_bet.number = i + 1; // Нумерація ставок
        current_bet.amount = bet;
        if (spin_wheel() == "black") {
            bankroll += bet;
            current_bet.result = "win";
            bet = max(1, bet - 1);
        } else {
            bankroll -= bet;
            current_bet.result = "lose";
            bet++;
            // Обмеження на максимальну ставку
            bet = min(bet, max_bet);
        }
        bet_history.push_back(current_bet);
    }
    return {bankroll, bet_history};
}
```

Рис 2.5 – Стратегія Д'Аламбера

### Функція `print_bet_history()`:

Функція `print_bet_history` призначена для виведення історії ставок гравця у вигляді таблиці. Вона приймає один параметр - константне посилання на вектор структур `Bet`, кожна з яких містить інформацію про окрему ставку (номер, суму ставки та результат).

Спочатку функція виводить заголовок "Історія ставок" і горизонтальну лінію, щоб візуально відокремити заголовок від даних. Далі виводиться заголовок таблиці з колонками "Номер", "Ставка" та "Результат".

Після цього функція проходить через кожен елемент вектора `bet\_history` за допомогою циклу `for` з ключовим словом `const auto&`, що дозволяє ітеруватися по елементах вектора без можливості їх модифікації. Для кожної ставки виводиться її номер, сума та результат у вигляді рядка таблиці. Всі значення вирівнюються відповідно до колонок таблиці для забезпечення зручності читання.

Після виведення всіх ставок функція завершується виведенням горизонтальної лінії, що відокремлює дані від наступного виводу в програмі.

```
// Функція для виведення історії ставок
void print_bet_history(const vector<Bet>& bet_history) {
    cout << "Історія ставок:" << endl;
    cout << "-----" << endl;
    cout << "| Номер | Ставка | Результат |" << endl;
    cout << "-----" << endl;
    for (const auto& bet : bet_history) {
        cout << "| " << bet.number << " | " << bet.amount << " | " << bet.result << " |" << endl;
    }
    cout << "-----" << endl;
}
```

Рис 2.6 – Функція для виведення історії ставок

## Функція main():

Спочатку генератор випадкових чисел ініціалізується за допомогою функції `srand(time(NULL))`, використовуючи поточний час як початкове значення для забезпечення різноманітності результатів. Далі програма зчитує вхідні дані від користувача: початковий баланс, початкову ставку, максимальну ставку та кількість ставок, які він хоче зробити. Ці значення зберігаються у відповідних змінних `initial_balance`, `initial_bet`, `max_bet`, та `num_bets`.

Після цього оголошується змінна `result`, яка є парою, що містить ціле число (баланс) і вектор структур `Bet` (історія ставок). Потім програма виводить меню з трьома варіантами стратегій: Мартингейл, Лабушер та Д'Аламбер. Користувач вводить свій вибір, і цей вибір зберігається у змінній `choice`.

Програма використовує оператор `switch`, щоб виконати стратегію, вибрану користувачем. Якщо користувач обирає алгоритм Мартингейл, викликається функція `martingale_strategy`, яка приймає початковий

баланс, початкову ставку, максимальну ставку і кількість ставок, і повертає результат у змінну `result`. Якщо користувач обирає стратегію Лабушера, спочатку ініціалізується початкова послідовність `initial_sequence` і послідовність `sequence`, після чого викликається функція `labouchere_strategy`, яка повертає результат у змінну `result`. Якщо користувач обирає стратегію Д'Аламбера, викликається функція `d_alembert_strategy`, яка також повертає результат у змінну `result`. Якщо вибір неправильний, програма виводить повідомлення про помилку і завершує виконання.

Далі програма викликає функцію `print_bet_history` для виведення історії ставок, використовуючи другу частину пари `result`. Потім виводиться поточний баланс, який міститься в першій частині пари `result`. Нарешті, програма завершується з кодом повернення 0, що означає успішне виконання.

```

// Зчитування вхідних даних від користувача
int initial_balance, initial_bet, max_bet, num_bets;
cout << "Введіть початковий баланс: ";
cin >> initial_balance;
cout << "Введіть початкову ставку: ";
cin >> initial_bet;
cout << "Введіть максимальну ставку: ";
cin >> max_bet;
cout << "Введіть кількість ставок: ";
cin >> num_bets;

// Оголошення результату перед використанням
pair<int, vector<Bet>> result;

// Введення стратегії гри
cout << "Виберіть стратегію гри:" << endl;
cout << "1. Алгоритм Мартингейл" << endl;
cout << "2. Стратегія Лабушера" << endl;
cout << "3. Стратегія Д'Аламбера" << endl;
int choice;
cout << "Ваш вибір: ";
cin >> choice;

// Вибір стратегії гри
switch (choice) {
    case 1:
        result = martingale_strategy(initial_balance, initial_bet, max_bet, num_bets);
        break;
    case 2: {
        vector<int> initial_sequence = {1, 2, 3, 4, 5}; // Початкова послідовність для стратегії Лабушера
        vector<int> sequence = initial_sequence;
        result = labouchere_strategy(initial_balance, sequence, max_bet, num_bets, initial_sequence);
        break;
    }
    case 3:
        result = d_alembert_strategy(initial_balance, initial_bet, max_bet, num_bets);
        break;
    default:
        cout << "Неправильний вибір стратегії!" << endl;
        return 1;
}

// Виведення результатів
print_bet_history(result.second);
cout << "Поточний баланс: " << result.first << endl;

```

Рис 2.7 – Головна функція програми

## 2.6 Тестування та налагодження програмної розробки

Для забезпечення якості та надійності програмного засобу, розробленого для моделювання гри на рулетці, було проведено комплексне тестування та налагодження. Початково, програма була піддана ретельному аналізу функціональності, зокрема алгоритмів симуляції гри та візуалізації історії ставок. Кожен з трьох основних алгоритмів гри (Мартингейл, Лабушер, Д'Аламбер) протестовано на різноманітних тестових наборах, що включали різні початкові умови банкрулу та параметри ставок.

Під час тестування було виявлено декілька ключових проблем. Наприклад, в алгоритмі Мартингейл було виявлено неправильне оброблення максимальної величини ставки, що призводило до великих втрат в разі послідовних програшів. Ця проблема була виправлена шляхом додаткової перевірки перед збільшенням ставки.

Також була виявлена недолік у стратегії Лабушера, де в некоректних випадках не відновлювалась початкова послідовність ставок після її споживання. Це призводило до неправильного розрахунку ставок та, як наслідок, до неправильного розподілу банкролу. Виправлення включало додаткові перевірки на порожнечу послідовності та коректну ініціалізацію початкових значень.

Програма була також протестована на різних операційних системах, що включали Windows, macOS та Linux. Це дозволило виявити платформозалежні проблеми, такі як різні методи генерації випадкових чисел або візуальні артефакти на певних ОС. Наприклад, на macOS виникали проблеми з рендерингом шрифтів у GUI, які було виправлено шляхом оновлення механізму рендерингу та інтеграції альтернативного шрифту, що коректно відображався на всіх платформах.

Загальний процес тестування включав запуск програми з різними комбінаціями вхідних даних для валідації коректності роботи алгоритмів гри та їх взаємодії з користувацьким інтерфейсом. Кожна функціональна можливість була ретельно протестована вручну для виявлення та виправлення всіх виявлених проблем.

В результаті всі ці заходи дозволили підвищити якість та надійність програмного забезпечення, зменшити кількість помилок та забезпечити коректну роботу програми в різних умовах експлуатації.

## 2.7 Аналіз результатів

Проведемо декілька тестів, з різними вхідними даними, та наведемо результати в таблиці:

Номер тесту	Стратегія	Кількість ігор	Ставка	Макс ставка	Початковий баланс	Кінцевий баланс	Прибуток
1	Мартингейл	25	5	100	1000	1055	55
1	Лабушер	25	5	100	1000	991	-9



1	Д'Аламбер	25	5	100	1000	993	-7
2	Мартингейл	25	5	-	1000	1055	60
2	Лабушер	25	5	-	1000	1030	30
2	Д'Аламбер	25	5	-	1000	1019	19
3	Мартингейл	100	5	100	1000	765	-235
3	Лабушер	100	5	100	1000	1133	133
3	Д'Аламбер	100	5	100	1000	1022	22
4	Мартингейл	100	5	-	1000	1285	285
4	Лабушер	100	5	-	1000	192	-808
4	Д'Аламбер	100	5	-	1000	1048	48
5	Мартингейл	1000	5	100	1000	30	-970
5	Лабушер	1000	5	100	1000	1775	775
5	Д'Аламбер	1000	5	100	1000	1178	178
6	Мартингейл	1000	5	-	1000	400	-600
6	Лабушер	1000	5	-	1000	65	-935
6	Д'Аламбер	1000	5	-	1000	1405	405
7	Мартингейл	5000	5	100	1000	22	-978
7	Лабушер	5000	5	100	1000	19	-981
7	Д'Аламбер	5000	5	100	1000	35	-965
8	Мартингейл	5000	5	-	1000	70	-930
8	Лабушер	5000	5	-	1000	12	-988
8	Д'Аламбер	5000	5	-	1000	10	-990
9	Мартингейл	1000000	5	100	1000000	562305	-437695
9	Лабушер	1000000	5	100	1000000	352403	-647597
9	Д'Аламбер	1000000	5	100	1000000	35	-999965
10	Мартингейл	1000000	5	-	1000000	861120	-138880
10	Лабушер	1000000	5	-	1000000	121684	-878316
10	Д'Аламбер	1000000	5	-	1000000	352	-999648

Аналіз результатів симуляції різних стратегій ставок на рулетці показує, як кожна з них впливає на кінцевий баланс гравця в залежності від кількості ігор, початкового балансу та максимальної ставки. Ось детальний розгляд результатів для кожного тесту:

**Тест 1:** Кількість ігор - 25, Початковий баланс – 1000

- Мартингейл: Кінцевий баланс - 1055, Прибуток - 55
- Лабушер: Кінцевий баланс - 991, Прибуток - -9
- Д'Аламбер: Кінцевий баланс - 993, Прибуток - -7

Стратегія Мартингейл тут показала невеликий позитивний результат, в той час як Лабушер і Д'Аламбер привели до незначних збитків. Це може бути пов'язано з тим, що Мартингейл компенсує програші, подвоюючи ставки, але ризик зростає з кожною наступною програною ставкою.

**Тест 2:** Кількість ігор - 25, Початковий баланс - 1000, без обмеження максимальної ставки

- Мартингейл: Кінцевий баланс - 1060, Прибуток - 60
- Лабушер: Кінцевий баланс - 1030, Прибуток - 30
- Д'Аламбер: Кінцевий баланс - 1019, Прибуток - 19

Без обмеження максимальної ставки всі стратегії показали прибуток. Це може свідчити про те, що обмеження на максимальну ставку знижує ефективність стратегій.

**Тест 3:** Кількість ігор - 100, Початковий баланс – 1000

- Мартингейл: Кінцевий баланс - 765, Прибуток - -235
- Лабушер: Кінцевий баланс - 1133, Прибуток - 133
- Д'Аламбер: Кінцевий баланс - 1022, Прибуток - 22

При збільшенні кількості ігор Мартингейл показує значний збиток, в той час як Лабушер демонструє прибуток, а Д'Аламбер залишається майже на нулі. Це може бути через високу волатильність стратегії Мартингейл на тривалих відрізках часу.

**Тест 4:** Кількість ігор - 100, Початковий баланс - 1000, без обмеження максимальної ставки

- Мартингейл: Кінцевий баланс - 1285, Прибуток - 285
- Лабушер: Кінцевий баланс - 192, Прибуток - -808
- Д'Аламбер: Кінцевий баланс - 1048, Прибуток - 48

Без обмеження максимальної ставки Мартингейл показує великий прибуток, в той час як Лабушер призводить до значних збитків. Д'Аламбер продовжує демонструвати невеликий прибуток.

**Тест 5:** Кількість ігор - 1000, Початковий баланс – 1000

- Мартингейл: Кінцевий баланс - 30, Прибуток - -970
- Лабушер: Кінцевий баланс - 1775, Прибуток - 775
- Д'Аламбер: Кінцевий баланс - 1178, Прибуток - 178

При збільшенні кількості ігор до 1000 стратегія Мартингейл призводить до майже повної втрати банкрулу. Лабушер і Д'Аламбер, навпаки, показують значний прибуток. Це підтверджує високий ризик стратегії Мартингейл на довгих дистанціях.

**Тест 6:** Кількість ігор - 1000, Початковий баланс - 1000, без обмеження максимальної ставки

- Мартингейл: Кінцевий баланс - 400, Прибуток - -600
- Лабушер: Кінцевий баланс - 65, Прибуток - -935
- Д'Аламбер: Кінцевий баланс - 1405, Прибуток - 405

Навіть без обмеження максимальної ставки, Мартингейл показує збиток, як і Лабушер. Д'Аламбер демонструє значний прибуток, що свідчить про його стабільність на довгих дистанціях.

**Тест 7:** Кількість ігор - 5000, Початковий баланс – 1000

- Мартингейл: Кінцевий баланс - 22, Прибуток - -978
- Лабушер: Кінцевий баланс - 19, Прибуток - -981
- Д'Аламбер: Кінцевий баланс - 35, Прибуток - -965

У довготривалій перспективі (5000 ігор) всі стратегії демонструють значні збитки, що підкреслює високий ризик та нестабільність таких стратегій при тривалому використанні.

**Тест 8:** Кількість ігор - 5000, Початковий баланс - 1000, без обмеження максимальної ставки

- Мартингейл: Кінцевий баланс - 70, Прибуток - -930
- Лабушер: Кінцевий баланс - 12, Прибуток - -988
- Д'Аламбер: Кінцевий баланс - 10, Прибуток - -990

Навіть без обмеження максимальної ставки, всі стратегії демонструють значні збитки після 5000 ігор. Це підкреслює, що без обмежень збитки можуть бути дуже великими.

**Тест 9:** Кількість ігор - 1000000, Початковий баланс – 1000000

- Мартингейл: Кінцевий баланс - 562305, Прибуток - -437695
- Лабушер: Кінцевий баланс - 352403, Прибуток - -647597
- Д'Аламбер: Кінцевий баланс - 35, Прибуток - -999965

Для дуже великої кількості ігор (1000000) всі стратегії показують значні збитки. Мартингейл і Лабушер демонструють збитки в сотні тисяч, а Д'Аламбер майже повністю втрачає баланс.

**Тест 10:** Кількість ігор - 1000000, Початковий баланс - 1000000, без обмеження максимальної ставки

- Мартингейл: Кінцевий баланс - 861120, Прибуток - -138880
- Лабушер: Кінцевий баланс - 121684, Прибуток - -878316
- Д'Аламбер: Кінцевий баланс - 352, Прибуток - -999648

Навіть без обмеження максимальної ставки, усі стратегії показують значні збитки на дуже довгих дистанціях. Мартингейл менш збитковий, але все одно неефективний. Лабушер і Д'Аламбер майже повністю втрачають баланс.

На основі проведеного аналізу результатів симуляції різних стратегій ставок на рулетці можна зробити кілька важливих висновків. В першу чергу, стратегія Мартингейла показала значні ризики, особливо при тривалих серіях ігор. Незважаючи на те, що в короткострокових симуляціях (25 та 100 ігор) вона показала певний прибуток, у довгострокових симуляціях (1000, 5000 та 1000000 ігор) стратегія привела до суттєвих втрат. Це підтверджує теорію про високу волатильність та небезпеку використання Мартингейла в реальних умовах, де існують обмеження на максимальну ставку.

Стратегія Лабушера виявилася більш стабільною порівняно з Мартингейлом у середньострокових симуляціях (100 та 1000 ігор), де вона показала прибуток. Однак, у довгострокових симуляціях стратегія Лабушера також зазнала значних втрат, що свідчить про її нестабільність при великій кількості ігор. Ця стратегія демонструє певну ефективність в короткостроковій

перспективі, але також потребує обережного підходу через можливі великі втрати в довготривалих іграх.

Стратегія Д'Аламбера показала найбільшу стабільність серед усіх трьох стратегій. У всіх симуляціях, навіть при великій кількості ігор, вона продемонструвала відносно невеликі втрати або навіть прибуток. Це вказує на те, що стратегія Д'Аламбера є менш ризикованою і може бути більш прийнятною для гравців, які прагнуть мінімізувати втрати.

## **2.8 Рекомендації по впровадженню та використанню програмної розробки**

Впровадження та використання програмної розробки для аналізу та моделювання стратегій ставок на рулетці потребує врахування кількох важливих аспектів, які допоможуть максимально ефективно інтегрувати цю програму в дослідницький чи навчальний процес. Перед початком використання програми необхідно встановити всі необхідні компоненти та забезпечити правильне налаштування середовища розробки. Це включає встановлення всіх залежностей та бібліотек, необхідних для коректного функціонування програми.

Перш за все, необхідно завантажити всі необхідні бібліотеки з офіційних джерел та інтегрувати їх у проект, налаштувавши компілятор для правильної лінковки. Зазвичай, це включає вказівку на шляхи до заголовочних файлів та бібліотек, а також правильне налаштування параметрів компіляції. Середовище розробки має бути налаштоване таким чином, щоб забезпечити безперебійне виконання програми та уникнути конфліктів між різними версіями бібліотек.

Після завершення налаштування середовища, потрібно конфігурувати початкові параметри програми. Це включає визначення початкового балансу, вибір стратегії ставок за замовчуванням та встановлення початкової кількості ігор для симуляції. Вихідний код надає достатньо гнучкості для модифікацій, тому можна легко адаптувати ці параметри під свої потреби, змінюючи відповідні значення у вихідному коді або додаючи нові параметри через інтерфейс.

Для запуску програми необхідно переконатися, що всі необхідні файли (конфігураційні файли, дані для симуляцій) знаходяться у вказаних директоріях. У разі виникнення проблем необхідно перевірити налаштування компіляції та переконатися, що всі необхідні бібліотеки правильно підключені. Інтерфейс програми надає можливість легко взаємодіяти з нею. Основні кнопки для вибору стратегії ставок та управління параметрами симуляції розташовані у

верхній частині вікна. Натискання на кнопки змінює стан програми, дозволяючи миттєво змінювати поточну стратегію або регулювати кількість ігор для симуляції.

Користувач може зупиняти або відновлювати процес симуляції, що дозволяє детально аналізувати кожен етап виконання стратегії. Такий підхід особливо корисний у навчальних середовищах, де важливо продемонструвати різні кроки та порівняти їх поведінку. Програма особливо корисна для навчальних цілей, оскільки дозволяє здобувачам освіти та викладачам спостерігати за роботою різних стратегій ставок у реальному часі. Рекомендується використовувати цю програму як інструмент візуалізації під час лекцій або лабораторних робіт з теорії ймовірностей та статистики.

Здобувачі освіти можуть змінювати параметри програми та аналізувати, як ці зміни впливають на результати симуляцій. Викладачі можуть використовувати програму для демонстрації теоретичних аспектів стратегій ставок, показуючи, як саме ці стратегії працюють на практиці. Це дозволяє краще зрозуміти механізми роботи стратегій, їхні сильні та слабкі сторони, а також потенційні ризики та вигоди.

Для успішного впровадження та використання даної програмної розробки важливо забезпечити ретельну підготовку середовища, активне використання можливостей інтерфейсу, а також забезпечити підтримку та можливості для подальшого розширення функціональності. Програма повинна бути налаштована таким чином, щоб забезпечити максимальну гнучкість та зручність для користувачів. Це включає можливість легко змінювати параметри симуляцій, додавати нові стратегії ставок та аналізувати результати.

Для цього важливо регулярно оновлювати програму, додаючи нові функції та виправляючи можливі помилки. Також корисно підтримувати активну комунікацію з користувачами, враховуючи їхні побажання та пропозиції щодо покращення програми. Це допоможе забезпечити високу якість та ефективність програмної розробки, а також зробить її корисною та зручною для широкого кола користувачів.

Особливу увагу слід приділити документації. Вона має бути детальною та зрозумілою, щоб користувачі могли легко ознайомитися з можливостями програми та правильно налаштувати її для своїх потреб. Документація повинна включати докладні інструкції з встановлення, налаштування та використання програми, а також приклади симуляцій та аналізу результатів.

Загалом, впровадження та використання програмної розробки для аналізу та моделювання стратегій ставок на рулетці може значно покращити процес дослідження та навчання у цій сфері. Важливо забезпечити належну підготовку та підтримку програми, щоб вона могла ефективно виконувати свої функції та приносити користь користувачам.



## **ВИСНОВКИ**

У даній дипломній роботі було проаналізовано та оцінено ефективність різних алгоритмів стратегій виграшу в онлайн-казино. Основною метою дослідження було створення програмного продукту для симуляції гри, що дозволило вивчити переваги та недоліки кожної стратегії. Проведене тестування підтвердило, що, хоча деякі стратегії можуть давати позитивні результати у короткостроковій перспективі, у довгостроковому періоді перевага завжди залишається на боці казино. Розроблене програмне забезпечення виявилось корисним інструментом для оцінки ризиків та можливостей різних стратегій, що може бути корисним для гравців у прийнятті обґрунтованих рішень щодо участі в азартних іграх. Виконана робота демонструє важливість обережного підходу до використання стратегій ставок та підкреслює необхідність усвідомлення ризиків, пов'язаних з азартними іграми.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ставки на рулетці: стратегії та їх ефективність. Wikipedia. URL: [https://uk.wikipedia.org/wiki/Стратегії\\_ставок\\_на\\_рулетці](https://uk.wikipedia.org/wiki/Стратегії_ставок_на_рулетці)
2. Аналіз стратегій ставок на рулетці. ResearchGate. URL: [https://www.researchgate.net/publication/335636999\\_Analysis\\_of\\_Betting\\_Strategies\\_in\\_Roulette](https://www.researchgate.net/publication/335636999_Analysis_of_Betting_Strategies_in_Roulette)
3. Мартингейл, Лабушер і Д'Аламбер: порівняльний аналіз. IEEE Xplore. URL: <https://ieeexplore.ieee.org/document/8893923>
4. Волатильність стратегій ставок у азартних іграх. Journal of Gambling Studies. URL: <https://link.springer.com/article/10.1007/s10899-019-09865-5>
5. Теорія і практика стратегій ставок. ScienceDirect. URL: <https://www.sciencedirect.com/science/article/pii/S0898122120300046>
6. Стратегії ставок: ефективність і ризику. SSRN. URL: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3400216](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3400216)
7. Математика в азартних іграх: аналіз стратегій ставок. SpringerLink. URL: <https://link.springer.com/book/10.1007/978-3-030-45525-5>
8. Системи ставок у рулетці: теорія і практика. Google Books. URL: <https://books.google.com.ua/books?id=R7w1DwAAQBAJ&printsec=frontcover>
9. Ризику та винагороди стратегій ставок на рулетці. Cambridge University Press. URL: <https://www.cambridge.org/core/journals/journal-of-financial-and-quantitative-analysis/article/abs/risk-and-reward-of-betting-strategies/AB89D3C4FA6DB657BD8F4E1DFFA6>
10. Економічний аналіз стратегій ставок. Taylor & Francis Online. URL: <https://www.tandfonline.com/doi/abs/10.1080/1331677X.2021.1874657>
11. Ставки на рулетці: моделювання та симуляція. Journal of Economic Dynamics and Control. URL: <https://www.sciencedirect.com/science/article/pii/S0165188917302168>
12. Мартингейл і його альтернативи у ставках. JSTOR. URL: <https://www.jstor.org/stable/2335517>
13. Лабушер та інші прогресивні стратегії ставок. Springer. URL: [https://link.springer.com/chapter/10.1007/978-3-030-01319-6\\_6](https://link.springer.com/chapter/10.1007/978-3-030-01319-6_6)
14. Вплив стратегій ставок на фінансові результати. Elsevier. URL: <https://www.elsevier.com/books/gambling-theory-and-other-topics/rosenthal/978-0-12-597670-2>
15. Аналіз ставок на рулетці за допомогою симуляцій. Wiley Online Library. URL: <https://onlinelibrary.wiley.com/doi/10.1002/wics.1329>
16. Стратегії ставок у казино: математичний підхід. Nature. URL: <https://www.nature.com/articles/s41586-019-1532-5>
17. Ставки на рулетці: емпіричний аналіз. MDPI. URL: <https://www.mdpi.com/2227-7390/8/9/1601>
18. Моделювання стратегій ставок на рулетці. ArXiv. URL: <https://arxiv.org/abs/2006.06519>

19. Стратегії ставок: аналіз ефективності. ACM Digital Library. URL: <https://dl.acm.org/doi/10.1145/3376464>
20. Вплив математичних стратегій на ставки у рулетці. SAGE Journals. URL: <https://journals.sagepub.com/doi/10.1177/2158244020923358>
21. Стратегії ставок у азартних іграх: огляд літератури. De Gruyter. URL: <https://www.degruyter.com/document/doi/10.1515/jtse-2020-0134/html>
22. Мартингейл і його вплив на ставки. PLOS ONE. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0237919>
23. Лабушер: теорія і практика. Frontiers in Psychology. URL: <https://www.frontiersin.org/articles/10.3389/fpsyg.2020.02063/full>
24. Д'Аламбер і інші негативні прогресії у ставках. Springer Nature. URL: <https://link.springer.com/article/10.1007/s12562-021-01541-3>
25. Азартні ігри і стратегії ставок. World Scientific. URL: <https://www.worldscientific.com/worldscibooks/10.1142/11972>

## ДОДАТКИ

### ПІДСТАВИ ДЛЯ РОЗРОБКИ

Розробка даної програми здійснюється у рамках навчального проекту для дослідження ефективності різних стратегій ставок в рулетці. Потреба у розробці даного програмного забезпечення обумовлена необхідністю аналізу результатів застосування стратегій для покращення розуміння їхньої ефективності та ризиків.

### ПРИЗНАЧЕННЯ РОЗРОБКИ

Програма призначена для симуляції та аналізу різних стратегій ставок в рулетці, таких як алгоритм Мартингейл, стратегія Лабушера та стратегія Д'Аламбера. Вона дозволить користувачам:

- Ознайомитися з принципами роботи різних стратегій ставок.
- Аналізувати результати використання різних стратегій.
- Порівнювати ефективність стратегій на різних наборах вхідних даних.

### ВИМОГИ ДО ПРОГРАМНОГО ЗАСОБУ

Вимоги до функціональних характеристик

#### 1. Симуляція стратегій ставок:

Програма повинна підтримувати симуляцію наступних стратегій ставок:

- Алгоритм Мартингейл.
- Стратегія Лабушера.
- Стратегія Д'Аламбера.

Кожна стратегія повинна мати можливість налаштування початкових параметрів, таких як початковий баланс, початкова ставка, максимальна ставка та кількість ставок.

#### 2. Виведення результатів:

Програма повинна виводити історію ставок для кожної стратегії, включаючи номер ставки, суму ставки та результат (виграш або програш).

Після завершення симуляції повинна відображатися кінцева сума на рахунку користувача.

### 3. Інтерфейс користувача:

Програма повинна мати текстовий інтерфейс, що дозволяє користувачу вводити початкові дані та обирати стратегію.

Виведення результатів повинно бути зрозумілим та зручним для аналізу.

## Нефункціональні вимоги

### 1. Продуктивність:

Програма повинна працювати без затримок та забезпечувати швидку обробку великих кількостей даних.

### 2. Надійність:

Програма повинна коректно працювати при будь-яких допустимих вхідних даних та не допускати збоїв.

## Вимоги до програмної сумісності

- Розробка повинна бути сумісною з операційними системами Windows, Linux та MacOS.
- Програма повинна бути написана на мові програмування C++ з використанням стандартної бібліотеки.

## СТАДІЇ І ЕТАПИ РОЗРОБКИ

1. Вивчення стратегій ставок в рулетці та їх реалізація в C++.
2. Розробка архітектури програми.
3. Написання коду для кожної стратегії ставок.
4. Створення текстового інтерфейсу користувача.
5. Тестування на різних наборах вхідних даних.
6. Виправлення виявлених помилок та оптимізація коду.

## ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ

Після розробки було перевірено:

- Коректність реалізації стратегій ставок.
- Правильність виведення історії ставок.
- Сумісність програми з різними операційними системами.
- Зручність інтерфейсу та легкість використання програми.

## АНОТАЦІЯ

### **Падука Д.А. Аналіз алгоритмів стратегій виграшу в онлайн-казино**

Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 122 Комп'ютерні науки. Волинський національний університет імені Лесі Українки, Луцьк, 2024 р.

Бакалаврська робота присвячена розробці комп'ютерного продукту для симуляції гри, проведенню серії симуляцій для тестування алгоритмів. Метою даної дипломної роботи є аналіз алгоритмів стратегій виграшу в онлайн-казино та розробка програмного продукту для оцінки їх ефективності та можливостей застосування в реальних умовах.

У ході роботи було проаналізовано та оцінено ефективність алгоритмів: Мартингейл, Лабушер, Д'Аламбер. Кожен алгоритм було розроблено і реалізовано програмно, що дозволило дослідити та проаналізувати його в реальному часі. Програма реалізована за допомогою Visual Studio Code який підтримує багато мов програмування, включаючи такі популярні, як C++, Python, JavaScript, Java та інші, що робить його. Проект базується на ітеративному підході, що дозволяє поетапно розширювати функціонал та вдосконалювати програму, а також використовує модель інкрементальної розробки для ефективного тестування і відлагодження. Гнучкий підхід до розробки дозволяє програмі швидко адаптуватися до змін у вимогах або виявлених можливостях, забезпечуючи при цьому управління потенційними ризиками для стабільності і надійності програми. Для забезпечення якості та надійності програмного засобу, розробленого для моделювання гри на рулетці, було проведено комплексне тестування та налагодження. Початково, програма була піддана ретельному аналізу функціональності, зокрема алгоритмів симуляції гри та візуалізації історії ставок. Кожен з трьох основних алгоритмів гри (Мартингейл, Лабушер, Д'Аламбер) протестовано на різноманітних тестових наборах, що включали різні початкові умови банкрулу та параметри ставок.

Здобувачі освіти можуть змінювати параметри програми та аналізувати, як ці зміни впливають на результати симуляцій. Викладачі можуть використовувати програму для демонстрації теоретичних аспектів стратегій ставок, показуючи, як саме ці стратегії працюють на практиці. Це дозволяє краще зрозуміти механізми роботи стратегій, їхні сильні та слабкі сторони, а також потенційні ризики та вигоди.

Ключові слова: алгоритми стратегій виграшу, Мартингейл, Лабушер, Д'Аламбер, банкрулл, ітеративний підхід, рулетка