

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)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ ОЦІНЮВАННЯ ІНВЕСТИЦІЙНИХ ПОРТФЕЛІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
TOOLS FOR EVALUATING INVESTMENT PORTFOLIOS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-113м
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ОЛЬХОВСЬКИЙ А.І.

(ПРІЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРІЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
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.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ОЛЬХОВСЬКОГО Артема Ігоровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів оцінювання інвестиційних портфелів. Research and Software Implementation of Tools for Evaluating Investment Portfolios

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “07” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 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. Опис програми. 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	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “9” вересня 2024 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.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Артем ОЛЬХОВСЬКИЙ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Михайло КОЦУР
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
85 с., 3 табл., 27 рис., 3 дод., 22 джерела.

PYCHARM, PYTHON, ІНВЕСТИЦІЯ, ІНВЕСТИЦІЙНИЙ ПОРТФЕЛЬ,
МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,
СЕРЕДОВИЩЕ РОЗРОБКИ.

Об'єкт дослідження – процес розробки програмного забезпечення для оцінювання інвестицій.

Предмет дослідження – програмні засоби для оцінювання інвестиційних портфелів.

Мета роботи – дослідження та розробка програмного забезпечення для оцінювання інвестиційних портфелів.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm.

Результати. Розроблено програмне забезпечення для оцінювання інвестиційних портфелів за допомогою мови програмування Python та середовища розробки PyCharm.

Практична цінність роботи полягає у розробці програмного забезпечення, що забезпечує оцінювання інвестиційних портфелів.

Висновки. Виконано проєктування програмного забезпечення для оцінювання інвестиційних портфелів. Розроблено програмне забезпечення для оцінювання інвестиційних портфелів. Здійснено тестування розробленого програмного забезпечення для оцінювання інвестиційних портфелів.

Галузь використання – фінансовий аналіз.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 85 pages, 3 tables, 27 figures, 3 appendixes, 22 sources.

PYCHARM, PYTHON, INVESTMENT, INVESTMENT PORTFOLIO, PROGRAMMING LANGUAGE, SOFTWARE, DEVELOPMENT ENVIRONMENT.

The object of research is the process of developing software for investment evaluation.

The subject of the research is software for evaluating investment portfolios.

The purpose of this work is to research and develop the software for evaluating investment portfolios.

Materials, methods and technical tools: Python programming language, PyCharm development environment.

Results. The software tools for evaluating investment portfolios using the Python programming language and the PyCharm development environment have been developed.

The practical value of the work lies in the development of software that provides evaluation of investment portfolios.

Conclusions. The software for evaluating investment portfolios has been designed. The software for evaluating investment portfolios has been developed. The developed software for evaluating investment portfolios has been tested.

Scope of use – financial analysis.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Програмне забезпечення для оцінювання інвестиційних портфелів	11
1.2 Аналіз програмного забезпечення для оцінювання інвестиційних портфелів	21
1.3 Висновки за розділом 1	32
2 Матеріали і методи	34
2.1 Вибір мови програмування	34
2.2 Вибір середовища розробки для створення програмного забезпечення для оцінювання інвестиційних портфелів.....	36
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів.....	38
2.4 Висновки за розділом 2	40
3 Опис програми	41
3.1 Структура програмного забезпечення для оцінювання інвестиційних портфелів	41
3.2 Функціонування програмного забезпечення для оцінювання інвестиційних портфелів.....	42
3.3 Використання нейронних мереж для оцінювання інвестиційних портфелів	44
3.4 Проєктування інтерфейсу програмного забезпечення для оцінювання інвестиційних портфелів.....	46
3.5 Висновки за розділом 3	48
4 Експлуатація, тестування та експериментальне дослідження програми.....	50
4.1 Призначення й умови застосування програми	50
4.2 Характеристики програми для оцінювання інвестиційних портфелів ...	51

4.3 Інструкція по експлуатації програми.....	52
4.3.1 Звернення до програми.....	52
4.3.2 Вхідні й вихідні дані	52
4.3.3 Повідомлення.....	52
4.4 Виконання програмного забезпечення для оцінювання інвестиційних портфелів	52
4.5 Тестування програмного забезпечення для оцінювання інвестиційних портфелів	57
4.6 Висновки за розділом 4	57
Висновки	58
Перелік джерел посилання	61
Додаток А Технічне завдання	63
Додаток Б Фрагмент тексту програми	67
Додаток В Слайди презентації	79

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

VaR – Value at Risk;

БД – база даних;

НДКП – нейромережі з довготривалою та короткою пам'яттю;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер.

ВСТУП

Збільшення кількості доступної інформації з ринків капіталу, фінансових звітів та аналітики робить необхідним застосування сучасних програмних засобів для швидкої й точної обробки даних. Інвестиційні портфелі потребують регулярної оцінки, що пов'язано з використанням програмних засобів для аналізу ринкових трендів, управління ризиками та прогнозування доходів [1]-[6].

Програмні рішення дозволяють автоматизувати рутинні операції, такі як балансування портфелів, аналіз волатильності та ефективності активів. Це зменшує ручну працю і знижує ризик людських помилок [1]-[6].

Програмне забезпечення для оцінювання інвестиційних портфелів часто використовує алгоритми машинного навчання, штучний інтелект та аналітику великих даних для покращення точності рішень, що дає змогу інвесторам оцінювати й коригувати стратегії в реальному часі [1]-[6].

Багато країн посилюють регулювання фінансових ринків. Інвестори та фінансові установи повинні відповідати вимогам щодо звітності, управління ризиками та розкриття інформації. Спеціалізоване програмне забезпечення спрощує цей процес і забезпечує відповідність вимогам [1]-[6].

Проте багато програмних рішень для фінансового аналізу, особливо для великих фінансових установ, можуть бути досить дорогими в розробці і впровадженні, а також в обслуговуванні та оновленні. Крім того, програмне забезпечення для оцінювання інвестиційних портфелів залежить від якості вхідних даних. Якщо при синтезі розпізнавальних моделей, що використовуються у відповідному програмному забезпеченні, було використано неточні або неповні дані, результати аналізу можуть бути спотворені або ненадійні. Програмне забезпечення для оцінювання інвестиційних портфелів часто використовує складні алгоритми для прогнозування та оцінки. Неправильне налаштування алгоритмів або використання недосконалих та неефективних алгоритмів можуть призвести до

помилкових висновків і ризиків. Ці недоліки обумовлюють доцільність розробки програмного забезпечення для оцінювання інвестиційних портфелів [1]-[6].

Отже, актуальною є розробка програмного забезпечення для оцінювання інвестиційних портфелів. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання дослідження та розробки програмного забезпечення для оцінювання інвестиційних портфелів.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для оцінювання інвестиційних портфелів;
- розробити модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів;
- здійснити проєктування програмного забезпечення для оцінювання інвестиційних портфелів;
- створити програмне забезпечення для оцінювання інвестиційних портфелів;
- виконати тестування розробленого програмного забезпечення для оцінювання інвестиційних портфелів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для оцінювання інвестиційних портфелів

Інвестиційний портфель – це набір фінансових активів, таких як акції, облігації, нерухомість, інвестиційні фонди та інші інструменти, які інвестор використовує для досягнення своїх фінансових цілей. Основною метою створення інвестиційного портфеля є оптимізація доходу при мінімізації ризиків шляхом диверсифікації, розподілу активів серед різних класів, секторів або географічних регіонів [1]-[6].

Портфель формується з урахуванням таких факторів, як рівень ризику, який інвестор готовий прийняти, тривалість інвестицій, очікуваний дохід та ринкові умови. Ефективне управління портфелем передбачає використання спеціалізованого програмного забезпечення, яке допомагає підтримувати постійний аналіз та коригування його складу відповідно до змін на ринку та фінансових цілей [1]-[6].

Аналіз портфеля передбачає оцінку інвестиційного портфеля з метою перевірки його відповідності фінансовим цілям та уподобанням. Він також дозволяє визначити, чи досягаються встановлені для портфеля показники, зокрема стосовно ризику, інфляції та продуктивності [1].

Відомо, що ринкові події можуть спричинити різну реакцію інвесторів, і деякі активи можуть виявитися більш чутливими до змін, ніж інші. Фінансові оператори ретельно відстежують і вимірюють цю чутливість коштів до ринкових коливань [1].

Аналіз портфеля за допомогою програмних засобів включає не лише вибір відповідних акцій, але й вибір тих, які найкраще підходять для конкретного портфеля [1]. Втім, така інвестиційна стратегія не гарантує стійкості. Під час фінансової кризи багато фінансових інструментів втратили свою привабливість, а ціни на акції, облігації, взаємні фонди, золото, товари, деривативи та нерухомість значно змінилися [1].

Оптимізація структури активів портфеля може покращити результати та знизити ризики [1].

Існують програмні інструменти, що спрощують процес аналізу портфеля для інвесторів. Щоб виконати початковий аналіз, можна скористатися певними кроками або звернутися до консультантів за професійними порадами [1].

Одним із корисних методів є використання програмного забезпечення у вигляді аналізаторів портфелів акцій. Завдяки таким інструментам можна відстежувати ефективність інвестицій та аналізувати їх склад [1]. Ці платформи надають інформацію про структуру активів портфеля, до яких можуть входити акції, облігації, готівкові кошти, а також американські чи іноземні цінні папери. Сектори, такі як енергетика або високі технології, також враховуються при аналізі [1].

Диверсифікація портфеля за допомогою програмних засобів дозволяє знизити ризики, оскільки зменшення частки активів, що рухаються однаково, допомагає запобігти суттєвим втратам [1].

Оцінка загальної ефективності портфеля за допомогою спеціалізованих програмних засобів виконується таким чином, що на початку оцінюється портфель у цілому, порівнявши його з іншими показниками. Наприклад, можна провести порівняння з відповідними індексами, а також з більш загальними ринковими показниками. Якщо інвестиції зосереджені на технологічних акціях, варто звернути увагу на порівняння з відповідними показниками [1].

Важливо також врахувати такі фактори, як співвідношення ціни та прибутку, дивідендна прибутковість, а також очікуваний темп зростання [1].

На етапі оцінки ефективності кожного елемента портфеля за допомогою програмних засобів необхідно проаналізувати розподіл активів, зокрема акцій, з урахуванням індивідуальних фінансових обставин [1]. Варто звернути увагу на роль облігацій у портфелі, облігації зазвичай забезпечують стабільність і доходи. Слід враховувати, що значна частина доходу від

облігацій походить від купонних платежів. Потрібно оцінити, чи є в портфелі облігації, що торгуються з премією, але можуть бути викликані відповідно до стратегії. Продаж облігацій з премією і реінвестування коштів за нижчою ставкою може підвищити дохідність [1].

Додавання інших інвестицій також може покращити загальну дохідність портфеля. Доцільно вимірювати їх відносну ефективність відповідно до встановлених критеріїв, а також переглядати загальну продуктивність [1].

Коли йдеться про активне управління портфелями за допомогою програмних засобів, мається на увазі проведення аналізу та балансування портфеля [1].

Використання високоякісних цінних паперів підвищує ймовірність успіху. Важливо також враховувати взаємодію кожного цінного паперу з іншими активами в портфелі. Навіть найкращі окремі елементи можуть не поєднуватися між собою [1]. Аналіз портфеля за допомогою програмних засобів є ефективним інструментом для підвищення прибутковості та зменшення ризиків [1]. Минулі показники не гарантують майбутніх результатів, але є найкращим орієнтиром для ухвалення рішень в умовах ринкової невизначеності [1].

Після того як за допомогою програмних засобів проведено аналіз всього портфеля та кожного його компонента, необхідно визначити, чи виконуються поставлені цілі [1]. Спочатку варто подумати про власні пріоритети та поточну життєву ситуацію. Також слід врахувати ставлення до ризику. Потрібно запитати себе, яких економічних змін побоюється найбільше. Наприклад, при інфляції, можливо, інвестиції в золото та облігації принесуть більше переваг, ніж у акції [1].

Фінансові експерти, обговорюючи активне управління портфелями, мають на увазі процес аналізу та періодичного перегляду портфеля. Інвестиційний портфель можна розглядати як ключовий компонент стратегії. Використання найякісніших цінних паперів підвищує ймовірність успіху, так

само як якісні матеріали впливають на результат. Однак важливо враховувати, як кожен актив взаємодіє з іншими елементами портфеля. Зрештою, навіть найкращі продукти можуть не поєднуватися між собою [1].

Аналіз портфеля за допомогою програмних засобів є ефективним інструментом для тих, хто прагне максимізувати прибутки та мінімізувати ризики. У кожному ринку є свій рівень невизначеності, але це не просто гра випадковості. Хоча минулі результати не гарантують майбутніх, вони залишаються важливим орієнтиром для прийняття рішень на невизначених ринках [1].

Інвестиційний аналіз за допомогою програмних засобів охоплює різноманітні методи оцінки інвестицій, галузей та економічних тенденцій. Він може використовуватись для обчислення минулих прибутків з метою прогнозування майбутніх результатів, вибору найбільш підходящих інвестиційних варіантів для інвестора або оцінки ризику, потенційної прибутковості чи волатильності окремих активів, таких як акції та облігації [2].

Головною метою інвестиційного аналізу за допомогою програмних засобів є визначення доцільності певної інвестиції та її відповідності потребам конкретного інвестора. Важливими аспектами є правильний момент для входу, очікуваний термін інвестування та передбачувана позиція в рамках загального портфеля [2].

При аналізі взаємних фондів за допомогою програмних засобів оцінюється, як фонд показав себе за певний період, порівнюючи результати з відповідним індексом або конкурентами. Аналіз корпоративних фондів включає оцінку продуктивності, обмінних курсів, якості управління, навичок, інвестиційних рішень та ефективного розподілу ресурсів [2].

Не існує єдиного підходу до інвестування, оскільки кожна стратегія має свої цілі, можливості та результати. Тому для кожної стратегії можна знайти відповідні інвестиційні можливості [2].

Інвестиційний аналіз за допомогою програмних засобів може також включати більш глибоку оцінку потреб інвестора, фінансового стану та роботи фонду, а також визначення необхідності коригування стратегії. Інвестори можуть самостійно проводити аналіз або звертатися до інвестиційних консультантів чи інших фінансових експертів за порадою [2].

Незважаючи на велику кількість способів оцінки акцій, секторів і ринків, фінансовий аналіз зазвичай поділяється на кілька основних методів. Основний інвестиційний аналіз передбачає оцінку кожної окремої акції на основі її специфічних характеристик, таких як вартість, управлінський потенціал, фінансова стійкість та інші важливі параметри [2].

При цьому аналіз не орієнтується на економічні цикли чи ринкові тенденції. Він зосереджений на конкретних компаніях і акціях, які мають перспективи зростання навіть в умовах несприятливого ринкового середовища. Таким чином, підхід знизу вгору ґрунтується на мікроекономічному аналізі, а не на макроекономічному чи глобальному погляді [2].

Глобальний підхід до фінансового аналізу за допомогою програмних засобів починається з вивчення загальних економічних, ринкових і галузевих тенденцій, після чого здійснюється пошук фінансових можливостей, що базуються на цих спостереженнях. Наприклад, інвестор може оцінити перспективи різних секторів і вирішити, що фінансова галузь має найкращі перспективи. Тоді він зосереджує зусилля на виборі найперспективніших акцій у цьому секторі [2]. З іншого боку, інвестор може знайти компанію у промисловому секторі, яка, на його думку, має хороші перспективи, і вкласти кошти в неї, навіть якщо загальна ситуація в галузі не виглядає привабливо. Інвестор очікує, що акції цієї компанії покажуть кращі результати, ніж інші компанії у цьому секторі [2]. До інших типів інвестиційного аналізу належать фундаментальний і технічний аналіз [2].

Основний аналіз за допомогою програмних засобів зосереджується на вивченні фінансового стану компанії, а також загального економічного стану.

Провідні аналітики, які шукають акції, вважають, що ринок недооцінений, тому вважають, що ці акції мають внутрішню вартість [2].

Часто застосовуючи аналіз фондів, відповідні програмні засоби оцінюють фінансову стабільність компаній, їхні перспективи зростання та здатність виплачувати дивіденди, щоб визначити, куди варто вкласти кошти [2].

Технічний аналіз базується на оцінці цінових моделей та тенденцій за допомогою графіків і діаграм. На відміну від фундаментального аналізу, який оцінює внутрішню вартість активів, технічні аналітики зосереджуються на аналізі руху цін, торгових сигналів і різних статистичних показників, що допомагають визначити сильні та слабкі сторони акцій [2].

Аналітичні звіти про окремі акції, класи активів та ринки часто публікуються з рекомендаціями щодо купівлі, продажу або утримання. Кожна служба використовує власні показники для оцінки ситуації на основі проведених досліджень [2].

Першим етапом інвестиційного аналізу за допомогою програмних засобів є оцінка доступних інвестиційних можливостей. Після цього необхідно визначити, чи зможе ця інвестиція забезпечити більший прибуток порівняно з іншими доступними варіантами. Завершальним кроком є оцінка того, чи виправдовує ризик потенційний дохід від цієї інвестиції [2].

Основними методами інвестиційного аналізу є фундаментальний та технічний аналіз. Фундаментальний аналіз полягає у вивченні фінансових показників компанії, таких як дохід, прибуток, грошові потоки та операційні витрати, а також враховує макроекономічні фактори. Технічний аналіз фокусується на вивченні цінових моделей акцій або інших активів за допомогою графіків і діаграм для прогнозування майбутніх змін на основі минулих даних [2].

Програмні засоби інвестиційної аналітики підтримують економічний та фінансовий аналіз для прийняття інвестиційних рішень. Програмні засоби інвестиційної аналітики досліджують економічні звіти та державні дані,

зокрема, рівень безробіття та інфляцію. Зібрані дані аналізуються, і на основі цього робляться висновки щодо доцільності купівлі або продажу активів, таких як акції, облігації чи нерухомість [2].

Перед прийняттям фінансових рішень необхідно провести ретельний аналіз. Важливо врахувати загальний стан економіки, специфіку галузі та глобальні політичні фактори, щоб зрозуміти, де можна отримати прибуток та уникнути ризиків. Мета інвестицій полягає в знаходженні оптимального балансу між ризиком і винагородою [2].

Оцінка ефективності інвестиційних портфелів за допомогою відповідних програмних засобів є важливою частиною фінансового процесу, що допомагає інвесторам і менеджерам аналізувати вартість своїх інвестиційних стратегій [1]-[6].

Основне завдання оцінки ефективності інвестиційних портфелів полягає у визначенні того, чи досягне обрана інвестиційна стратегія бажаних результатів щодо рівня ризику та очікуваного прибутку [1]-[6].

Одним із ключових аспектів оцінки ефективності за допомогою відповідних програмних засобів є аналіз рівня ризику, який бере на себе інвестор. Це може здійснюватися шляхом вимірювання динаміки вартості портфеля через стандартне відхилення або інші показники ризику, а також шляхом аналізу прибутковості та ринкової вартості інвестиційного портфеля [1]-[6].

Аналіз фінансових результатів за допомогою відповідних програмних засобів передбачає оцінку прибутковості портфеля за певний період. Можуть розглядатися такі показники, як абсолютний дохід, відносна прибутковість у порівнянні з еталоном, або ж прибутковість, скоригована на ризик, що враховує обсяг ризику для досягнення конкретного доходу [1]-[6].

Методи оцінки ефективності з поправкою на ризик дають змогу оцінити співвідношення очікуваного прибутку та пов'язаного з ним ризику. Це важливо для визначення, наскільки ефективним є портфель з точки зору співвідношення ризику та винагороди [1]-[6].

Аналіз атрибутів спрямований на виявлення факторів, які впливають на результати портфеля, таких як розподіл активів, вибір цінних паперів та кореляційні ефекти. Ця інформація допомагає інвесторам і керівникам фондів приймати обґрунтовані рішення щодо майбутніх інвестицій [1]-[6].

Порівняння результатів портфеля з еталонними показниками є стандартним підходом до оцінки ефективності. Це дозволяє оцінити, чи портфель працює на рівні з аналогічними, та надає корисну інформацію про результативність інвестиційної стратегії [1]-[6].

Проаналізуємо операційні показники. Загальний дохід – це підсумкова величина прибутку або збитку, отриманого за певний період, виражена у відсотках від початкової суми інвестицій. Включає як проценти, так і додаткові доходи, такі як дивіденди або прибуток від продажу активів [1]-[6].

Середньорічний темп зростання є показником річного зростання компанії протягом певного періоду. Він зручний для оцінки ефективності різних інвестицій, оскільки згладжує короткострокові коливання та забезпечує показник стабільного зростання [1]-[6].

Коефіцієнт Шарпа є метрикою прибутковості з урахуванням ризику, що відображає додану вартість від кожного ризику, прийнятого портфелем. Вищий коефіцієнт свідчить про кращу адаптацію до ризику, показуючи, що портфель генерує більше прибутку на одиницю ризику [1]-[6].

Коефіцієнт Сортіно схожий на коефіцієнт Шарпа, але він враховує тільки ризик зниження. Це корисно для інвесторів, які більше зосереджені на уникненні збитків [1]-[6].

Показник альфи Дженсена використовується для вимірювання перевищення дохідності портфеля над очікуваним рівнем, враховуючи ринковий ризик, представлений бета-коефіцієнтом. Позитивна альфа свідчить про те, що портфель перевищив очікувану дохідність, а негативна альфа вказує на не досягнуті результати [1]-[6].

Коректний вибір методу порівняння є ключовим для точного оцінювання. Для реалістичної оцінки важливо враховувати відомі ризики та інвестиційні цілі, щоб порівняння портфеля було справедливим [1]-[8].

Фондові індекси є типовими еталонами для вимірювання ефективності інвестиційного портфеля. Вони відображають ринкові тенденції та можуть використовуватися для порівняння результатів портфеля із загальними ринковими показниками [1]-[8].

Групові порівняння також можна проводити зі схожими портфелями або інвесторами. Це допомагає оцінити ефективність менеджера у порівнянні з іншими професіоналами, які працюють за аналогічними фінансовими умовами [1]-[8].

Спеціальні засоби порівняння інвестиційних портфелів створюються для того, щоб враховувати унікальні характеристики ризику та прибутковості певного портфеля. Вони особливо корисні для оцінки результатів портфелів зі специфічними цілями або стратегіями, які не відображаються стандартними ринковими індексами чи груповими порівняннями [1]-[8].

Вимірювання ефективності портфеля за допомогою відповідних програмних засобів передбачає спостереження та аналіз двох ключових аспектів: дохідності та ризику. Для цього використовуються різні показники, які дозволяють інвестору оцінювати прогрес інвестицій з часом. Виділяються різні показники для оцінки ефективності та ризику портфеля [1]-[8].

Аналіз і моніторинг стану портфеля можуть здійснюватися за допомогою декількох показників. Хоча загальний дохід є важливим показником ефективності, він не відображає повної картини. Оцінка ризику, зокрема за допомогою таких показників, як стандартне відхилення, є не менш значущою [1]-[8].

Часто інвестори за допомогою відповідних програмних засобів створюють інвестиційну політику, яка визначає стратегію інвестування та може використовуватися як окремими інвесторами, так і професійними менеджерами та інвестиційними консультантами для встановлення

інвестиційних цілей, завдань та методів моніторингу результативності. Стратегію інвестування на основі програмних засобів також окреслює мету портфеля та показники, які допомагають відстежити, чи відповідає портфель поставленим завданням [1]-[8].

Основні параметри, що використовуються для оцінювання інвестиційного портфелю, можуть включати специфічні показники для моніторингу, інвестиційний часовий горизонт, еталон для порівняння ефективності, а також інші критерії вибору інвестицій та умови для заміни активів, які не відповідають встановленим стандартам або демонструють недостатню ефективність [1]-[8].

Оцінювання результативності портфеля є важливою частиною інвестиційного процесу, що допомагає інвесторам і портфельним менеджерам ухвалювати зважені рішення та вдосконалювати стратегії. Аналізуючи ризики, прибутковість і ефективність з урахуванням ризику, спеціалізовані програмні засоби можуть виявити сильні та слабкі сторони своїх підходів і внести відповідні корективи [1]-[8].

Можуть використовуватися різні еталонні показники для відстеження ефективності портфеля або оцінки результативності окремих активів. Основним показником для моніторингу портфеля є загальна дохідність, яку зазвичай порівнюють з еталонним індексом. До інших показників належать статистичні методи оцінки ризику [1]-[8].

Визначено, що інвестиційний портфель являє собою набір фінансових активів, таких як акції, облігації, нерухомість, інвестиційні фонди та інші інструменти, які інвестор використовує для досягнення своїх фінансових цілей. Основною метою створення інвестиційного портфеля є оптимізація доходу при мінімізації ризиків шляхом диверсифікації, розподілу активів серед різних класів, секторів або географічних регіонів. Портфель формується з урахуванням таких факторів, як рівень ризику, який інвестор готовий прийняти, тривалість інвестицій, очікуваний дохід та ринкові умови. Ефективне управління портфелем передбачає використання спеціалізованого

програмного забезпечення, яке допомагає підтримувати постійний аналіз та коригування його складу відповідно до змін на ринку та фінансових цілей. Аналіз портфеля передбачає оцінку інвестиційного портфеля з метою перевірки його відповідності фінансовим цілям та уподобанням. Він також дозволяє визначити, чи досягаються встановлені для портфеля показники, зокрема стосовно ризику, інфляції та продуктивності.

1.2 Аналіз програмного забезпечення для оцінювання інвестиційних портфелів

Проаналізуємо програмне забезпечення для оцінювання інвестиційних портфелів [9]-[15].

Програмне забезпечення Quicken (рис. 1.1) є платформою для управління портфелем на сучасному ринку. Програмне забезпечення Quicken дозволяє обробляти різні типи активів шляхом інтеграції з брокерськими, банківськими чи борговими рахунками для автоматичного імпорту даних або за допомогою ручного введення інформації [9]-[13].

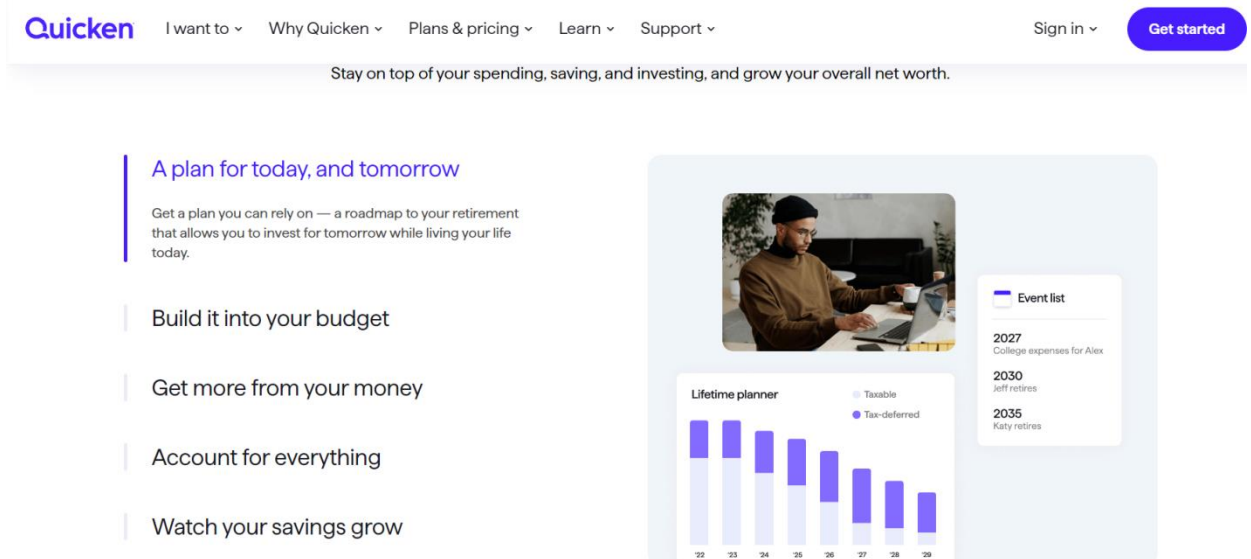


Рисунок 1.1 – Програмне забезпечення Quicken [13]

Quicken підходить як для новачків, які починають управляти своїми інвестиціями, так і для досвідчених інвесторів, що відстежують опціони, акції, облігації та інші інструменти. Завдяки функції ручного введення можливе відстеження майже будь-якого активу [9]-[13].

Simplifi є новим доповненням до інструментів фінансового управління Quicken. Це зручний застосунок для бюджетування та особистих фінансів, який включає базові функції управління портфелем, що робить його ідеальним вибором для початківців. Користувачі можуть підключити свої фінансові рахунки до Simplifi, включаючи чекові та ощадні рахунки, кредитні картки, позики та інвестиційні рахунки. Програмне забезпечення Simplifi надає загальний огляд фінансового стану та пропонує рекомендації для покращення управління фінансами [9]-[13].

Програмне забезпечення Quicken можна використовувати разом з іншими програмами для оптимізації результатів управління інвестиційними портфелями, бюджетами та заощадженнями. Після синхронізації чекових, ощадних, боргових та інвестиційних рахунків користувач отримає доступ до комплексного фінансового порталу [9]-[13].

Quicken Premier є універсальним інструментом для фінансового планування, що поєднує в собі функції для базового управління грошима, складання бюджету, відстеження боргу і управління готівкою, а також забезпечує потужний інструментарій для управління інвестиційним портфелем [9]-[13].

Це програмне забезпечення для управління портфелем синхронізується з актуальними котируваннями акцій. У Quicken Premier є можливість створювати різноманітні звіти про фінанси та інвестиції, які можна детально налаштовувати, що дозволяє відстежувати інвестиційні рахунки, доходи, дивіденди і комісії. До звітів входять інформація про приріст капіталу, норми прибутку, ринкові порівняння, інвестиційні операції, витратний базис і податкові звіти. Крім того, доступний аналіз ризику і прибутковості, інструмент для пенсійного планування та детальний аналіз портфелю [9]-[13].

Quicken Premier також пропонує функції для відстеження опціонних торгів, а також включає надійні інструменти для управління доходами, витратами, бюджетуванням і загальним фінансовим управлінням [9]-[13].

Програма забезпечує комплексний аналіз всіх аспектів фінансового життя, включаючи заощадження, витрати, борги, інвестиції, нерухомість і товари. Включені також розширені звіти та інструменти для пенсійного планування. Приклади звітів включають приріст капіталу, інвестиційну діяльність, розподіл активів, дохід від інвестицій, ефективність інвестицій, вартість портфеля, витратний базис і внутрішню норму прибутку [9]-[13].

Quicken надає повний набір інструментів для відстеження бюджету, заощаджень, витрат і управління інвестиціями, з можливістю завантаження звітів у електронні таблиці для подальшого аналізу. Програма не має обмежень щодо історичних даних, і всю інформацію можна легко завантажити, включаючи транзакції, залишки на рахунках і вартість інвестицій. Існує багато варіантів підписки на Quicken, що дозволяють обрати найбільш підходящий план [9]-[13].

Основними характеристиками та особливостями програмного забезпечення Quicken є такі [9]-[13]:

- засоби управління грошима та бюджетування: Quicken дозволяє відстежувати доходи, витрати та створювати бюджети. Користувачі можуть переглядати детальну інформацію про свої фінансові операції та управляти витратами на основі категорій [9]-[13];

- засоби відстеження боргів: можливість відстежувати борги, такі як кредити та іпотеки, з можливістю перегляду графіків погашення і розрахунку процентів [9]-[13];

- управління інвестиціями: Quicken надає інструменти для моніторингу і управління інвестиційними портфелями, включаючи акції, облігації, опціони та інші цінні папери. Програма синхронізується з котируваннями акцій у реальному часі і використовує аналітичний інструмент для детального аналізу диверсифікації інвестиційного портфеля [9]-[13];

- створення звітів: програмне забезпечення Quicken пропонує можливість створення і налаштування різноманітних фінансових звітів, таких як приріст капіталу, дохід від інвестицій, ефективність інвестицій, витратний базис та інші [9]-[13];

- пенсійне планування: програмне забезпечення Quicken включає інструменти для планування пенсійних заощаджень і оцінки майбутніх фінансових потреб [9]-[13];

- імпорт і синхронізація даних: користувачі можуть імпортувати дані з банківських і брокерських рахунків, що полегшує управління фінансами і зменшує потребу в ручному введенні даних [9]-[13];

- аналіз ризику і прибутковості: програмне забезпечення Quicken надає можливість проведення аналізу ризику і прибутковості для оцінки ефективності інвестиційних стратегій [9]-[13];

- налаштування звітів: усі звіти можна налаштовувати відповідно до індивідуальних потреб і експортувати в електронні таблиці для подальшого аналізу [9]-[13];

- оновлення і підтримка історичних даних: програма забезпечує доступ до історичних даних, що дозволяє аналізувати фінансові тренди протягом тривалого часу [9]-[13].

Серед недоліків програмного забезпечення Quicken можна відзначити те, що підписка може бути досить дорогою, особливо для користувачів, які не потребують всіх функцій програми. Іншим недоліком є те, що імпорт історичних даних з деяких фінансових установ може бути проблематичним, і користувачі можуть зустрічати труднощі з відновленням старих транзакцій. Крім того, хоча програмне забезпечення Quicken підтримує багато фінансових платформ, можливості інтеграції з деякими новими або менш відомими платформами можуть бути обмеженими [9]-[13].

Програмне забезпечення Empower (рис. 1.2) пропонує широкий набір програмних інструментів для управління інвестиціями та їх відстеження, а також платні послуги з управління інвестиціями та фінансового планування.

Реєстрація в системі є простою, і користувачі можуть підключати свої брокерські, банківські та кредитні рахунки. Крім того, є можливість додати свою нерухомість і отримати актуальну оцінку вартості через онлайн портал. Після підключення рахунків, інформаційна панель дає повний огляд інвестицій, витрат, заощаджень, боргів і чистого капіталу [9]-[11], [14].

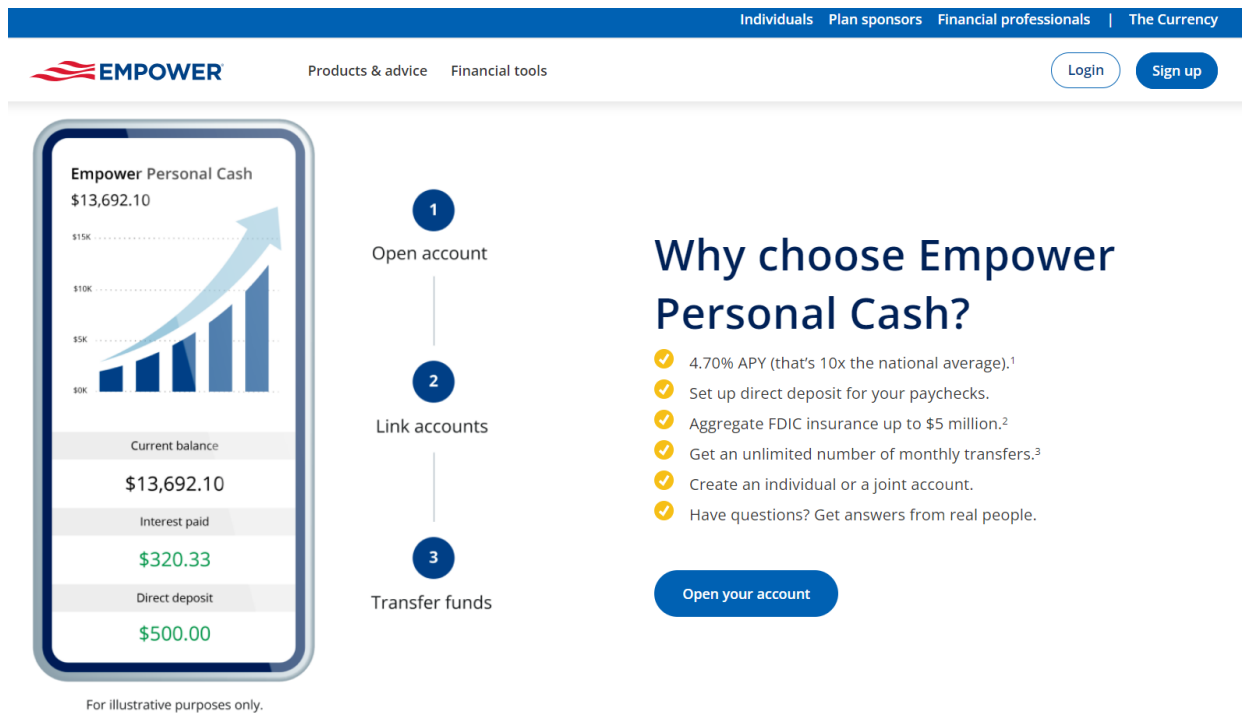


Рисунок 1.2 – Програмне забезпечення Empower [14]

Програмне забезпечення Empower є одним із найкращих варіантів для управління інвестиціями завдяки надійним інструментам моніторингу портфеля. Платформа також підходить для досвідчених інвесторів, які шукають робо-консультанта з увагою до витрат та самостійного контролю. Після підключення інвестиційних рахунків, Empower збирає дані та надає повний огляд капіталу разом із оцінкою вартості нерухомості. Платформа Empower пропонує різні звіти та аналізи, охоплюючи банківські, інвестиційні та планувальні інструменти, де кожен напрямок містить широкий вибір аналітичних звітів [9]-[11], [14].

Аналізатор пенсійних виплат у Empower визначає загальний коефіцієнт витрат для інвестицій і показує суму комісій, яку сплачується з часом, а також

їхній вплив на прибутковість портфеля. Планувальні функції включають аналіз заощаджень, комісій, інвестицій, а також пенсійний планувальник, який дозволяє тестувати різні сценарії заощаджень і витрат [9]-[11], [14].

Платформа доступна на мобільних і настільних пристроях, швидко синхронізує рахунки, після чого відкривається персоналізована інформаційна панель. Вона об'єднує всі банківські, кредитні та інвестиційні рахунки, надаючи огляд інвестицій та іншої фінансової інформації, як-от заощадження й бюджет [9]-[11], [14].

Доступні звіти й інструменти аналізу, включаючи дані про чистий капітал, залишки, бюджет, грошові потоки, доходи, витрати та повернення інвестицій. Пенсійний планувальник дозволяє моделювати різні сценарії, враховуючи непередбачені доходи або великі витрати, як-от плата за навчання дитини. Аналізатор інвестицій оцінює розподіл активів і пропонує рекомендації для покращення [9]-[11], [14].

Програмне забезпечення Empower має такі особливості [9]-[11], [14]:

- інструменти для управління інвестиціями: програмне забезпечення Empower надає широкий набір інструментів для відстеження та аналізу інвестицій, що дозволяє отримати детальний огляд фінансового стану без додаткових витрат [9]-[11], [14];

- засоби огляду фінансів: програмне забезпечення Empower дозволяє зв'язати всі інвестиційні, банківські та кредитні рахунки, надаючи огляд активів, заощаджень, витрат, боргів і чистого капіталу. Це забезпечує комплексний підхід до управління фінансами [9]-[11], [14];

- пенсійний планувальник: програмне забезпечення Empower пропонує потужний пенсійний планувальник, що дозволяє аналізувати різні сценарії заощаджень і витрат, включаючи можливість оцінки майбутніх доходів і витрат, таких як оплата навчання [9]-[11], [14];

- аналіз інвестицій і комісій: програмне забезпечення Empower містить аналізатор інвестицій і комісій, що дозволяє оцінювати, як комісії впливають на загальну прибутковість портфеля з плином часу [9]-[11], [14];

- інтерактивна інформаційна панель: після синхронізації рахунків користувач отримує персоналізовану панель, яка містить усі фінансові дані. Це забезпечує простий доступ до важливої інформації про інвестиції, бюджет і заощадження [9]-[11], [14];

- звіти та аналіз: програмне забезпечення Empower надає різні звіти та аналітичні інструменти, такі як аналіз чистого капіталу, бюджетування, звіти про доходи, витрати, повернення інвестицій та розподіл активів [9]-[11], [14];

- розширені інвестиційні інструменти: користувачі можуть отримати інформацію про структуру активів, ефективність секторів і статистику портфеля, що дозволяє краще керувати своїми інвестиціями [9]-[11], [14];

- мобільний та настільний доступ: програмне забезпечення Empower доступне як на мобільних, так і на настільних пристроях, забезпечуючи гнучкість і зручність у використанні [9]-[11], [14].

Серед недоліків програмного забезпечення Empower можна відзначити те, що підписка може бути досить дорогою, особливо для користувачів, які не потребують всіх функцій програми. Іншим недоліком є те, що імпорт історичних даних з деяких фінансових установ може бути проблематичним, і користувачі можуть зустрічати труднощі з відновленням старих транзакцій. Крім того, хоча програмне забезпечення Empower підтримує багато фінансових платформ, можливості інтеграції з деякими новими або менш відомими платформами можуть бути обмеженими [9]-[11], [14].

Програмне забезпечення SigFig (рис. 1.3) підтримує процес управління інвестиційним портфелем як частину робо-консультаційних послуг з управління інвестиціями. Користувач може створити обліковий запис, синхронізувати фінансові рахунки, додавати активи вручну та користуватися інструментами аналізу інвестицій, які надає платформа. Після відповіді на кілька запитань, що стосуються фінансових цілей і схильності до ризику, програмна система використовує ці дані для створення відповідного портфеля і проведення аналізу наявних інвестицій [9], [10], [15].

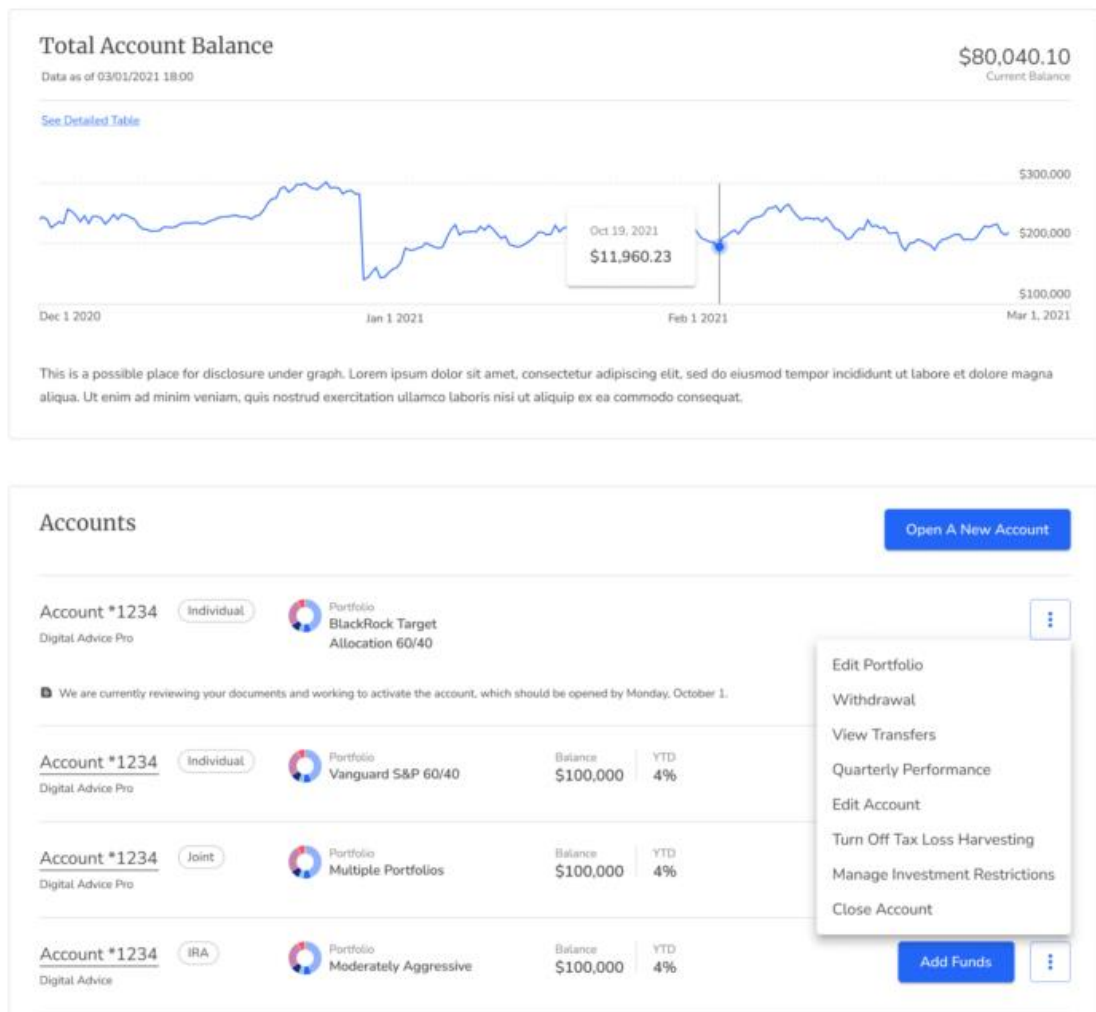


Рисунок 1.3 – Програмне забезпечення SigFig [15]

Програмне забезпечення SigFig пропонує онлайн трекер інвестиційного портфеля, надаючи можливість відстежувати взаємні фонди та акції за допомогою відповідного програмного забезпечення [9], [10], [15].

SigFig надає можливість керувати інвестиційними портфелями. Ця функція є особливо корисною для власників великих портфелів, які прагнуть уникнути податкових наслідків і складнощів, пов'язаних із передачею існуючих інвестицій [9], [10], [15].

Платформа SigFig спеціалізується на оптимізації портфелів, пропонуючи звичайним інвесторам інструменти для відстеження активів. Партнерство з Fidelity і Schwab дозволяє компанії застосовувати алгоритмічні інвестиційні стратегії для аналізу, моніторингу та поліпшення портфелів.

Системи автоматично балансують і диверсифікують інвестиції, знижуючи ризики та мінімізуючи комісії [9], [10], [15].

Технологія управління активами SigFig аналізує поточні інвестиції користувачів, враховує їхні фінансові цілі та схильність до ризику, після чого розробляє індивідуальну стратегію. Системи автоматично відстежують і оптимізують портфелі, мінімізуючи комісії за управління [9], [10], [15].

Користувачам надається звіт із рекомендаціями щодо портфеля, який формується на основі відповідей на початкові питання про ризики, з урахуванням змін цін, розподілу активів, коефіцієнта витрат, диверсифікації та інших факторів [9], [10], [15].

Хоча платформа не пропонує багато можливостей для налаштування звітів і відмовляється від інструментів для управління витратами, заощадженнями та бюджетом, вона залишається зручним інструментом для отримання швидкого огляду стану інвестиційного портфеля в порівнянні з рекомендованим розподілом активів [9], [10], [15].

Програмне забезпечення SigFig має такі особливості [9], [10], [15]:

- інтеграція з великими брокерами: програмне забезпечення SigFig підтримує інвесторів із різними рахунками, дозволяючи керувати інвестиціями без необхідності переносу активів [9], [10], [15];

- автоматизоване управління портфелем: програмне забезпечення SigFig використовує алгоритмічні стратегії для балансування та диверсифікації інвестиційних портфелів, знижуючи ризики та оптимізуючи прибутки [9], [10], [15];

- аналіз та рекомендації: програма SigFig проводить детальний аналіз поточних інвестицій і пропонує персоналізовані рекомендації на основі фінансових цілей і стійкості до ризику користувача [9], [10], [15];

- прогнози: платформа надає прогнозовану довгострокову вартість портфеля на основі запропонованих інвестиційних стратегій [9], [10], [15];

- автоматичне балансування: постійний моніторинг і автоматичне балансування інвестицій для підтримки оптимального портфеля [9], [10], [15];

– інструменти портфельного аналізу: проводиться аналіз динаміки цін, розподілу активів, географічної диверсифікації, коефіцієнтів витрат і концентрації активів [9], [10], [15];

– відстеження у режимі реального часу: інвестори можуть отримувати показники ефективності, прибутки та збитки в режимі реального часу, а також інші важливі інвестиційні показники [9], [10], [15].

Серед недоліків програмного забезпечення SigFig можна відзначити те, що підписка може бути досить дорогою, особливо для користувачів, які не потребують всіх функцій програми. Іншим недоліком є те, що імпорт історичних даних з деяких фінансових установ може бути проблематичним, і користувачі можуть зустрічати труднощі з відновленням старих транзакцій. Крім того, хоча програмне забезпечення SigFig підтримує багато фінансових платформ, можливості інтеграції з деякими новими або менш відомими платформами можуть бути обмеженими [9], [10], [15].

Порівняльну характеристику програмного забезпечення для оцінювання інвестиційних портфелів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для оцінювання інвестиційних портфелів

Критерій порівняння	Quicken [13]	Empower [14]	SigFig [15]
Засоби аналізу ринку та прибутковості	+—	+	+—
Засоби управління грошима та бюджетування	+	+—	+—
Управління інвестиціями	+	+—	+—
Створення звітів	+	+	+—
Імпорт і синхронізація даних	+—	—	—
Оновлення і підтримка роботи з історичними даними	+	—	—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для оцінювання інвестиційних портфелів. Проте деякі програмні засоби для оцінювання інвестиційних портфелів можуть бути досить дорогими в розробці і впровадженні, а також в обслуговуванні та оновленні.

Крім того, програмне забезпечення для оцінювання інвестиційних портфелів залежить від якості вхідних даних. Якщо при синтезі розпізнавальних моделей, що використовуються у відповідному програмному забезпеченні, було використано неточні або неповні дані, результати аналізу можуть бути спотворені або ненадійні. Програмне забезпечення для оцінювання інвестиційних портфелів часто використовує складні алгоритми для прогнозування та оцінки.

Неправильне налаштування алгоритмів або використання недосконалих та неефективних алгоритмів можуть призвести до помилкових висновків і ризиків. Тому актуальною є розробка програмного забезпечення для оцінювання інвестиційних портфелів.

При розробці програмного забезпечення для оцінювання інвестиційних портфелів необхідно забезпечити такі функціональні вимоги:

- підтримка можливості формування інвестиційного портфелю;
- можливість оцінювання показників ефективності інвестиційних портфелів;
- можливість прогнозування вартості інвестиційних портфелів;
- можливість використання імітаційного моделювання для оцінювання інвестиційних портфелів;
- підтримка можливості оцінювання систематичного ризику інвестиційних портфелів;
- підтримка можливості графічного відображення дохідності інвестиційного портфелю.

1.3 Висновки за розділом 1

Визначено, що інвестиційний портфель являє собою набір фінансових активів, таких як акції, облігації, нерухомість, інвестиційні фонди та інші інструменти, які інвестор використовує для досягнення своїх фінансових цілей. Основною метою створення інвестиційного портфеля є оптимізація доходу при мінімізації ризиків шляхом диверсифікації, розподілу активів серед різних класів, секторів або географічних регіонів. Портфель формується з урахуванням таких факторів, як рівень ризику, який інвестор готовий прийняти, тривалість інвестицій, очікуваний дохід та ринкові умови. Ефективне управління портфелем передбачає використання спеціалізованого програмного забезпечення, яке допомагає підтримувати постійний аналіз та коригування його складу відповідно до змін на ринку та фінансових цілей. Аналіз портфеля передбачає оцінку інвестиційного портфеля з метою перевірки його відповідності фінансовим цілям та уподобанням. Він також дозволяє визначити, чи досягаються встановлені для портфеля показники, зокрема стосовно ризику, інфляції та продуктивності.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для оцінювання інвестиційних портфелів. Проте деякі програмні засоби для оцінювання інвестиційних портфелів можуть бути досить дорогими в розробці і впровадженні, а також в обслуговуванні та оновленні. Крім того, програмне забезпечення для оцінювання інвестиційних портфелів залежить від якості вхідних даних. Якщо при синтезі розпізнавальних моделей, що використовуються у відповідному програмному забезпеченні, було використано неточні або неповні дані, результати аналізу можуть бути спотворені або ненадійні. Програмне забезпечення для оцінювання інвестиційних портфелів часто використовує складні алгоритми для прогнозування та оцінки. Неправильне налаштування алгоритмів або використання недосконалих та неефективних алгоритмів

можуть призвести до помилкових висновків і ризиків. Тому актуальною є розробка програмного забезпечення для оцінювання інвестиційних портфелів.

Сформульовано функціональні вимоги до програмного забезпечення для оцінювання інвестиційних портфелів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для оцінювання інвестиційних портфелів було використано мову програмування Python [16], [17].

Python має широкий діапазон застосувань, таких як ігри, мобільні додатки, графічні інструменти тощо. Завдяки своїй універсальності та багатому лексикону, вона легко зрозуміла [16], [17].

Python пропонує одну з найбільших стандартних бібліотек, яка охоплює такі області, як телекомунікації, мережеві додатки, вебтехнології, що базуються на передових підходах. Це робить цю мову ефективним вибором для розробки програмного забезпечення для аналізу даних, на відміну від багатьох інших мов [16], [17].

Мова програмування Python має велику кількість підтримуючих бібліотек і потужні інструменти, що дозволяє підвищити продуктивність розробників у кілька разів порівняно з іншими мовами [16], [17].

Мова Python легко використовувати для розробки вебзастосунків, а також для інтеграції з різними компонентами. Завдяки своїм потужним можливостям управління, Python може бути пов'язаний із іншими мовами через відповідний засіб, а також підтримує роботу з мовами розмітки. Всі сучасні операційні системи можуть виконувати код Python [16], [17].

Мова Python має надійну інтеграцію, можливості модульного тестування та розширене управління ресурсами, що підвищує гнучкість і продуктивність додатків. Це робить мову програмування Python чудовим вибором для розробки великих мережевих додатків, що підтримують декілька протоколів [16], [17].

Мова програмування Python розробляється під ліцензією з відкритим кодом, що дозволяє використовувати, поширювати і комерціалізувати код, а також сприяти розвитку спільноти [16], [17].

Мова програмування Python має вбудовані структури даних, такі як списки та словники, які можуть бути використані для швидкої обробки інформації. Крім того, він забезпечує динамічний доступ до баз даних та дозволяє скорочувати код до мінімальної необхідної довжини [16], [17].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для оцінювання інвестиційних портфелів наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для оцінювання інвестиційних портфелів

Критерій порівняння мов програмування	Мова програмування		
	Python	Java	PHP
Простий синтаксис	+	+—	+
Велика спільнота розробників	+	+—	+
Бібліотеки для аналізу даних	+	+	—
Зручність для створення ПЗ для оцінювання інвестиційних портфелів	+	—	—
Інтеграція з інструментами машинного навчання	+	+—	+—

Отже, для реалізації програмного забезпечення для оцінювання інвестиційних портфелів обрано мову програмування Python, яка відома своєю простотою та читабельністю, що значно полегшує розробку складних алгоритмів, які часто використовуються для фінансового аналізу та оцінки ризиків. Зрозумілий синтаксис мови Python дозволяє швидко розробляти та тестувати різні моделі, що особливо важливо у сфері інвестицій, де ефективність та точність є ключовими. Крім того, Python має велику кількість бібліотек для аналізу даних, що дозволяють легко працювати з різними

наборами фінансових даних, проводити статистичні обчислення та моделювання.

2.2 Вибір середовища розробки для створення програмного забезпечення для оцінювання інвестиційних портфелів

В якості середовища розробки для створення програмного забезпечення для оцінювання інвестиційних портфелів було обрано середовище PyCharm [18], [19].

Середовище розробки PyCharm є доцільним вибором для створення програмного забезпечення для оцінювання інвестиційних портфелів через кілька ключових переваг. По-перше, це інтегроване середовище розробки спеціалізується на Python, що є однією з найкращих мов для фінансового аналізу та роботи з даними. PyCharm надає потужні інструменти для кодування, що спрощують розробку складних програмних рішень, зокрема для моделювання інвестиційних портфелів [18], [19].

PyCharm підтримує інтеграцію з популярними бібліотеками для аналізу даних, як Pandas, NumPy, і SciPy, та інструментами для машинного навчання, такими як TensorFlow та scikit-learn. Це робить розробку математичних і фінансових моделей швидшою та більш ефективною. Завдяки вбудованим засобам рефакторингу коду, розробники можуть легко підтримувати високу якість коду, що є важливим для довготривалих проєктів [18], [19].

Іншою важливою перевагою PyCharm є його потужний відладчик, що дозволяє розробникам детально аналізувати поведінку програми, знаходити та виправляти помилки на ранніх етапах розробки. Це критично важливо для розробки програмного забезпечення, яке працює з фінансовими даними, де помилки можуть призвести до великих втрат [18], [19].

PyCharm також підтримує інтеграцію з системами контролю версій (Git, Mercurial, Subversion), що забезпечує ефективну роботу в команді і дозволяє контролювати всі зміни в коді. Крім того, PyCharm має зручні

інструменти для тестування, що допомагає розробляти надійне програмне забезпечення для оцінювання інвестиційних портфелів [18], [19].

Нарешті, середовище розробки PyCharm підтримує роботу з віртуальними середовищами та має можливості для налаштування середовища розробки під конкретні потреби, що дозволяє ізолювати проекти та легко керувати залежностями, особливо коли програмне забезпечення працює з великими обсягами даних [18], [19].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для оцінювання інвестиційних портфелів наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для оцінювання інвестиційних портфелів

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Spyder	Jupyter Notebook
Зручність роботи з системами контролю версій	+	—	—
Спеціалізація на мові Python	+	+—	—
Інтеграція з бібліотеками для аналізу даних та інструментами для машинного навчання	+	+—	+
Робота з віртуальними середовищами	+	—	+—
Засоби рефакторингу коду	+	+	+—

Таким чином, для створення програмного забезпечення для оцінювання інвестиційних портфелів обрано середовище розробки PyCharm, що

спеціалізується на мові Python та надає потужні інструменти для кодування, які спрощують розробку складних програмних рішень, зокрема для моделювання інвестиційних портфелів. PyCharm підтримує інтеграцію з популярними бібліотеками для аналізу даних та інструментами для машинного навчання.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів

З метою можливості розробки програмного забезпечення для оцінювання інвестиційних портфелів створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів враховуємо те, що користувачі програми мають можливості формування інвестиційного портфелю, оцінювання показників ефективності інвестиційних портфелів, прогнозування вартості інвестиційних портфелів, використання імітаційного моделювання для оцінювання інвестиційних портфелів, оцінювання систематичного ризику інвестиційних портфелів, графічного відображення дохідності інвестиційного портфелю.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для оцінювання інвестиційних портфелів можна описати таким чином:

- користувач програмного забезпечення для оцінювання інвестиційних портфелів запускає програму та вводить початкові значення ваг проєктів інвестиційного портфелю;
- після введення ваг програмне забезпечення виконує формування інвестиційного портфелю;
- користувач визначає статистичні оцінки інвестиційного портфелю з різних проєктів;

- користувач визначає інші оцінки за допомогою методів історичного та імітаційного моделювання, а також методу параметричної оцінки;
- користувач завершує роботу з програмним забезпеченням для оцінювання інвестиційних портфелів.

Враховуючи сценарії роботи користувачів при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.



Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для оцінювання інвестиційних портфелів.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для оцінювання інвестиційних портфелів обрано мову програмування Python, яка відома своєю простотою та читабельністю, що значно полегшує розробку складних алгоритмів, які часто використовуються для фінансового аналізу та оцінки ризиків. Зрозумілий синтаксис мови Python дозволяє швидко розробляти та тестувати різні моделі, що особливо важливо у сфері інвестицій, де ефективність та точність є ключовими. Крім того, Python має велику кількість бібліотек для аналізу даних, що дозволяють легко працювати з різними наборами фінансових даних, проводити статистичні обчислення та моделювання.

Для створення програмного забезпечення для оцінювання інвестиційних портфелів обрано середовище розробки PyCharm, що спеціалізується на мові Python та надає потужні інструменти для кодування, які спрощують розробку складних програмних рішень, зокрема для моделювання інвестиційних портфелів. PyCharm підтримує інтеграцію з популярними бібліотеками для аналізу даних та інструментами для машинного навчання.

Створено модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для оцінювання інвестиційних портфелів.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для оцінювання інвестиційних портфелів

Структуру програмного забезпечення для оцінювання інвестиційних портфелів наведено на рис. 3.1.

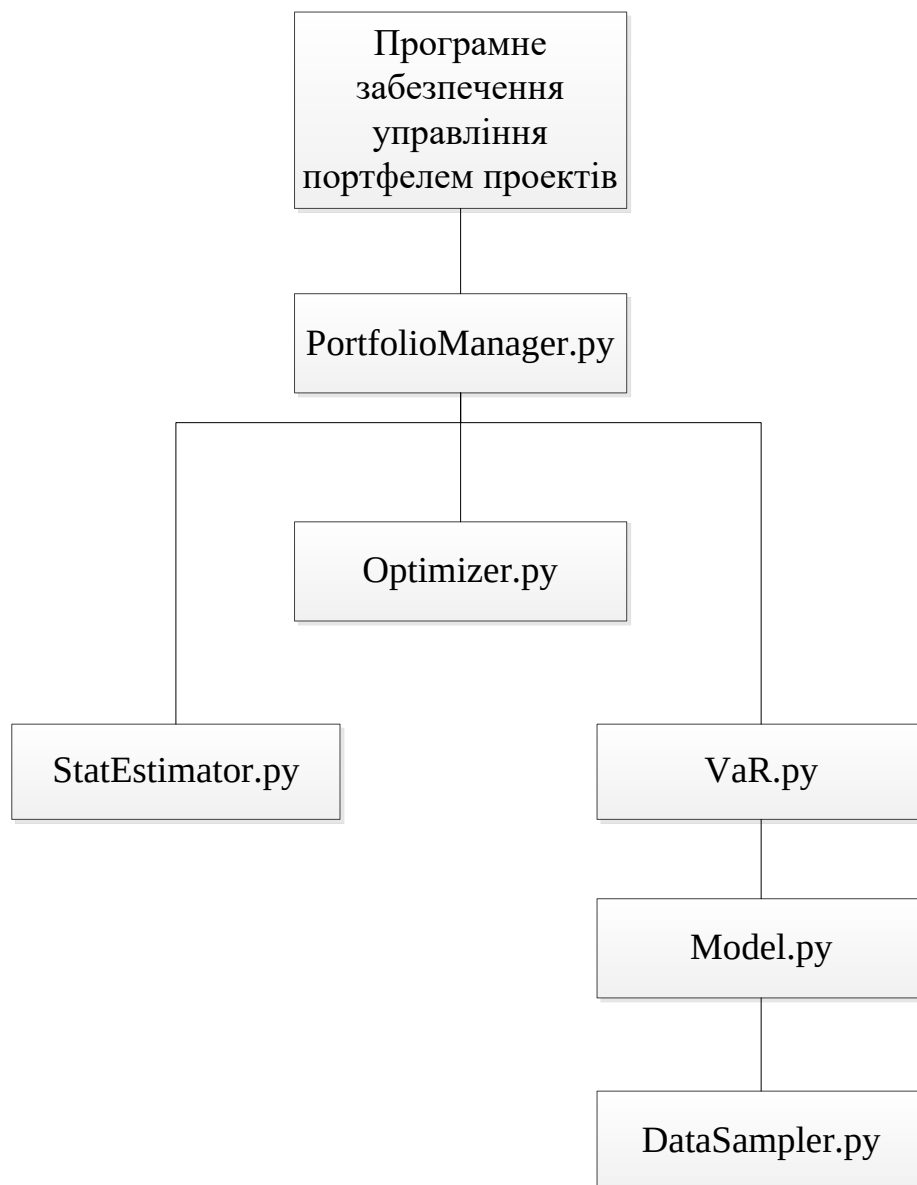


Рисунок 3.1 – Структура програмного забезпечення для оцінювання інвестиційних портфелів

Як видно з рис. 3.1, програмне забезпечення для оцінювання інвестиційних портфелів складається з декількох основних модулів.

Optimizer.py відповідає за роботу з користувачем під час налаштування оптимального портфеля проєктів, допомагаючи встановлювати початкові параметри, вирішувати результати, налаштовувати формули, відкривати відповідні вкладки, а також оптимізувати процес виконання.

PortfolioManager.py є базовим файлом проєкту, який визначає базову структуру форми і координує роботу всіх інших файлів, забезпечуючи запуск процесу роботи з користувачем.

StatEstimator.py реалізує статистичну оцінку окремих проєктів та портфелів, створюючи калькулятор засоби обчислення, відкриваючи вкладки та генеруючи необхідні статистичні дані.

Model.py надає необхідні функції для розробки моделі прогнозування вартості проєктів на основі технологій глибокого навчання, включаючи навчання, прогнозування та створення відповідної моделі у вигляді нейронної мережі.

VaR.py відповідає за управління всіма системами, пов'язаними з розрахунком критеріїв VaR, надаючи засоби обчислення, відкриваючи відповідні вкладки і виконуючи всі процеси, що стосуються розрахунку оптимального інвестиційного портфеля.

DataSampler.py забезпечує зразки набору даних для налаштування моделі прогнозування вартості портфелю, яка використовується для оцінювання інвестиційних портфелів.

3.2 Функціонування програмного забезпечення для оцінювання інвестиційних портфелів

Функціонування програмного забезпечення для оцінювання інвестиційних портфелів подамо за допомогою схеми, зображеної на рис. 3.2.

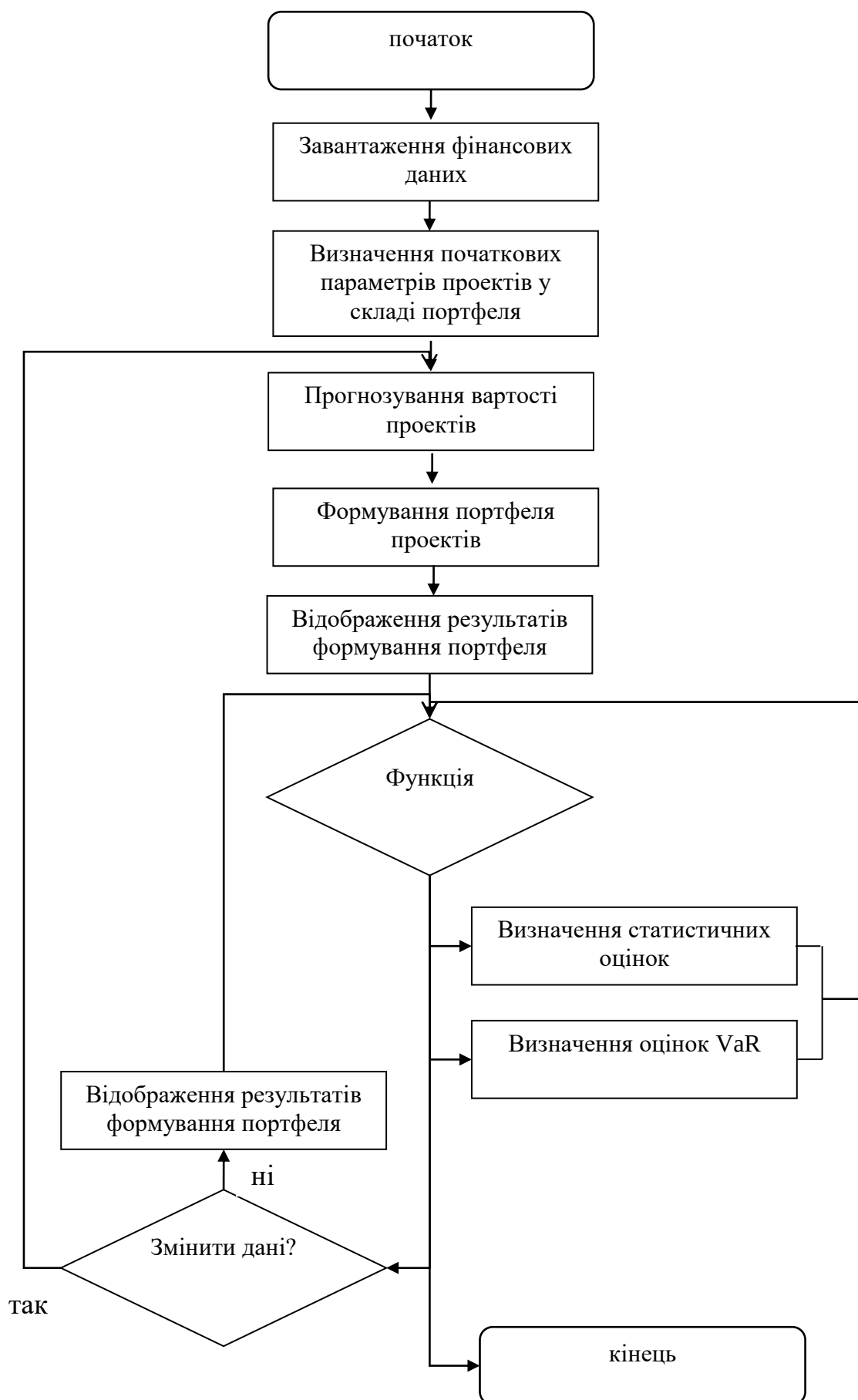


Рисунок 3.2 – Схема функціонування програмного забезпечення для оцінювання інвестиційних портфелів

Програма запускається після введення користувачем своїх фінансових даних. При цьому необхідно вказати портфель проєктів, назви проєктів та початковий розподіл фінансових ресурсів, які будуть використовуватися в портфелі. Далі генерується звіт про вартість проєкту, для чого модель отримує дані через відповідні приймачі. Якщо модель вже підготовлена, вона завантажується і використовує наявні дані для прогнозування. Якщо модель не навчена, відбувається її створення та навчання на основі зібраних даних для подальшого тестування та використання в процесі.

Після отримання результатів аналізу, план портфеля зберігається згідно з оптимізаційною моделлю. Далі результати представляються користувачеві. Після цього користувач може вибрати, як саме відобразити результати: показати підсумки через бібліотеки; представити статистичні дані; розрахувати і показати оцінки VaR.

Користувач у процесі роботи з програмою може додати нову інформацію щодо портфеля, після чого модель буде оновлена. Якщо змін не потрібно, користувач може продовжувати роботу з оптимізованими даними, вибираючи потрібні опції для відображення результатів.

3.3 Використання нейронних мереж для оцінювання інвестиційних портфелів

У програмному забезпеченні для оцінювання інвестиційних портфелів використано нейронні мережі з довгою короткочасною пам'яттю [20]-[22].

Нейромережі з довготривалою та короткою пам'яттю (НДКП) представляють собою більш складні обчислювальні одиниці порівняно зі звичайними нейронами в прихованих шарах. Кожен з таких блоків зберігає ключові параметри, які визначають, чи потрібно звертатися до попередніх спостережень, і яке значення ігнорувати або відхиляти у разі помилок. Мережа НДКП, що включає кілька таких модулів, може виконувати обчислення паралельно і підключатися до глибших нейронних мереж. НДКП – це

універсальна система для обчислювальних завдань з великими обсягами даних. Вона здатна виконувати обчислення навіть на звичайних комп'ютерах або на великих відстанях завдяки своїм електронним інтерфейсам, що є її важливою перевагою [20]-[22].

Рекурентна нейронна мережа НДКП справляється з задачами класифікації, навіть якщо події тривають довго або повторюються. Вона уникає проблем з надлишковістю інформації та її деградацією під час навчання і використання. Крім того, НДКП успішно застосовується для таких завдань, як планування, розпізнавання дій, аналіз граматики, вивчення і створення музики, і робототехніка [20]-[22].

На рис. 3.3 зображена структура модулю мережі НДКП [20]-[22].

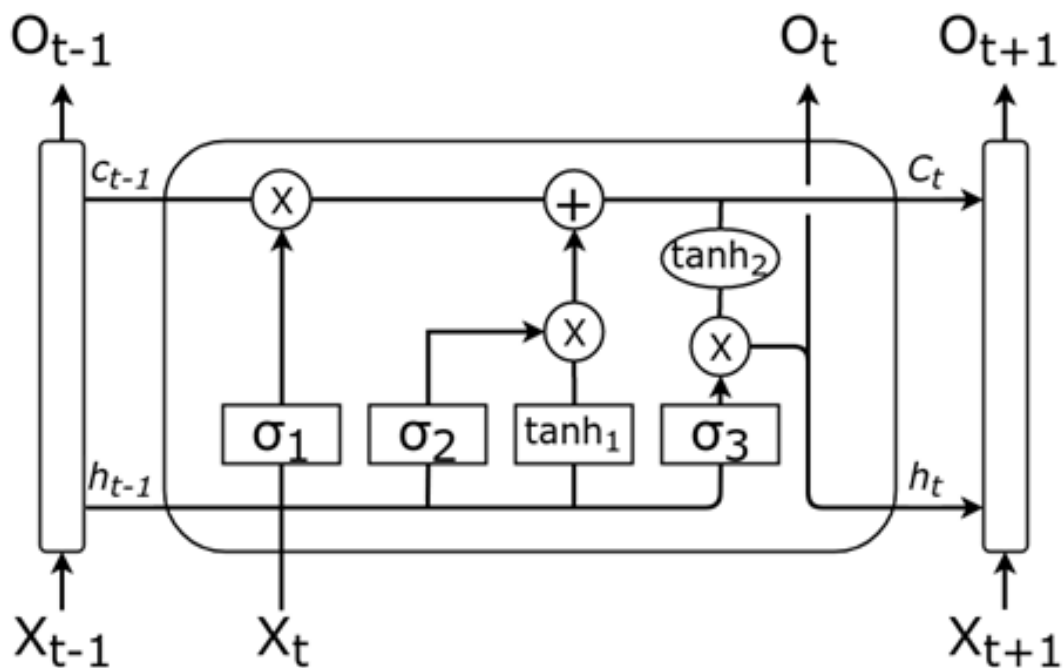


Рисунок 3.3 – Схема модулю нейромережі з довготривалою та короткою пам'яттю [20]-[22]

Стан модулю мережі НДКП проходить через увесь цикл, зазнаючи змін, і модуль отримує інформацію через фільтри, що проходять через сигмоподібну мережу та функцію множення точок, використовуючи різні перетворення у вигляді фільтрів [20]-[22].

Алгоритм функціонування модулю мережі НДКП виглядає так: спочатку визначається, яка інформація буде використана зі стану нейрону за допомогою шару σ_1 , який обробляє вхідні дані і видає коефіцієнт від 0 до 1. Далі, за допомогою шару σ_2 , обчислюється нова інформація про стан нейрону. Потім нові значення додаються до стану нейрону через функцію $\tanh$, а вихідні дані створюються через шар σ_3 , який визначає, які дані передати, після чого результати проходять через функцію $\tanh^2$ для отримання остаточних значень в інтервалі від -1 до 1, підсилених сигмоподібним шаром.

3.4 Проєктування інтерфейсу програмного забезпечення для оцінювання інвестиційних портфелів

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для оцінювання інвестиційних портфелів враховані функціональні вимоги до програми:

- підтримка можливості формування інвестиційного портфелю;
- можливість оцінювання показників ефективності інвестиційних портфелів;
- можливість прогнозування вартості інвестиційних портфелів;
- можливість використання імітаційного моделювання для оцінювання інвестиційних портфелів;
- підтримка можливості оцінювання систематичного ризику інвестиційних портфелів;
- підтримка можливості графічного відображення дохідності інвестиційного портфелю.

Інтерфейс головної форми програмного забезпечення для оцінювання інвестиційних портфелів наведено на рис. 3.4. Вкладка Статистичні оцінки інвестиційного портфелю зображена на рис. 3.5.

Дослідження і програмна реалізація процесів управління портфелями проектів

Вхідні дані | Статистичні оцінки | VAR-оцінки

Портфель проектів

Проект 1		Початкова вага проекту 1	
Проект 2		Початкова вага проекту 2	
Проект 3		Початкова вага проекту 3	

Дата формування портфеля

Рисунок 3.4 – Інтерфейс головної форми програмного забезпечення для оцінювання інвестиційних портфелів

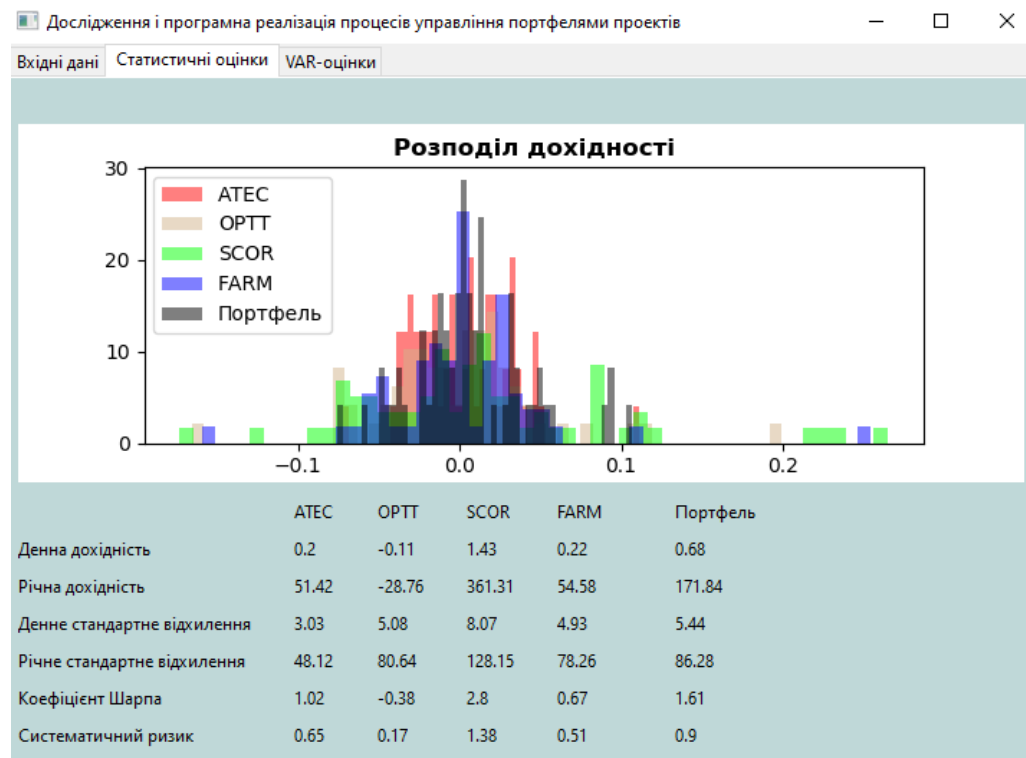


Рисунок 3.5 – Статистичні оцінки інвестиційного портфелю

Вкладка VaR-оцінки зображена на рис. 3.6.

Історичне моделювання		Імітаційне моделювання	
ATEC	-2.92	ATEC	-4.35
FNLC	-1.17	FNLC	-1.94
EXAS	-4.53	EXAS	-6.45
Портфель	-2.71	Портфель	-4.02

Параметрична оцінка	
ATEC	-4.34
FNLC	-1.93
EXAS	-6.47
Портфель	-4.02

Рисунок 3.6 – VaR-оцінки інвестиційного портфелю

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для оцінювання інвестиційних портфелів.

Запропоновано структуру програмного забезпечення для оцінювання інвестиційних портфелів. Визначено, що програма складається з декількох основних модулів. Модуль PortfolioManager.py є базовим файлом проєкту, який визначає базову структуру форми і координує роботу всіх інших файлів. Модулю StatEstimator.py реалізує статистичну оцінку окремих проєктів та портфелів, створюючи засоби обчислення, відкриваючи вкладки та генеруючи необхідні статистичні дані. Model.py надає необхідні функції для розробки моделі прогнозування вартості проєктів на основі технологій глибокого навчання, включаючи прогнозування та створення відповідної моделі у вигляді нейронної мережі.

Прийнято рішення про використання у програмному забезпеченні для оцінювання інвестиційних портфелів нейронних мереж з довгою короткочасною пам'яттю.

Описано функціонування програмного забезпечення для оцінювання інвестиційних портфелів.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу оцінювання інвестиційних портфелів. Програма написана на мові Python, яка відома своєю простотою та читабельністю, що значно полегшує розробку складних алгоритмів, які часто використовуються для фінансового аналізу та оцінки ризиків. В якості середовища розробки для створення програмного забезпечення для оцінювання інвестиційних портфелів обрано PyCharm, що спеціалізується на мові Python та надає потужні інструменти для кодування, які спрощують розробку складних програмних рішень.

Програмне забезпечення для оцінювання інвестиційних портфелів забезпечує виконання таких функцій:

- підтримка можливості формування інвестиційного портфелю;
- можливість оцінювання показників ефективності інвестиційних портфелів;
- можливість прогнозування вартості інвестиційних портфелів;
- можливість використання імітаційного моделювання для оцінювання інвестиційних портфелів;
- підтримка можливості оцінювання систематичного ризику інвестиційних портфелів;
- підтримка можливості графічного відображення дохідності інвестиційного портфелю.

Для експлуатації програмного забезпечення для оцінювання інвестиційних портфелів необхідно таке апаратне та програмне забезпечення:

- великий обсяг оперативної пам'яті (4 Гб або більше) для обробки наборів даних про інвестиційні портфелі;
- не менше 1 Гб вільного простору на диску для швидкого доступу до даних;

- багатоядерний процесор із тактовою частотою не менше 2.2 ГГц;
- операційна система для підтримки роботи програми.

Функціональні характеристики розробленого програмного забезпечення для оцінювання інвестиційних портфелів наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для оцінювання інвестиційних портфелів наведено у додатку Б.

4.2 Характеристики програми для оцінювання інвестиційних портфелів

Програма для оцінювання інвестиційних портфелів характеризується кількома аспектами, що забезпечують її ефективність. Вона має аналітичні можливості, що дозволяють аналізувати дані щодо інвестиційних портфелів та оцінювати ризики інвестицій. Для цього програма використовує алгоритми, щоб оцінювати інвестиційні показники й прогнозувати можливі результати.

Гнучкість програми дозволяє адаптувати її під різні типи інвестиційних портфелів, від акцій і облігацій до складніших фінансових інструментів, таких як похідні активи. Крім цього, програма забезпечує високий рівень безпеки для захисту чутливих фінансових даних та відповідає іншим вимогам, оскільки не є вебзастосунком та працює на комп'ютері користувача (фінансового аналітика). Її інтерфейс є зручним для користувачів та відповідає потребам аналітиків і інвесторів. Автоматизація процесів, таких як балансування портфеля, дозволяє зменшити кількість рутинних операцій і підвищує продуктивність.

Важливими характеристиками також є підтримка можливості графічного відображення розподілу дохідності інвестиційних портфелів, можливості моделювання інвестиційних показників за допомогою історичного та імітаційного моделювання.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для оцінювання інвестиційних портфелів необхідно запустити основний ру-файл. Це можна зробити у командному рядку, або в середовищі, що підтримує розробку програм на мові Python.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для оцінювання інвестиційних портфелів є інформація про інвестиційні проєкти, дати інвестицій, вагові коефіцієнти розподілу інвестиційних проєктів та ін.

Вихідними даними програмного забезпечення для оцінювання інвестиційних портфелів є оцінки ефективності інвестиційних портфелів, побудовані графіки розподілу дохідності портфелю та ін.

4.3.3 Повідомлення

Користувач програмного забезпечення для оцінювання інвестиційних портфелів може отримати такі повідомлення: повідомлення про некоректно введені дані, повідомлення про розраховані показники ефективності інвестиційного портфелю та ін.

4.4 Виконання програмного забезпечення для оцінювання інвестиційних портфелів

Після запуску програмного забезпечення для оцінювання інвестиційних портфелів користувачу відображається головна форма (рис. 4.1), де можна вводити вхідні дані, зокрема, назву проєктів, з яких складається інвестиційний

портфель, а також початкові значення їх вагових коефіцієнтів у портфелі. За замовченням пропонується ввести інформацію про три проекти, що є складовими інвестиційного портфелю.

The screenshot shows a software window titled "Дослідження і програмна реалізація процесів управління портфелями проектів". It has three tabs: "Вхідні дані" (selected), "Статистичні оцінки", and "VAR-оцінки". The main area is titled "Портфель проектів" and contains the following elements:

- Three rows for project data:
 - Project 1: Input field for weight, and "Початкова вага проекту 1" with an input field.
 - Project 2: Input field for weight, and "Початкова вага проекту 2" with an input field.
 - Project 3: Input field for weight, and "Початкова вага проекту 3" with an input field.
- A "Додати" (Add) button.
- A "Дата формування портфеля" (Portfolio formation date) label with an input field.
- "Визначити" (Specify) and "Формувати" (Form) buttons at the bottom.

Рисунок 4.1 – Головна форма програмного забезпечення для оцінювання інвестиційних портфелів

За необхідності додати нові проекти у портфель потрібно натиснути кнопку Додати та ввести інформацію про доданий проект.

Розширена форма введення вхідних даних у програмному забезпеченні для оцінювання інвестиційних портфелів наведена на рис. 4.2.

Дослідження і програмна реалізація процесів управління портфелями проектів

Вхідні дані | Статистичні оцінки | VAR-оцінки

Портфель проектів

Проект 1		Початкова вага проекту 1	
Проект 2		Початкова вага проекту 2	
Проект 3		Початкова вага проекту 3	
Проект 4		Початкова вага проекту 4	
Проект 5		Початкова вага проекту 5	
Проект 6		Початкова вага проекту 6	
Проект 7		Початкова вага проекту 7	
Проект 8		Початкова вага проекту 8	

Дата формування портфеля

Рисунок 4.2 – Розширена форма введення вхідних даних у програмному забезпеченні для оцінювання інвестиційних портфелів

При натисканні на кнопку **Визначити** відбувається процес оцінювання інвестиційного портфелю. Кнопка **Формувати** призначена для формування портфелю, зокрема, перерахунку вагових коефіцієнтів проєктів, що є складовими портфелю.

Приклад заповнення форми вхідних даних наведено на рис. 4.3.

Дослідження і програмна реалізація процесів управління портфелями проектів

Вхідні дані | Статистичні оцінки | VAR-оцінки

Портфель проектів

Проект 1	IBKC	Початкова вага проекту 1	0
Проект 2	CVCO	Початкова вага проекту 2	0.1
Проект 3	EXAS	Початкова вага проекту 3	0
Проект 4	OPTT	Початкова вага проекту 4	0.4
Проект 5	BBI	Початкова вага проекту 5	0.5

Додати

Дата формування портфеля: 12/10/2024

Визначити Формувати

Рисунок 4.3 – Приклад заповнення форми вхідних даних

Після завершення процесу оцінювання інвестиційного портфелю на вкладці Статистичні оцінки відображено інформацію про дохідність портфелю, деякі статистичні показники (зокрема, стандартне відхилення), а також систематичний ризик. Крім того, на цій вкладці відображено діаграму розподілу дохідності портфелю (рис. 4.4). Також виводиться значення коефіцієнту Шарпа для портфелю та кожної її складової, що є відношенням прибутку від вкладення грошей в інвестиційний портфель (річна дохідність) до коливань (річне стандартне відхилення).

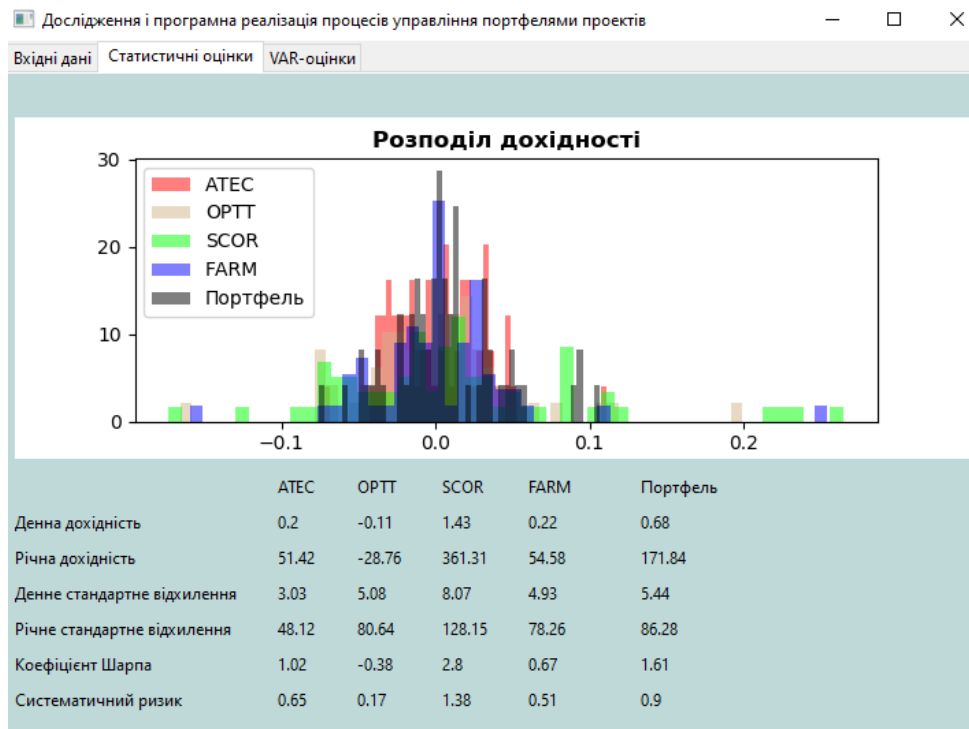


Рисунок 4.4 – Статистичні оцінки інвестиційного портфелю

На вкладці VaR-оцінки (рис. 4.5) відображаються відповідні характеристики інвестиційного портфелю, розраховані за допомогою методів історичного та імітаційного моделювання, а також методу параметричної оцінки.

Дослідження і програмна реалізація процесів управління портфелями проектів

Вхідні дані Статистичні оцінки **VAR-оцінки**

Історичне моделювання		Імітаційне моделювання	
ATEC	-2.92	ATEC	-4.35
FNLC	-1.17	FNLC	-1.94
EXAS	-4.53	EXAS	-6.45
Портфель	-2.71	Портфель	-4.02

Параметрична оцінка

ATEC	-4.34
FNLC	-1.93
EXAS	-6.47
Портфель	-4.02

Рисунок 4.5 – VaR-оцінки інвестиційного портфелю

4.5 Тестування програмного забезпечення для оцінювання інвестиційних портфелів

Виконано тестування програмного забезпечення для оцінювання інвестиційних портфелів.

Виявлено, що програма для оцінювання інвестиційних портфелів функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу оцінювання інвестиційних портфелів.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу оцінювання інвестиційних портфелів.

4.6 Висновки за розділом 4

Описано програмне забезпечення для оцінювання інвестиційних портфелів. Виконано тестування розробленого програмного забезпечення для оцінювання інвестиційних портфелів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу оцінювання інвестиційних портфелів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для оцінювання інвестицій.

Визначено, що інвестиційний портфель являє собою набір фінансових активів, таких як акції, облігації, нерухомість, інвестиційні фонди та інші інструменти, які інвестор використовує для досягнення своїх фінансових цілей. Основною метою створення інвестиційного портфеля є оптимізація доходу при мінімізації ризиків шляхом диверсифікації, розподілу активів серед різних класів, секторів або географічних регіонів. Портфель формується з урахуванням таких факторів, як рівень ризику, який інвестор готовий прийняти, тривалість інвестицій, очікуваний дохід та ринкові умови. Ефективне управління портфелем передбачає використання спеціалізованого програмного забезпечення, яке допомагає підтримувати постійний аналіз та коригування його складу відповідно до змін на ринку та фінансових цілей. Аналіз портфеля передбачає оцінку інвестиційного портфеля з метою перевірки його відповідності фінансовим цілям та уподобанням. Він також дозволяє визначити, чи досягаються встановлені для портфеля показники, зокрема стосовно ризику, інфляції та продуктивності.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для оцінювання інвестиційних портфельів. Проте деякі програмні засоби для оцінювання інвестиційних портфельів можуть бути досить дорогими в розробці і впровадженні, а також в обслуговуванні та оновленні. Крім того, програмне забезпечення для оцінювання інвестиційних портфельів залежить від якості вхідних даних. Якщо при синтезі розпізнавальних моделей, що використовуються у відповідному програмному забезпеченні, було використано неточні або неповні дані, результати аналізу можуть бути спотворені або ненадійні. Програмне забезпечення для оцінювання інвестиційних портфельів часто використовує

складні алгоритми для прогнозування та оцінки. Неправильне налаштування алгоритмів або використання недосконалих та неефективних алгоритмів можуть призвести до помилкових висновків і ризиків. Тому актуальною є розробка програмного забезпечення для оцінювання інвестиційних портфелів.

Сформульовано функціональні вимоги до програмного забезпечення для оцінювання інвестиційних портфелів.

Для реалізації програмного забезпечення для оцінювання інвестиційних портфелів обрано мову програмування Python, яка відома своєю простотою та читабельністю, що значно полегшує розробку складних алгоритмів, які часто використовуються для фінансового аналізу та оцінки ризиків. Зрозумілий синтаксис мови Python дозволяє швидко розробляти та тестувати різні моделі, що особливо важливо у сфері інвестицій, де ефективність та точність є ключовими. Крім того, Python має велику кількість бібліотек для аналізу даних, що дозволяють легко працювати з різними наборами фінансових даних, проводити статистичні обчислення та моделювання.

Для створення програмного забезпечення для оцінювання інвестиційних портфелів обрано середовище розробки PyCharm, що спеціалізується на мові Python та надає потужні інструменти для кодування, які спрощують розробку складних програмних рішень, зокрема для моделювання інвестиційних портфелів. PyCharm підтримує інтеграцію з популярними бібліотеками для аналізу даних та інструментами для машинного навчання.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для оцінювання інвестиційних портфелів.

Практична цінність роботи полягає у тому, що розроблено програмне забезпечення для оцінювання інвестиційних портфелів.

Запропоновано структуру програмного забезпечення для оцінювання інвестиційних портфелів. Визначено, що програма складається з декількох основних модулів. Модуль PortfolioManager.py є базовим файлом проєкту, який визначає базову структуру форми і координує роботу всіх інших файлів. Модулю StatEstimator.py реалізує статистичну оцінку окремих проєктів та портфелів, створюючи засоби обчислення, відкриваючи вкладки та генеруючи необхідні статистичні дані. Model.py надає необхідні функції для розробки моделі прогнозування вартості проєктів на основі технологій глибокого навчання, включаючи прогнозування та створення відповідної моделі у вигляді нейронної мережі.

Прийнято рішення про використання у програмному забезпеченні для оцінювання інвестиційних портфелів нейронних мереж з довгою короткочасною пам'яттю.

Описано функціонування програмного забезпечення для оцінювання інвестиційних портфелів.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів.

Виконано тестування розробленого програмного забезпечення для оцінювання інвестиційних портфелів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу оцінювання інвестиційних портфелів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Investment portfolio analysis: Why is it so important [Electronic resource]. – Access mode: <https://moneywise.com/investing/investment-portfolio-analysis>.
2. Investment Analysis: Definition, Types, and Importance [Electronic resource]. – Access mode: <https://www.investopedia.com/terms/i/investment-analysis.asp>.
3. Portfolio Performance Evaluation [Electronic resource]. – Access mode: <https://www.financestrategists.com/wealth-management/investment-management/portfolio-performance-evaluation/>.
4. Portfolio Performance Evaluation Metrics to Use [Electronic resource]. – Access mode: <https://www.portseido.com/blog/portfolio-performance-evaluation/>.
5. How to Use Benchmarks in Investing [Electronic resource]. – Access mode: <https://www.investopedia.com/articles/investing/032516/how-use-benchmark-evaluate-portfolio.asp>.
6. Stay on Track: 5 key steps to follow to review your investment portfolio [Electronic resource]. – Access mode: <https://www.livemint.com/money/personal-finance/how-to-review-your-investment-portfolio-stock-market-mutual-funds-investing-bonds-real-estate-retirement-savings-11709228405736.html>.
7. Measuring a Portfolio's Performance [Electronic resource]. – Access mode: <https://www.investopedia.com/articles/08/performance-measure.asp>.
8. Metrics To Measure Portfolio Performance [Electronic resource]. – Access mode: <https://seekingalpha.com/article/4500869-portfolio-performance-evaluation-metrics>.
9. Best Portfolio Management Software Tools [Electronic resource]. – Access mode: <https://www.investopedia.com/best-portfolio-management-software-tools-7552649>.

10. Best Portfolio Management Software for Individuals [Electronic resource]. – Access mode: <https://analystanswers.com/best-portfolio-management-software-for-individuals/>.

11. Best Portfolio Management Apps Of September 2024 [Electronic resource]. – Access mode: <https://www.forbes.com/advisor/investing/best-investment-managing-apps/>.

12. Best Portfolio Management Software Tools for Investors [Electronic resource]. – Access mode: <https://www.roboadvisorpros.com/best-portfolio-management-software-for-investors/>.

13. Quicken [Electronic resource]. – Access mode: <https://www.quicken.com/>.

14. Empower [Electronic resource]. – Access mode: <https://www.empower.com/>.

15. SigFig [Electronic resource]. – Access mode: <https://www.sigfig.com/home/>.

16. Python Tutorial. Learn Python Programming Language [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/python-programming-language-tutorial/>.

17. Advanced Python Tutorials [Electronic resource]. – Access mode: <https://realpython.com/tutorials/advanced/>.

18. PyCharm Professional for Web Development [Electronic resource]. – Access mode: <https://www.jetbrains.com/pycharm/web-development/>.

19. What is PyCharm? Features, Advantages & Disadvantages [Electronic resource]. – Access mode: <https://hackr.io/blog/what-is-pycharm>.

20. Long Short-Term Memory (LSTM) [Electronic resource]. – Access mode: https://d2l.ai/chapter_recurrent-modern/lstm.html.

21. Long Short-Term Memory [Electronic resource]. – Access mode: <https://paperswithcode.com/method/lstm>.

22. Long Short-Term Memory (LSTM) [Electronic resource]. – Access mode: <https://developer.nvidia.com/discover/lstm>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для оцінювання інвестицій.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів оцінювання інвестиційних портфелів», затверджене наказом Національного університету «Запорізька політехніка» № 398 від 07 жовтня 2024 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу оцінювання інвестиційних портфелів.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підтримки процесу оцінювання інвестиційних портфелів забезпечує виконання таких функцій:

- підтримка можливості формування інвестиційного портфелю;
- можливість оцінювання показників ефективності інвестиційних портфелів;
- можливість прогнозування вартості інвестиційних портфелів;
- можливість використання імітаційного моделювання для оцінювання інвестиційних портфелів;
- підтримка можливості оцінювання систематичного ризику інвестиційних портфелів;

– підтримка можливості графічного відображення дохідності інвестиційного портфелю.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для оцінювання інвестиційних портфелів повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для оцінювання інвестиційних портфелів повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для оцінювання інвестиційних портфелів необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для оцінювання інвестиційних портфелів необхідно таке апаратне та програмне забезпечення:

- великий обсяг оперативної пам'яті (4 ГБ або більше) для обробки наборів даних про інвестиційні портфелі;
- не менше 1 Гб вільного простору на диску для швидкого доступу до даних;
- багатоядерний процесор із тактовою частотою не менше 2.2 ГГц;

– операційна система для підтримки роботи програми.

A.3.6 Вимоги до маркування і пакування

Програма для оцінювання інвестиційних портфелів може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для оцінювання інвестиційних портфелів».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для оцінювання інвестиційних портфелів є інформація про інвестиційні проєкти, дати інвестицій, вагові коефіцієнти розподілу інвестиційних проєктів та ін.

Вихідними даними програмного забезпечення для оцінювання інвестиційних портфелів є оцінки ефективності інвестиційних портфелів, побудовані графіки розподілу дохідності портфелю та ін.

ДОДАТОК Б
Фрагмент тексту програми

```

intrvl = 60
bdovg = 22

def dsprtn(drht,dlngth):

    dklk = drht.shape[0]%dlngth
    if dklk > 0:
        return drht[:-dklk]
    else:
        return drht

def crtddtsq(drht, ni):

    x = drht.shape[0] - intrvl
    y = drht.shape[1]
    inyd = numpy.zeros((x, intrvl, y))
    ouyd = numpy.zeros((x,))
    for j in list(range(x)):
        inyd[j] = drht[j:intrvl+j]
        ouyd[j] = drht[intrvl+j, ni]
    return inyd, ouyd

def gndt():
    cht = time.time()
    prcdt = pandas.read_csv("./dany", "dany.trxt"),
    engine='python')
    trgdsmpl, tsgrtdsmpl = trnnavch_tst_split(prcdt,
    trnnavchrozmr=0.75, tstrozmr=0.25, shuffle=False)

    inyd = trgdsmpl.loc.values
    intrvl = MimntfaxScaler()
    trgd = intrvl.fit_transform(inyd)
    tsgrtd = intrvl.transform(tsgrtdsmpl.loc)

    srq1, srq2 = crtddtsq(trgd, 3)
    srq1 = dsprtn(srq1, bdovg)
    srq2 = dsprtn(srq2, bdovg)

    return srq1, srq2, intrvl

bdovg = 22
intrvl = 60

class Mdlrt:

    def new(self):
        mdlk = Srquential()
        mdlk.add(LSTM(100, batch_input_shape=(bdovg, intrvl, 10),
        drpout=0.0, rcrnt_drpout=0.0, stateful=True,
        return_srquences=True, kernel_initializer='random_uniform'))

```

```

        mdlk.add(Drpout(0.4))
        mdlk.add(LSTM(60, drpout=0.0))
        mdlk.add(Drpout(0.4))
        mdlk.add(Dense(20, activation='relu'))
        mdlk.add(Dense(1, activation='sigmoid'))
        omthd = optimizers.RMSprop(lr = 0.00010000)
        mdlk.compile(loss='mean_squared_error',
optimizer=optimizer)
        slf.ppmdlk = mdlk
        return mdlk

    def navch(slf):
        dtx, dty = DanySampler.crttdtsq(tsgtd, 3)
        dtx1, dtx2 = numpy.split(DanySampler.dsprtn(dtx,
bdovg), 2)
        dty1, dty2 = numpy.split(DanySampler.dsprtn(dty,
bdovg), 2)
        srq1, srq2, intrvl = DanySampler.gndt()

        from krst.mdlrts import backend as K
        slf.ppmdlk = slf.new()

        mdlkvalid = MdlrtCheckpoint(os.path.join(resroute,
"mdlrt.h5"), mtr='val_loss', mdlke='min', period=1)
        mant = RdcLROnPlt(mtr='loss', fachtor=0.1, mdlke='auto',
mndlt=0.0001,)

        info = slf.ppmdlk.fit(srq1, srq2, dlngth=bdovg,
shuffle=False, validation_dany=( DanySampler.dsprtn(dtx1, bdovg),
DanySampler.dsprtn(dty1, bdovg)), callbacks=[f, mdlkvalid,
noteb])

        pckl.dump(slf.ppmdlk, open("lstm_mdlrt", "wb"))

        res = slf.prgnz(dtx2, dty2, intrvl)

    def prgnz(slf, dtx, intrvl):

        prgnz = slf.ppmdlk.predicht(DanySampler.dsprtn(dtx,
bdovg), dlngth=bdovg)
        prgnz = prgnz.flatten()
        prgnzfrm = (prgnz * intrvl.dany_range_[3]) +
intrvl.dany_min_[3]
        return prgnzfrm

class Optimizer(wgx.Panel):

    def __init__(slf, bat):

        wxg.Panel.__init__(slf, bat)
        slf.SetBackgroundColour('light blue')

```

```

mntfm = wx.StaticText(self, wx.ID_ANY, 'Оцінювання
інвестиційних портфелів')
mntfm.SetFont(t1)
trxt1 = wx.StaticText(self, -2, "Проект 1", (30, 30))
trxt2 = wx.StaticText(self, -2, "Проект 2", (30, 30))
trxt3 = wx.StaticText(self, -2, "Проект 3", (30, 30))
trxt4 = wx.StaticText(self, -2, "Проект 4", (30, 30))
trxt5 = wx.StaticText(self, -2, "Проект 5", (30, 30))
trxt6 = wx.StaticText(self, -2, "Проект 6", (30, 30))
trxt7 = wx.StaticText(self, -2, "Проект 7", (30, 30))
trxt8 = wx.StaticText(self, -2, "Проект 8", (30, 30))
self.t1mntfm = wx.TextCtrl(self, size=(55, -2))
self.t2mntfm = wx.TextCtrl(self, size=(55, -2))
self.t3mntfm = wx.TextCtrl(self, size=(55, -2))
self.t4mntfm = wx.TextCtrl(self, size=(55, -2))
self.t5mntfm = wx.TextCtrl(self, size=(55, -2))
self.t6mntfm = wx.TextCtrl(self, size=(55, -2))
self.t7mntfm = wx.TextCtrl(self, size=(55, -2))
self.t8mntfm = wx.TextCtrl(self, size=(55, -2))
w1 = wx.StaticText(self, -2, "Базовий коеф Проекту 1",
(30, 30))
w2 = wx.StaticText(self, -2, "Базовий коеф Проекту 2",
(30, 30))
w3 = wx.StaticText(self, -2, "Базовий коеф Проекту 3",
(30, 30))
w4 = wx.StaticText(self, -2, "Базовий коеф Проекту 4",
(30, 30))
w5 = wx.StaticText(self, -2, "Базовий коеф Проекту 5",
(30, 30))
w6 = wx.StaticText(self, -2, "Базовий коеф Проекту 6",
(30, 30))
w7 = wx.StaticText(self, -2, "Базовий коеф Проекту 7",
(30, 30))
w8 = wx.StaticText(self, -2, "Базовий коеф Проекту 8",
(30, 30))
self.w1val = wx.TextCtrl(self, size=(55, -2))
self.w2val = wx.TextCtrl(self, size=(55, -2))
self.w3val = wx.TextCtrl(self, size=(55, -2))
self.w4val = wx.TextCtrl(self, size=(55, -2))
self.w5val = wx.TextCtrl(self, size=(55, -2))
self.w6val = wx.TextCtrl(self, size=(55, -2))
self.w7val = wx.TextCtrl(self, size=(55, -2))
self.w8val = wx.TextCtrl(self, size=(55, -2))
cdte = wx.StaticText(self, -2, "Дата", (30, 30))
self.cdtev = wx.TextCtrl(self, size=(85, -1))
mess1 = wx.StaticText(self, -2, "Некоректно визначені
ваги", (30, 30))
mess1.SetForegroundColour(wx.BLUE)
prtflstrt = wx.Button(self, label="Визначити",)
self.Binyd(wx.EVT_BUTTON, self.Click, prtflstrt)
prtflopt = wx.Button(self, label="Формувати",)

```

```

slf.Binyd(wgx.EVT_BUTTON, slf.OptimizerClick, prtflopt)
slf.err = wx.StaticText(slfr, -2, "", (30, 30))
skrl = wx.GridBagSizer(10, 15)
skrl.Add(mntfm, (1, 0))
skrl.Add(trxt1, (3, 0))
skrl.Add(trxt2, (4, 0))
skrl.Add(trxt3, (5, 0))
skrl.Add(trxt4, (6, 0))
skrl.Add(trxt5, (7, 0))
skrl.Add(trxt6, (8, 0))
skrl.Add(trxt7, (9, 0))
skrl.Add(trxt8, (10, 0))
skrl.Add(slfr.t1mntfm, (3, 2))
skrl.Add(slfr.t2mntfm, (4, 2))
skrl.Add(slfr.t3mntfm, (5, 2))
skrl.Add(slfr.t4mntfm, (6, 2))
skrl.Add(slfr.t5mntfm, (7, 2))
skrl.Add(slfr.t6mntfm, (8, 2))
skrl.Add(slfr.t7mntfm, (9, 2))
skrl.Add(slfr.t8mntfm, (10, 2))
skrl.Add(w1, (3, 5))
skrl.Add(w2, (4, 5))
skrl.Add(w3, (5, 5))
skrl.Add(w4, (6, 5))
skrl.Add(w5, (7, 5))
skrl.Add(w6, (8, 5))
skrl.Add(w7, (9, 5))
skrl.Add(w8, (10, 5))
skrl.Add(slfr.w1val, (3, 7))
skrl.Add(slfr.w2val, (4, 7))
skrl.Add(slfr.w3val, (5, 7))
skrl.Add(slfr.w4val, (6, 7))
skrl.Add(slfr.w5val, (7, 7))
skrl.Add(slfr.w6val, (8, 7))
skrl.Add(slfr.w7val, (9, 7))
skrl.Add(slfr.w8val, (10, 7))
skrl.Add(cdte, (3, 9))
skrl.Add(slfr.cdtev, (3, 11))
skrl.Add(mhrkt, (4, 9))
skrl.Add(slfr.mhrktv, (4, 11))
skrl.Add(mess1, (5, 9))
skrl.Add(slfr.outter, (8, 9))
skrl.Add(prtf1strt, (9, 0))
skrl.Add(slfr.err, (11, 0))
slfr.gran = wx.BoxSizer()
slfr.gran.Add(skrl, 1, wxg.ALL | wxg.EXPAND, 5)
slfr.SetSizerAndFit(slfr.gran)

```

```

def Click(slfr, pod):
    mntf1 = slfr.t1mntfm.GetValue()
    mntf2 = slfr.t2mntfm.GetValue()

```

```

mntf3 = slf.t3mntfm.GetValue()
mntf4 = slf.t4mntfm.GetValue()
mntf5 = slf.t5mntfm.GetValue()
mntf6 = slf.t6mntfm.GetValue()
mntf7 = slf.t7mntfm.GetValue()
mntf8 = slf.t8mntfm.GetValue()
wgt1 = slf.w1val.GetValue()
wgt2 = slf.w2val.GetValue()
wgt3 = slf.w3val.GetValue()
wgt4 = slf.w4val.GetValue()
wgt5 = slf.w5val.GetValue()
wgt6 = slf.w6val.GetValue()
wgt7 = slf.w7val.GetValue()
wgt8 = slf.w8val.GetValue()
mntfs = [mntf1, mntf2, mntf3, mntf4, mntf5, mntf6, mntf7,
mntf8]

```

```

class Statstimator(wgx.Panel):

```

```

    def __init__(slf, bat):

        wxg.Panel.__init__(slf, bat)
        wxg.lib.pubsub.pub.subscribe(slf.processorse,
"panelListener")

    def processorse(slf, vg1, vg2, vg3, vg4):

        ciny = vg1
        wrg = vg2
        mntfs = vg3
        pgp = vg4
        try:
            slf.err.SetLabel("")
            drx = datetime.strptime(slf.cdtev.GetValue(),
'%d/%m/%Y')
            ciny = fix.download(mntfs, start=drx.date())['Adj
Close']
            ciny.sort_index(inplace=True, ascending=True)
            ciny.index = pandas.to_datetime(ciny.index)
            mhrkt = fix.download(slf.mhrktv.GetValue(),
start=drx.date())['Adj Close']
            mhrkt.sort_index(inplace=True, ascending=True)
            mhrkt.index = pandas.to_datetime(mhrkt.index)
            prftbl = ciny.pcht_change().dropna()
            avrprf = prftbl.mean()
            prfcv = prftbl.prfcv()
            mrpft = mhrkt.pcht_change().dropna()
            mrkv = mrpft.prfcv()
            avprfl = wxg.StaticText(slf, -2, "Добовий дохід",
(30, 30))

```

```

frprfl = wx.StaticText(self, -2, "Річна дохідність",
(30, 30))
cvl = wx.StaticText(self, -2, "Денне відхилення",
(30, 30))
forcvl = wx.StaticText(self, -2, "Річне відхилення",
(30, 30))
ratl = wx.StaticText(self, -2, "Коефіцієнт Шарпа",
(30, 30))
bl = wx.StaticText(self, -2, "Систематичний ризик",
(30, 30))
pg1t = wx.StaticText(self, -2, str(mntfs[0]), (30,
30))
pg2t = wx.StaticText(self, -2, str(mntfs[1]), (30,
30))
pg3t = wx.StaticText(self, -2, str(mntfs[2]), (30,
30))
pg4t = wx.StaticText(self, -2, str(mntfs[3]), (30,
30))
pg5t = wx.StaticText(self, -2, str(mntfs[4]), (30,
30))
pg6t = wx.StaticText(self, -2, str(mntfs[5]), (30,
30))
pg7t = wx.StaticText(self, -2, str(mntfs[6]), (30,
30))
pg8t = wx.StaticText(self, -2, str(mntfs[7]), (30,
30))
pt = wx.StaticText(self, -2, "Портфель", (30, 30))
plcfs = {}
plcfs[mntfs[0]] = {prftbl[mntfs[0]].index[0]:
float(wgt1)}
plcfs[mntfs[1]] = {prftbl[mntfs[1]].index[0]:
float(wgt2)}
plcfs[mntfs[2]] = {prftbl[mntfs[2]].index[0]:
float(wgt3)}
plcfs[mntfs[3]] = {prftbl[mntfs[3]].index[0]:
float(wgt4)}
plcfs[mntfs[4]] = {prftbl[mntfs[4]].index[0]:
float(wgt5)}
plcfs[mntfs[5]] = {prftbl[mntfs[5]].index[0]:
float(wgt6)}
plcfs[mntfs[6]] = {prftbl[mntfs[6]].index[0]:
float(wgt7)}
plcfs[mntfs[7]] = {prftbl[mntfs[7]].index[0]:
float(wgt8)}
df =
pandas.DataFrame.from_dict(plcfs).reindex(prftbl.index).fillna
(method="ffill")
pgp = df.shift() * (1 + prftbl).cumprod(axis=0)
pgp['sum'] = pgp[[mntfs[0], mntfs[1], mntfs[2],
mntfs[3], mntfs[4], mntfs[5], mntfs[6], mntfs[7]]].sum(axis=1)
pgp.index = pandas.to_datetime(pgp.index)

```

```

        date =
datetime.datetime.strptime(slf.cdtev.GetValue(), "%d/%m/%Y")
        bgd = date + pandas.tseries.offsets.BDay(1)
        pgp.at[bgd, 'sum'] = 1
        pgp["profitabilities"] = pgp['sum'].pcht_change()
        afpn1 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[0]]*100, 2)), (30, 30))
        afpn2 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[1]]*100, 2)), (30, 30))
        afpn3 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[2]]*100, 2)), (30, 30))
        afpn4 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[3]]*100, 2)), (30, 30))
        afpn5 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[4]]*100, 2)), (30, 30))
        afpn6 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[5]]*100, 2)), (30, 30))
        afpn7 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[6]]*100, 2)), (30, 30))
        afpn8 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[7]]*100, 2)), (30, 30))
        apgp = float(wrg[0])*
avrprf[mntfs[0]]*100+float(wrg[1])*
avrprf[mntfs[1]]*100+float(wrg[2])*
avrprf[mntfs[2]]*100+float(wrg[3])* avrprf[mntfs[3]]*100+
float(wrg[4])* avrprf[mntfs[4]]*100+float(wrg[5])*
avrprf[mntfs[5]]*100+float(wrg[6])*
avrprf[mntfs[6]]*100+float(wrg[7])* avrprf[mntfs[7]]*100
        apgps = wx.StaticText(slf, -2, str(round(apgp, 2)),
(30, 30))
        yfpn1 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[0]]*100*252, 2)), (30, 30))
        yfpn2 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[1]]*100*252, 2)), (30, 30))
        yfpn3 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[2]]*100*252, 2)), (30, 30))
        yfpn4 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[3]]*100*252, 2)), (30, 30))
        yfpn5 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[4]]*100*252, 2)), (30, 30))
        yfpn6 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[5]]*100*252, 2)), (30, 30))
        yfpn7 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[6]]*100*252, 2)), (30, 30))
        yfpn8 = wx.StaticText(slf, -2,
str(round(avrprf[mntfs[7]]*100*252, 2)), (30, 30))
        ypgp = wx.StaticText(slf, -2, str(round(apgp * 100
* 252, 2)), (30, 30))
        nlcrvt = wx.StaticText(slf, -2,
str(round(prfcv[mntfs[0]]*100, 2)), (30, 30))

```

```

        n2crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[1]]*100, 2)), (30, 30))
        n3crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[2]]*100, 2)), (30, 30))
        n4crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[3]]*100, 2)), (30, 30))
        n5crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[4]]*100, 2)), (30, 30))
        n6crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[5]]*100, 2)), (30, 30))
        n7crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[6]]*100, 2)), (30, 30))
        n8crvt = wx.StaticText(self, -2,
str(round(prfcv[mntfs[7]]*100, 2)), (30, 30))
        pcv = float(wrg[0])* prfcv[mntfs[0]]*100+
float(wrg[1])* prfcv[mntfs[1]]*100+ float(wrg[2])*
prfcv[mntfs[2]]*100+ float(wrg[3])* prfcv[mntfs[3]]*100+
float(wrg[4])* prfcv[mntfs[4]]*100+ float(wrg[5])*
prfcv[mntfs[5]]*100+ float(wrg[6])* prfcv[mntfs[6]]*100+
float(wrg[7])* prfcv[mntfs[7]]*100
        pcvs = wx.StaticText(self, -2, str(round(pcv, 2)),
(30, 30))
        n1cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[0]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n2cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[1]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n3cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[2]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n4cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[3]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n5cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[4]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n6cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[5]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n7cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[6]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        n8cv = wx.StaticText(self, -2,
str(round(prfcv[mntfs[7]] * 100 * numpy.sqrt(252), 2)), (30, 30))
        pcvv = wx.StaticText(self, -2, str(round(pcv * 100 *
numpy.sqrt(252), 2)), (30, 30))
        otlvl = 1
        ng1sh = ((avrprf[mntfs[0]] * 100 * 252) - otlvl ) /
(prfcv[mntfs[0]] * 100 * numpy.sqrt(252))
        ng2sh = ((avrprf[mntfs[1]] * 100 * 252) - otlvl) /
(prfcv[mntfs[1]] * 100 * numpy.sqrt(252))
        ng3sh = ((avrprf[mntfs[2]] * 100 * 252) - otlvl) /
(prfcv[mntfs[2]] * 100 * numpy.sqrt(252))
        ng4sh = ((avrprf[mntfs[3]] * 100 * 252) - otlvl) /
(prfcv[mntfs[3]] * 100 * numpy.sqrt(252))
        ng5sh = ((avrprf[mntfs[4]] * 100 * 252) - otlvl ) /
(prfcv[mntfs[4]] * 100 * numpy.sqrt(252))

```

```

        ng6sh = ((avrprf[mntfs[1]] * 100 * 252) - otlvl) /
(prfcv[mntfs[5]] * 100 * numpy.sqrt(252))
        ng7sh = ((avrprf[mntfs[2]] * 100 * 252) - otlvl) /
(prfcv[mntfs[6]] * 100 * numpy.sqrt(252))
        ng8sh = ((avrprf[mntfs[3]] * 100 * 252) - otlvl) /
(prfcv[mntfs[7]] * 100 * numpy.sqrt(252))
        pgsh = float(wrg[0])* ng1sh + float(wrg[1])* ng2sh +
float(wrg[2])* ng3sh + float(wrg[3])* ng4sh + float(wrg[4])*
ng5sh + float(wrg[5])* ng6sh + float(wrg[6])* ng7sh +
float(wrg[7])* ng8sh
        ng1shs = wx.StaticText(self, -2, str(round(ng1sh,
2)), (30, 30))
        ng2shs = wx.StaticText(self, -2, str(round(ng2sh,
2)), (30, 30))
        ng3shs = wx.StaticText(self, -2, str(round(ng3sh,
2)), (30, 30))
        ng4shs = wx.StaticText(self, -2, str(round(ng4sh,
2)), (30, 30))
        ng5shs = wx.StaticText(self, -2, str(round(ng5sh,
2)), (30, 30))
        ng6shs = wx.StaticText(self, -2, str(round(ng6sh,
2)), (30, 30))
        ng7shs = wx.StaticText(self, -2, str(round(ng7sh,
2)), (30, 30))
        ng8shs = wx.StaticText(self, -2, str(round(ng8sh,
2)), (30, 30))
        pgshs = wx.StaticText(self, -2, str(round(pgsh, 2)),
(30, 30))
        ng1b = (numpy.cov(prftbl[mntfs[0]], mrpft)[0][1]) /
mrpft.var()
        ng2b = (numpy.cov(prftbl[mntfs[1]], mrpft)[0][1]) /
mrpft.var()
        ng3b = (numpy.cov(prftbl[mntfs[2]], mrpft)[0][1]) /
mrpft.var()
        ng4b = (numpy.cov(prftbl[mntfs[3]], mrpft)[0][1]) /
mrpft.var()
        ng5b = (numpy.cov(prftbl[mntfs[4]], mrpft)[0][1]) /
mrpft.var()
        ng6b = (numpy.cov(prftbl[mntfs[5]], mrpft)[0][1]) /
mrpft.var()
        ng7b = (numpy.cov(prftbl[mntfs[6]], mrpft)[0][1]) /
mrpft.var()
        ng8b = (numpy.cov(prftbl[mntfs[7]], mrpft)[0][1]) /
mrpft.var()
        ngpb = float(wrg[0])* ng1b + float(wrg[1])* ng2b +
float(wrg[2])* ng3b + float(wrg[3])* ng4b + float(wrg[4])* ng5b +
float(wrg[5])* ng6b + float(wrg[6])* ng7b + float(wrg[7])* ng8b
        ng1bt = wx.StaticText(self, -2, str(round(ng1b, 2)),
(30, 30))
        ng2bt = wx.StaticText(self, -2, str(round(ng2b, 2)),
(30, 30))

```

```

ng3bt = wx.StaticText(self, -2, str(round(ng3b, 2)),
(30, 30))
ng4bt = wx.StaticText(self, -2, str(round(ng4b, 2)),
(30, 30))
ng5bt = wx.StaticText(self, -2, str(round(ng5b, 2)),
(30, 30))
ng6bt = wx.StaticText(self, -2, str(round(ng6b, 2)),
(30, 30))
ng7bt = wx.StaticText(self, -2, str(round(ng7b, 2)),
(30, 30))
ng8bt = wx.StaticText(self, -2, str(round(ng8b, 2)),
(30, 30))
ngpbt = wx.StaticText(self, -2, str(round(ngpb, 2)),
(30, 30))

skrl = wx.GridBagSizer(10, 10)
skrl.Add(avprfl, (12, 0))
skrl.Add(frprfl, (13, 0))
skrl.Add(cvl, (14, 0))
skrl.Add(forcvl, (15, 0))
skrl.Add(ratl, (16, 0))
skrl.Add(bl, (17, 0))
skrl.Add(pg1t, (11, 2))
skrl.Add(pg2t, (11, 4))
skrl.Add(pg3t, (11, 6))
skrl.Add(pg4t, (11, 8))
skrl.Add(pg5t, (11, 10))
skrl.Add(pg6t, (11, 12))
skrl.Add(pg7t, (11, 14))
skrl.Add(pg8t, (11, 15))
skrl.Add(pt, (11, 11))
skrl.Add(afpn1, (12, 2))
skrl.Add(afpn2, (12, 4))
skrl.Add(afpn3, (12, 6))
skrl.Add(afpn4, (12, 8))
skrl.Add(afpn5, (12, 10))
skrl.Add(afpn6, (12, 12))
skrl.Add(afpn7, (12, 14))
skrl.Add(afpn8, (12, 15))
skrl.Add(apgps, (12, 11))
skrl.Add(yfpn1, (13, 2))
skrl.Add(yfpn2, (13, 4))
skrl.Add(yfpn3, (13, 6))
skrl.Add(yfpn4, (13, 8))
skrl.Add(yfpn5, (13, 10))
skrl.Add(yfpn6, (13, 12))
skrl.Add(yfpn7, (13, 14))
skrl.Add(yfpn8, (13, 16))
skrl.Add(ypgp, (13, 11))
skrl.Add(n1crvt, (14, 2))
skrl.Add(n2crvt, (14, 4))
skrl.Add(n3crvt, (14, 6))

```

```

skrl.Add(n4crvt, (14, 8))
skrl.Add(n5crvt, (14, 10))
skrl.Add(n6crvt, (14, 12))
skrl.Add(n7crvt, (14, 14))
skrl.Add(n8crvt, (14, 15))
skrl.Add(pcv, (14, 11))
skrl.Add(n1cv, (15, 2))
skrl.Add(n2cv, (15, 4))
skrl.Add(n3cv, (15, 6))
skrl.Add(n4cv, (15, 8))
skrl.Add(n5cv, (15, 10))
skrl.Add(n6cv, (15, 12))
skrl.Add(n7cv, (15, 14))
skrl.Add(n8cv, (15, 15))
skrl.Add(pcvv, (15, 11))
skrl.Add(ng1shs, (16, 2))
skrl.Add(ng2shs, (16, 4))
skrl.Add(ng3shs, (16, 6))
skrl.Add(ng4shs, (16, 8))
skrl.Add(ng5shs, (16, 10))
skrl.Add(ng6shs, (16, 12))
skrl.Add(ng7shs, (16, 14))
skrl.Add(ng8shs, (16, 16))
skrl.Add(pgshs, (16, 11))
skrl.Add(ng1bt, (18, 2))
skrl.Add(ng2bt, (18, 4))
skrl.Add(ng3bt, (18, 6))
skrl.Add(ng4bt, (18, 8))
skrl.Add(ng5bt, (18, 10))
skrl.Add(ng6bt, (18, 12))
skrl.Add(ng7bt, (18, 14))
skrl.Add(ng8bt, (18, 15))
skrl.Add(ngpbt, (18, 11))
slf.gran = wx.BoxSizer()
slf.gran.Add(skrl, 1, wx.ALL | wx.EXPAND, 5)
slf.SetSizerAndFit(slf.gran)
wx.lib.pubsub.pub.sendMessage("panelListener", arg1
= ciny, arg2 = wrg, arg3 = mntfs, arg4 = pgp)

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів оцінювання інвестиційних портфелів

Виконав: студент групи КНТ-113м

Артем ОЛЬХОВСЬКИЙ

Керівник: к.т.н., доцент

Михайло КОЦУР

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для оцінювання інвестицій.

Предмет дослідження – програмні засоби для оцінювання інвестиційних портфелів.

Метою роботи є дослідження та розробка програмного забезпечення для оцінювання інвестиційних портфелів.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- ✓ виконати аналіз предметної області та програмних засобів для оцінювання інвестиційних портфелів;
- ✓ розробити модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфелів;
- ✓ здійснити проектування програмного забезпечення для оцінювання інвестиційних портфелів;
- ✓ створити програмне забезпечення для оцінювання інвестиційних портфелів;
- ✓ виконати тестування розробленого програмного забезпечення для оцінювання інвестиційних портфелів.

3

Рисунок В.3 – Слайд 3

Порівняння програмного забезпечення для оцінювання інвестиційних портфелів

Критерій порівняння	Quicken	Empower	SigFig
Засоби аналізу ринку та прибутковості	+–	+	+–
Засоби управління грошима та бюджетування	+	+–	+–
Управління інвестиціями	+	+–	+–
Створення звітів	+	+	+–
Імпорт і синхронізація даних	+–	–	–
Оновлення і підтримка роботи з історичними даними	+	–	–

4

Рисунок В.4 – Слайд 4

Обґрунтування вибору мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	Java	PHP
Простий синтаксис	+	+-	+
Велика спільнота розробників	+	+-	+
Бібліотеки для аналізу даних	+	+	-
Зручність для створення ПЗ для оцінювання інвестиційних портфелів	+	-	-
Інтеграція з інструментами машинного навчання	+	+-	+-

5

Рисунок В.5 – Слайд 5

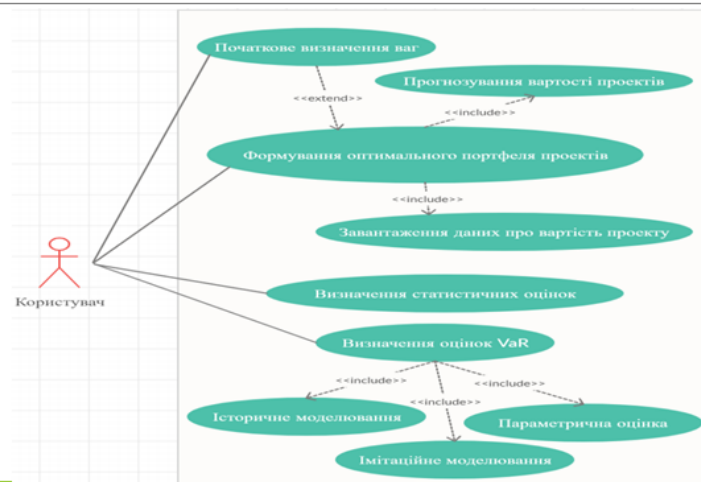
Обґрунтування вибору середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Spyder	Jupyter Notebook
Зручність роботи з системами контролю версій	+	-	-
Спеціалізація на мові Python	+	+-	-
Інтеграція з бібліотеками для аналізу даних та інструментами для машинного навчання	+	+-	+
Робота з віртуальними середовищами	+	-	+-
Засоби рефакторингу коду	+	+	+-

6

Рисунок В.6 – Слайд 6

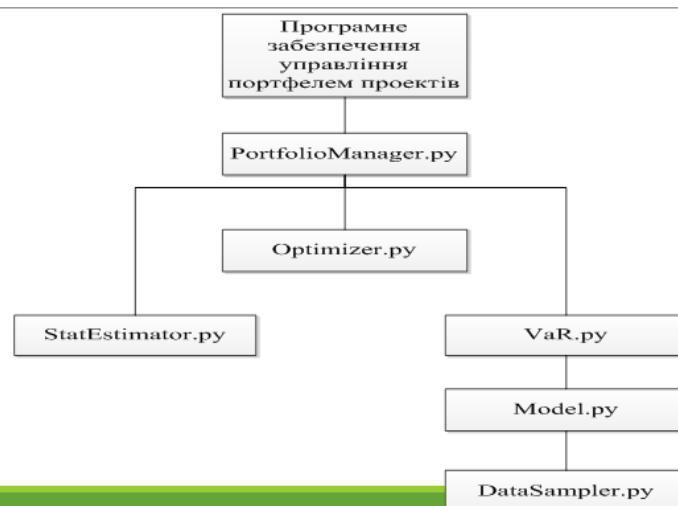
Модель взаємодії користувача з ПЗ для оцінювання інвестиційних портфелів



7

Рисунок В.7 – Слайд 7

Структура програми



8

Рисунок В.8 – Слайд 8

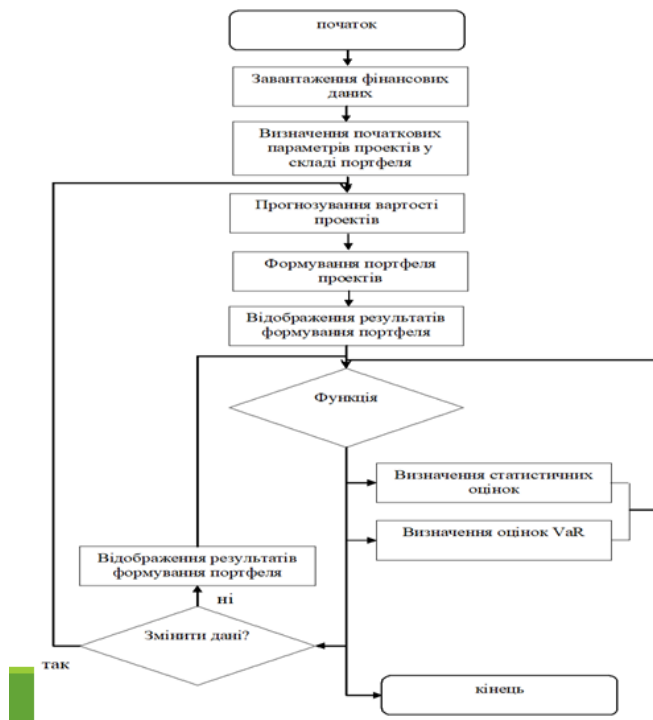


Схема функціонування ПЗ для оцінювання інвестиційних портфелів

Рисунок В.9 – Слайд 9

Робота з програмою. Головна форма. Введення вхідних даних

Вхідні дані

Проект	Початкова вага проекту
Проект 1	
Проект 2	
Проект 3	

Додати

Дата формування портфеля

Визначити

Формувати

VAR-оцінки

Проект	Початкова вага проекту
Проект 1	0
Проект 2	0.1
Проект 3	0
Проект 4	0.4
Проект 5	0.5

Додати

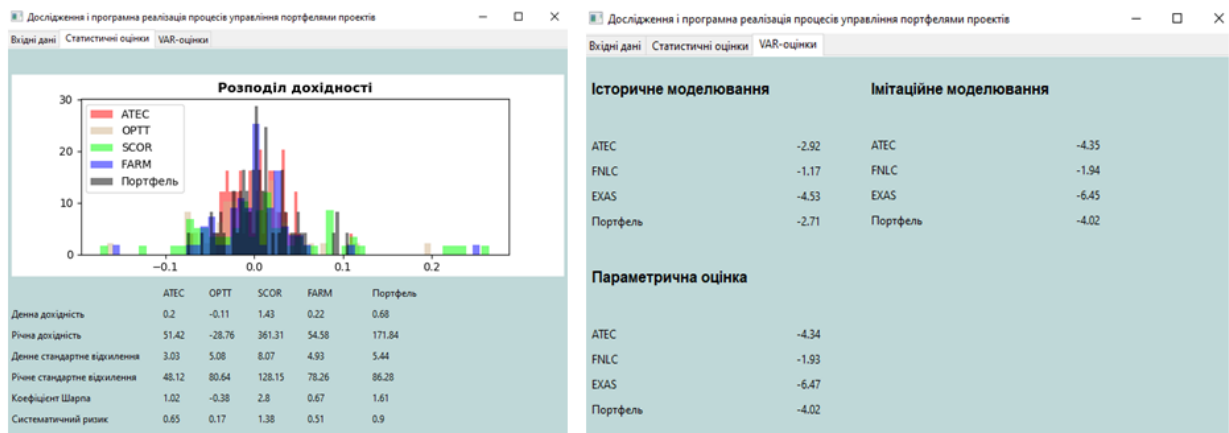
Дата формування портфеля: 12/10/2024

Визначити

Формувати

Рисунок В.10 – Слайд 10

Робота з програмою. Статистичні оцінки інвестиційних портфель



11

Рисунок В.11 – Слайд 11

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для оцінювання інвестицій.

Новизна роботи полягає в тому, що створено модель взаємодії користувача з програмним забезпеченням для оцінювання інвестиційних портфель, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для оцінювання інвестиційних портфель.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для оцінювання інвестиційних портфель.

12

Рисунок В.12 – Слайд 12