

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕСІВ
ЕНЕРГОЗБЕРЕЖЕННЯ НА МЕТАЛУРГІЙНИХ ПІДПРИЄМСТВАХ.
RESEARCH AND SOFTWARE IMPLEMENTATION OF ENERGY SAVING
PROCESSES AT IRON AND STEEL WORKS

Виконав: студент(ка) 2 курсу, групи КНТ-210м
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Ільон Д.Г.

(прізвище та ініціали)

Керівник Дубровін В.І.

(прізвище та ініціали)

Рецензент Касьян М.М.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Ільонка Дмитра Геннадійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація процесів енергозбереження на металургійних підприємствах. Research and Software Implementation of Energy Saving Processes at Iron and Steel Works

керівник проєкту (роботи) Дубровін Валерій Іванович, к.т.н., професор,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “05” листопада 2021 року № 435

2. Строк подання студентом проєкту (роботи) 20 грудня 2021 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. Оптимізація інвестиційного портфелю. 3. Проєктування програмного додатку. 4. Програмна реалізація. 5. Експлуатація, тестування та експериментальне дослідження програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Дубровін В.І., професор		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “06” вересня 2021 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.	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	9-10 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	11-13 тиждень	Додатки
8	Нормоконтроль та рецензування.	14-15 тижні	
9	Захист роботи.	16 тиждень	

Студент(ка)

(підпис)

Льонюк Д.Г.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Дубровін В.І.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
79 с., 1 табл., 50 рис., 2 дод., 10 джерел.

ЕФЕКТИВНА МЕЖА РИЗИКІВ, КЕРУВАННЯ РЕСУРСАМИ,
ОПТИМІЗАЦІЯ ПОРТФЕЛЯ, ОПТИМІЗАЦІЯ ПРОЦЕСІВ
ЕНЕРГОЗБЕРЕЖЕННЯ, КОМП'ЮТЕРНА ПРОГРАМА.

Об'єкт дослідження – програмні засоби моніторингу процесів енергозбереження на металургійних підприємствах.

Предмет дослідження – процеси енергозбереження на металургійних підприємствах.

Мета роботи – створення програмної системи для автоматизації процесів енергозбереження на металургійних підприємствах.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мови програмування C++ та фреймворк QT, персональний комп'ютер з процесором Intel Core i5 під управлінням операційної системи Microsoft Windows 10.

Результати. Створено програмний Windows-застосунок, який дозволяє автоматизувати процеси енергозбереження на металургійних підприємствах завдяки використанню портфельної оптимізації.

Висновки. Виконано огляд процесів енергозбереження на металургійних підприємствах. Проаналізовані методи портфельного інвестування. Розроблено програмний Windows-застосунок, який дозволяє автоматизувати зазначені процеси, завдяки використанню портфельної оптимізації.

Галузь використання – підприємства середнього та високого класу.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 79 pages, 1 tables, 50 figures, 2 appendixes, 10 sources.

EFFICIENT RISK TOLERANCE, RESOURCE MANAGEMENT, PORTFOLIO OPTIMIZATION, ENERGY SAVING OPTIMIZATION, COMPUTER PROGRAM.

Object of study is a software for monitoring of energy saving processes at iron and steel works.

Subject of study are energy saving processes at iron and steel works.

The purpose of the work is to create a software system for the automation energy saving processes at iron and steel works.

Materials, methods, and tools: structured and object-oriented programming, programming languages C++ with QT framework, IBM-compatible personal computer with an Intel Core i5 processor under the control of the Microsoft Windows 10 operating system.

Results. The Windows software application is created which automate energy saving processes at iron and steel works using portfolio optimization methods.

Conclusions. The review of energy saving processes at iron and steel works was did. The portfolio investing methods was analyzed. The Windows software application is created which automate energy saving processes at iron and steel works using portfolio optimization methods was developed.

The field of use is middle and hight class of enterprises.

ЗМІСТ

С.

Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	10
1.1 Ризики та об'єкти інвестування	10
1.2 Портфельні інвестиції	13
1.3 Основні принципи портфельної оптимізації	14
1.4 Особливості та переваги портфельного інвестування	16
1.5 Висновки до розділу	17
2 Оптимізація інвестиційного портфелю	18
2.1 Основні концепції сучасної портфельної теорії	18
2.2 Баланс ризику та прибутковості	19
2.3 Математична реалізація портфельної оптимізації	21
2.4 Методи оптимізації портфелю	23
2.5 Оптимізація інвестиційного портфеля за методом Марковіца	27
2.6 Висновки до розділу	31
3 Проектування програмного додатку	32
3.1 UML-діаграми. Діаграма Use-Case	32
3.2 Розробка діаграми послідовності	34
3.3 Розробка діаграми розгортання	34
3.4 Розробка діаграми діяльності	35
3.5 Висновки до розділу	36
4 Програмна реалізація	37
4.1 Технічні засоби розробки	37
4.2 Вихідні дані до проекту	39
4.3 Розробка інтерфейсу програмного продукту	39
4.4 Висновки до розділу	45
5 Експлуатація, тестування та експериментальне дослідження програми	46
5.1 Експериментальне дослідження програмного продукту	46

5.2 Програмне тестування. Unit-tests 54

5.3 Висновки до розділу 55

Висновки..... 56

Перелік джерел посилання 57

Додаток А Текст програми 58

Додаток Б Слайди презентації..... 74

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
CDMA	– Code Division Multiple Access;
IDE	– Integrated Development Environment;
GCC	– GNU Compiler Collection;
GUI	– Graphical User Interface;
OpenGL	– Open Graphics Library;
UML	– Unified Modeling Language;
ОС	– операційна система;
ПЗ	– програмне забезпечення;
СПТ	– сучасна портфельна теорія.

ВСТУП

Металургійні підприємства є великими споживачами електричної і теплової енергії, тому вирішення завдання енергозбереження неможливо без розроблення комплексної програми й портфеля проєктів за основними напрямками енергозбереження з обов'язковою координацією їх з програмою розвитку основного виробництва [1]. Гострота вирішення цих питань для металургійної галузі України спричинена необхідністю поліпшення економічної стабільності підприємств, підвищення конкурентоспроможності продукції та зменшення залежності від постачальників енергоресурсів [2].

Оптимізація інвестиційного портфеля є одним із поширених, типових і значимих фінансових завдань, яке виникає в практиці ресурсного забезпечення, енергозбереження, інвестування. Його рішення дозволяє знайти найбільш ефективний спосіб вкладення інвестором свого капіталу в акції декількох компаній. Основними принципами формування інвестиційного портфеля є надійність і дохідність активів, їх стабільне зростання і висока ліквідність. Метою оптимізації портфеля цінних паперів є формування такого портфеля цінних паперів, який би відповідав вимогам інвестора, підприємства, як по прибутковості, так і щодо можливого ризику, що досягається шляхом розподілом цінних паперів в портфелі [3].

Актуальність даної теми полягає в дослідженні методів оптимізації портфельних інвестицій та обумовлена тим, що основне завдання портфельного інвестування – поліпшити умови інвестування, додавши сукупності цінних паперів такі інвестиційні характеристики, які недосяжні з позиції окремо взятого цінного паперу, і можливі тільки при їх комбінації [4].

Таким чином, дослідження та програмна реалізація процесів енергозбереження на металургійних підприємствах є актуальною проблемою, вирішення якої допоможе оптимізувати свою роботу підприємствам.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Ризики та об'єкти інвестування

Для промислових об'єктів в Україні в сучасних умовах існують суттєві проблеми, які пов'язані із фінансуванням енергозберігаючих заходів, вибором оптимальних способів вкладення коштів в інвестиційні проєкти з підвищення енергоефективності, відсутністю інформаційних систем та недосконалістю організаційно-економічного механізму забезпечення енергозбереження на промислових підприємствах [1]. Для запобігання фінансових втрат при формуванні сукупності енергозберігаючих заходів (портфелів проєктів) потрібна розробка та вдосконалення методів оцінки ефективності програм енергозбереження, які враховують широкий вибір у використанні джерел інвестицій, призначених для їх реалізації.

Оскільки методи управління портфелями проєктів енергозбереження в основному розглядають методи оптимізації витрат на ці проєкти, то для розв'язання даної задачі доречно розглянути саме методи оптимізації інвестиційних портфелів. Роблячи таке узагальнення, потрібно враховувати специфіку предметної області та використовувати лише ті методи оптимізації, які дозволяють залишатися в заданих предметною областю обмеженнях [3].

Узагальнення до більш широкої теми дозволяє використовувати багато напрацювань та теоретичних матеріалів в більш загальній сфері

Ризик – це діяльність, пов'язана з подоланням невизначеності в ситуації неминучого вибору, в процесі якої є можливість кількісно і якісно оцінити ймовірність досягнення передбачуваного результату, невдачі і відхилення від мети. Різниця між ризиком і невизначеністю відноситься до способу завдання інформації і визначається наявністю (у разі ризику) або відсутністю (при невизначеності) ймовірнісних характеристик неконтрольованих змінних. Якщо існує можливість якісно і кількісно визначити ступінь ймовірності того або іншого варіанту, це і буде ситуація ризику [3].

Таким чином, ситуація ризику (ризикована ситуація) - це різновид невизначеності, коли настання подій ймовірно і може бути визначено. Ризик завжди пов'язаний з вибором певних альтернатив і розрахунком імовірності їхнього результату – в цьому виявляється його суб'єктивна сторона. Разом з тим, величина ризику не тільки суб'єктивна, але й об'єктивна, тому що вона є формою якісно-кількісного вираження реально множиною невизначеності.

Для розуміння природи підприємницького ризику фундаментальне значення має зв'язок ризику і прибутку. Підприємець виявляє готовність йти на ризик в умовах невизначеності, оскільки поряд з ризиком втрат існує можливість додаткових доходів. Хоча ясно, що отримання прибутку підприємцю не гарантовано, винагороду за витрачений їм час, зусилля і здібності можуть виявитися як прибуток, так і збитки. Як видно з рис. 1.1 значення прибутку напряму залежить від ризику. Можна вибрати рішення, що містить менше значення ризику ($r_1 = 0$), але при цьому менше буде і отримується прибуток (Π_1), а при найвищому ризик r_3 прибуток має найбільш високе значення, рівне Π_3 . Іншими словами, для отримання економічного прибутку підприємець повинен усвідомлено піти на прийняття ризикового рішення [2].

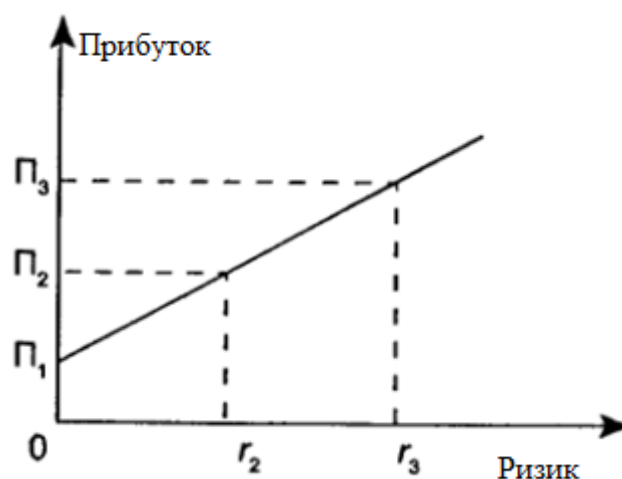


Рисунок 1.1 – Залежність прибутку від ризику

В процесі оцінки невизначеності і кількісного визначення ризику використовують імовірнісні розрахунки. Імовірностне завдання характеризується тим, що ефективність прийнятих рішень залежить не тільки від детермінованих чинників, але і від ймовірностей їх появи, тобто відомий закон розподілу керованих факторів X , що зображено в табл. 1.1, де P_i , є ймовірність появи керованого фактора x_i , а $i = \overline{1, n}$ [5].

Таблиця 1.1 – Відповідність ознак їх імовірнісним значенням

X	X_1	X_2	...	X_n
P	P_1	P_2	...	P_n

Кожній парі (X_i, P_i) відповідає значення функції ефективності $M(x_i, P_i)$. Як показники ефективності можуть виступати: математичне очікування ($M = \sum x_i p_i$), дисперсія $D = \sigma^2 = \sum (x_i - M)^2 \cdot P_i = \overline{x^2} - (M)^2$, коефіцієнт варіації $V = \pm \frac{\sigma}{E} \cdot 100\%$ та інші ймовірні характеристики.

Коефіцієнт варіації – безрозмірна величина. З його допомогою можна порівнювати навіть коливання ознак, виражених у різних одиницях виміру. Коефіцієнт варіації змінюється від 0 до 100%. Чим більше коефіцієнт, тим більше коливання.

Ризиком операції називається число – середньоквадратичне відхилення керованого фактора (наприклад, доходу) X операції, яке позначимо.

Перевагою даного методу оцінки підприємницького ризику є простотаматематичних розрахунків, а явним недоліком – необхідність великої кількості вихідних даних (чим більше масив, тим точніше оцінка ризику).

1.2 Портфельні інвестиції

Портфельні інвестиції – інвестиції у цінні папери та різноманітні фінансові активи за кошти на фондовому ринку. Портфельна оптимізація – процес створення портфелю фінансових активів, для яких ваш портфель має певні значення деяких параметрів. Зазвичай, власник портфелю бажає мінімізувати ризики та максимізувати віддачу.

Під інвестиційним портфелем розуміють декілька інвестиційних об'єктів – активів. Під активом розуміють майже будь-які інвестиції, такі як [2]:

- облігації;
- акції;
- депозити та кредити;
- нерухомість;
- товари;
- валюта (готівка).

Для усіх активів у інвестиційному портфелі власник отримує прибуток через певний проміжок часу. Цей прибуток у портфельній теорії і називається віддачою.

Основним сценарієм отримання віддачі є зростання ринкової вартості акцій. Але інколи компанії, акції якої наявні у портфелі, банкрутує. У такому випадку вартість акцій даної компанії на фондовому ринку падає і інвестор втрачає гроші.

Врахування можливості падіння вартості активів у портфелі і називається ризиком інвестування.

Найбільш популярним є інвестування в портфелі цінних паперів. На це є декілька вагомих причин [1]:

- багато реальних фінансових активів мають еквіваленти у якості цінних паперів (такі як ф'ючерсні контракти, опціони, облігації);

- керування цінними паперами більш гнучке, ніж керування нерухомістю або товарами;

- інвестиції у реальні (фізичні) активи потребує дуже великих фінансових витрат, в той час як папери значно дешевші;

- зазвичай компанії надають інформацію о дохідності цінних паперів частіше, ніж інформацію о дохідності фізичних активів, що і дозволяє інвестору оцінити ризик та віддачу.

Для створення надійнішого портфеля використовують різноманітні стратегії, вибір стратегії залежить від результату, якого хоче досягти інвестор.

1.3 Основні принципи портфельної оптимізації

До ключових елементів та стратегій створення ефективного інвестиційного портфелю відносять [5]:

- диверсифікація;
- ребалансування;
- активне управління портфелем;
- пасивне управління портфелем.

1.3.1 Диверсифікація

Розподіл активів базується на тому, що різні типи активів не є залежними один від одного. Одні активи є більш стабільними, інші – більш мінливими. Різнманітність таких активів у портфелі і дозволяє зберігати баланс та захищає від ризику. Основною проблемою інвестицій є те, що неможливо послідовно правильно прогнозувати, які активи принесуть прибуток, а які навпаки приносять збитки. Диверсифікація вирішує цю проблему. Диверсифікація – розподіл інвестицій у дуже різні активи. Зазвичай

вона здійснюється стосовно різних класів цінних паперів, різних галузей економіки, різних географічних регіонів [3].

1.3.2 Ребалансування

Ребалансування використовується для відновлення балансу у розподілі вартості активів, з яких складається портфель. Припустимо, що портфель складався на 50% з акцій та на 50% з облігацій. Після довготривалого зростання акцій баланс може зміститися за їх користь, і тепер портфель на 70% складається з акцій та на 30% з облігацій [1]. З таким розподілом ризик та віддача портфелю змінилися, і для відновлення початкового балансу інвестору потрібно продати акції та переінвестувати на користь облігацій. Якщо говорити більш загально, ребалансування передбачає продаж цінних паперів з високою ціною та використання цих грошей у цінних паперах з нижчою ціною. Потрібно проводити ребалансування один раз у певний проміжок часу. При чому, цей проміжок часу не повинен бути занадто довгим чи занадто коротким. Занадто довгий проміжок часу для проведення ребалансування змінить початкове співвідношення ризику і віддачі. Занадто короткий проміжок часу для проведення ребалансування не дозволить інвестору отримати максимальний прибуток у секторах ринку з високим потенціалом.

1.3.3 Активне та пасивне керування портфелем

Для розуміння понять активного та пасивного керування інвестиційним портфелем доцільно розібрати поняття фондового індексу. Під фондовим індексом розуміють індекс, що вираховується на основі цін конкретної групи цінних паперів, також відомої як індексної корзини. Для індексної корзини відомо кількість активів, їх алокацію у портфелі, а також віддачу за певний період часу та прогнозовану віддачу [5].

Активне управління портфелем передбачає швидку купівлю та продаж акцій на різні строки, намагаючись перевершити деякий фондовий індекс.

Для прийняття рішень про купівлю та продаж акцій використовують не лише менеджерів фондів або брокерів, але й поглиблені дослідження ринку, короткострокові прогнозування, зміни у політиці та навіть деякі неекономічні та неполітичні новини, що можуть вплинути на деякі компанії. Урахування такої великої кількості даних дозволяє отримувати більший прибуток, ніж пропонуються відомими індексними корзинами, проте такий підхід до керування портфелем включає додатковий ризик.

Пасивне управління портфелем (або управління індексними фондами) має на меті повторити віддачу конкретної індексної корзини. Це досягається за рахунок купівлі тих самих акцій, що перераховані в індексі, з використанням тієї самої ваги (алокції), що використовується в індексі.

Віддача від пасивного управління портфелем нижча, ніж віддача від активного управління, проте пасивне управління має значно нижчий ризик.

1.4 Особливості та переваги портфельного інвестування

Під інвестиційним портфелем розуміють сукупність кількох інвестиційних об'єктів, якими керують як єдиним цілим. Портфельне інвестування має ряд особливостей і переваг [2]:

- за допомогою такого інвестування можливо дати сукупності цінних паперів, об'єднаних в портфель, інвестиційні якості, які не можуть бути досягнуті при вкладенні коштів в цінні папери якогось одного емітента;
- підбір і управління портфелем цінних паперів дозволяє отримувати оптимальне поєднання прибутковості і ризику для кожного конкретного інвестора;
- портфель цінних паперів вимагає невисоких витрат в порівнянні з інвестістированієм в реальні активи, тому доступний для великого числа інвесторів;

Мета формування портфеля полягає в прагненні отримати необхідний рівень очікуваної прибутковості при більш низькому рівні очікуваного ризику. Вона досягається, по-перше, за рахунок диверсифікації портфеля за складом інструментів і, по-друге, ретельного підбору фінансових інструментів.

Головними параметрами при управлінні портфелем, які необхідно визначити менеджеру, є його очікувана прибутковість і ризик. Формуючи портфель, менеджер не може точно визначити майбутню динаміку його прибутковості і ризику [3]. Тому свій інвестиційний вибір він будує на очікуваних значеннях прибутковості і ризику. Дані величини оцінюються, в першу чергу, на основі статистичної інформації за попередні періоди часу. Оскільки майбутнє навряд повторить минуле зі стовідсотковою ймовірністю, отримані оцінки менеджер може коригувати відповідно до своїх очікувань розвитку кон'юнктури [4].

Завдання портфельного інвестування – отримати очікуваний прибуток при мінімально допустимому ризику.

Під диверсифікацією розуміється процес розподілу інвестиційних активів між різними об'єктами вкладення капіталу, які безпосередньо не пов'язані між собою, з метою зниження ступеня ризику і втрат доходів.

1.5 Висновки до розділу

В даному розділі описані основні концепти портфельної теорії. Детально описані ризики при інвестуванні в об'єкт господарської власності, а також показана залежність між прибутком та ризиками. Враховуючи основні принципи інвестиційної теорії, задачею портфельного інвестування є формування інвестиційного портфелю з відомими частками певних активів для максимізації віддачі та мінімізації ризику. В наступному розділі пояснювальної записки даний вибір теоретично обгрунтовано та описано, зроблено вибір методів інвестування щодо поставленої задачі дослідження.

2 ОПТИМІЗАЦІЯ ІНВЕСТИЦІЙНОГО ПОРТФЕЛЮ

2.1 Основні концепції сучасної портфельної теорії

Гарі Марковіц створив теорію, яка дозволяє інвестору математично балансувати терпимість до ризику та очікування винагороди, що призводить до ідеального портфеля. Ця теорія базувалася на двох основних концепціях [6]:

- метою кожного інвестора є максимізація прибутковості для будь-якого рівня ризику;
- ризик можна зменшити, диверсифікувавши портфель за допомогою окремих, не пов'язаних між собою цінних паперів.

Сучасна портфельна теорія (СПТ) працює, припускаючи, що інвестори не схильні до ризику, віддаючи перевагу портфелю з меншим ризиком для певного рівня прибутковості [6]. Згідно з цим припущенням, інвестори будуть здійснювати інвестиції з високим ризиком лише тоді, коли вони можуть розраховувати на більшу винагороду.

Розглянемо наступний приклад. «Раціональному інвестору» пропонується вибрати між двома інвестиціями: інвестицією А та інвестицією В. Очікується, що вартість обох зростатиме на 6 відсотків щороку. Однак інвестиція В вважається вдвічі більш мінливою, ніж інвестиція А, тобто її вартість коливається удвічі більше, ніж коливання вартості інвестиції А.

СПТ пропонує, щоб раціональний інвестор завжди вибирав менш мінливий актив, в даному випадку інвестицію А, якщо обидва варіанти забезпечують рівноцінну очікувану прибутковість.

Загальний ризик портфеля обчислюється на основі функції дисперсії кожного активу разом із кореляцією між кожною парою активів. Кореляції активів впливають на загальний ризик портфеля, формулюючи менше середньоквадратичне відхилення, ніж можна було б знайти у зваженій сумі.

Згідно з аналізом СПТ – (або аналізом дисперсії, аналізом середньоквадратичного відхилення) – інвестор може мати актив із високим ризиком, індексний фонд або цінний папір, доки ці ризикові інвестиції будуть

мінімізовані всіма базовими активами. Сам портфель збалансований таким чином, що загальний ризик є нижчим, ніж деякі основні інвестиції. Ризик визначається як діапазон, в якому ціна активу в середньому буде змінюватися, але Марковіц поділяє ризик на дві наступні категорії [7].

Систематичний ризик: це стосується ринкових ризиків, які неможливо зменшити за рахунок диверсифікації, або можливості того, що весь ринок та економіка покажуть збитки, що негативно впливають на інвестиції. Важливо зазначити, що СПТ не претендує на можливість пом'якшити цей тип ризику, оскільки він притаманний цілому ринку або сегменту ринку.

Несистематичний ризик: також називається специфічним ризиком, несистематичний ризик характерний для окремих акцій, тобто він може бути диверсифікований у міру збільшення кількості запасів у вашому портфелі.

У добре диверсифікованій комбінації активів ризик кожного активу дуже мало сприяє загальному ризику портфеля. Швидше, коваріація між окремими активами визначає більшу частину загального ризику портфеля.

Отже, інвестори можуть зменшити індивідуальний ризик активів, комбінуючи диверсифікований портфель активів.

2.2 Баланс ризику та прибутковості

Хоча вигоди від диверсифікації очевидні, інвестори повинні визначити рівень диверсифікації, який найкраще їм підходить. Це можна визначити за допомогою так званої ефективної межі, графічного представлення всіх можливих комбінацій ризикових цінних паперів для оптимального рівня прибутковості з урахуванням певного рівня ризику [5]:

- на кожному рівні прибутковості інвестори можуть створити портфель, що пропонує мінімально можливий ризик;
- для кожного рівня ризику інвестори можуть створити портфель, що пропонує найвищу віддачу.

Будь-який портфель, що потрапляє за межі ефективної межі, вважається неоптимальним з однієї з двох причин: він несе занадто великий ризик щодо його прибутку або занадто малу віддачу щодо його ризику. Портфель, який лежить нижче ефективної межі, не забезпечує достатньої прибутковості порівняно з рівнем ризику. Портфелі, знайдені праворуч від ефективної межі, мають вищий рівень ризику для визначеної норми прибутку.

У кожному пункті «Ефективної межі» інвестори можуть побудувати принаймні один портфель з усіх доступних інвестицій, що характеризує очікуваний ризик і дохід, що відповідає цій точці. Портфель, розміщений у верхній частині кривої, є ефективним, оскільки дає максимальну очікувану дохідність для даного рівня ризику.

Ефективна межа пропонує наочну демонстрацію потужності диверсифікації [5]. Не існує єдиної ефективної межі, оскільки інвестори можуть змінювати кількість та характеристики активів відповідно до своїх потреб. На рис. 2.1 проілюстроване поняття ефективної межі.



Рисунок 2.1 – Ефективна межа ризиків

2.3 Математична реалізація портфельної оптимізації

Для реалізації оптимізації портфелю потрібно розглянути декілька важливих метрик та формул, а саме [6]:

- дисперсія активу та нестабільність активу;
- коваріація та кореляція активів;
- очікувана прибутковість активу та портфеля;
- дисперсія портфеля;
- коефіцієнт Шарпа.

Дисперсія активу – важливий індикатор, що відображає як може коливатися прибуток від інвестиції у даний актив.

Нестабільність активу вимірюється як середньоквадратичне відхилення акцій компанії. Фактично, це просто корінь квадратний з дисперсії активу.

Дисперсія активу розраховується за формулою (2.1) [6]:

$$\sigma^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}, \quad (2.1)$$

де x_i – прибутковість активу за i -й період часу;

$\bar{x}$ – середнє значення прибутковості активу;

n – кількість екземплярів вибірки (кількість днів/місяців/періодів часу).

Коваріація вимірює залежність між прибутковістю двох окремих активів. Більш загально, коваріація вимірює взаємодію двох випадкових величин. Для сфери інвестицій це міра, що враховує дисперсію індивідуальних значень дохідності цінного паперу та силу зв'язку між змінами дохідності даного цінного паперу та інших.

Позитивна коваріація означає, що віддачі двох активів прямо пропорційні, негативна коваріація означає, що віддачі двох активів обернено пропорційні.

Ця міра може приймати необмежені значення, на відміну від свого стандартизованого родича, кореляції. Інтерпретувати коваріацію дещо важко за рахунок великих чисельних значень, тому найчастіше для оцінки зв'язку між активами використовують коефіцієнт кореляції.

Коефіцієнт кореляції лежить в інтервалі від -1 до 1. Значення -1 та 1 означає сильну лінійну залежність, але у різних напрямках, значення ближче до 0 означає відсутність лінійної залежності.

Розрахування коваріації та кореляції відбувається за наступними формулами (2.2) та (2.3) [6]:

$$cov(X, Y) = \frac{\sum (R_x - \bar{R}_x)(R_y - \bar{R}_y)}{n-1}, \quad (2.2)$$

$$cor(X, Y) = \frac{cov(X, Y)}{\sigma_x \sigma_y}, \quad (2.3)$$

де R_x R_y – значення прибутковості активів;

n – розмір вибірки (кількість місяців);

R_x та R_y – середні значення прибутковості активів;

σ_x та σ_y – середньоквадратичні відхилення активів.

Очікувана прибутковість активу вимірюється як середнє арифметичне прибутковості за певний період, найчастіше за останній рік. Очікувана прибутковість портфелю вираховується за формулою (2.4) як сума добутків ваги активу на його індивідуальну очікувану прибутковість [6]:

$$R_p = \sum w_i R_i, \quad (2.4)$$

де R_p – очікувана прибутковість портфелю;

w_i – вага i -го активу в портфелі;

R_i – очікувана прибутковість i -го активу в портфелі.

Дисперсія портфелю (2.5) використовується як міра ризику в побудованій моделі. Висока дисперсія (волатильність) означає високий ризик для набору активів у портфелі [7]:

$$\sigma^2 = \sum w_i^2 \sigma_i^2 + \sum \sum w_i w_j \sigma_i \sigma_j \text{cov}(i, j), \quad (2.5)$$

де w_i та w_j – ваги активів у портфелі;

σ_i та σ_j – середньоквадратичне відхилення;

$\text{cov}(i, j)$ – коваріація активів.

Коефіцієнт Шарпа вимірює, скільки надлишкової віддачі ви отримуєте за додаткову нестабільність, яку ви зазнаєте за зберігання більш ризикованого активу. Загалом, вище значення коефіцієнта Шарпа вказує на кращі та вигідніші інвестиції. Таким чином, якщо порівнювати два портфелі з подібними профілями ризику, враховуючи всі інші рівні, було б краще інвестувати в портфель із вищим коефіцієнтом Шарпа.

Розрахування коефіцієнту Шарпа проводиться за формулою (2.6) [6]:

$$\frac{R_p - R_f}{\sigma_p}, \quad (2.6)$$

де R_p – віддача портфелю;

R_f – безризикова ставка;

σ_p – волатильність портфелю.

2.4 Методи оптимізації портфелю

2.4.1 Випадкова генерація портфелів

Під випадковою генерацією портфелів розуміється випадковий розподіл активів у портфелі таким чином, щоб сумарна вага усіх активів становила одиницю. Випадкова генерація активів не гарантує оптимального результату,

саме тому для випадкового згенерованих портфелів потрібно проводити фільтрації та обробки. Попри такий важливий недолік, у випадкової генерації портфелів наявні дуже важливі переваги [8]:

- випадкова генерація портфелів часто дозволяє згенерувати велику кількість портфелів, не потребуючи даних про розподіл з індексних фондів;
- згенеровані портфелі можна використовувати для математичної оптимізації, кожного разу створюючи нові умови для мінімізації чи максимізації функції за заданими обмеженнями.

З випадково згенерованих портфелів було обрано найкращі за коефіцієнтом Шарпа та за мінімізацією волатильності портфелю. На рис. 2.2 зображено результат випадкової генерації портфелів.

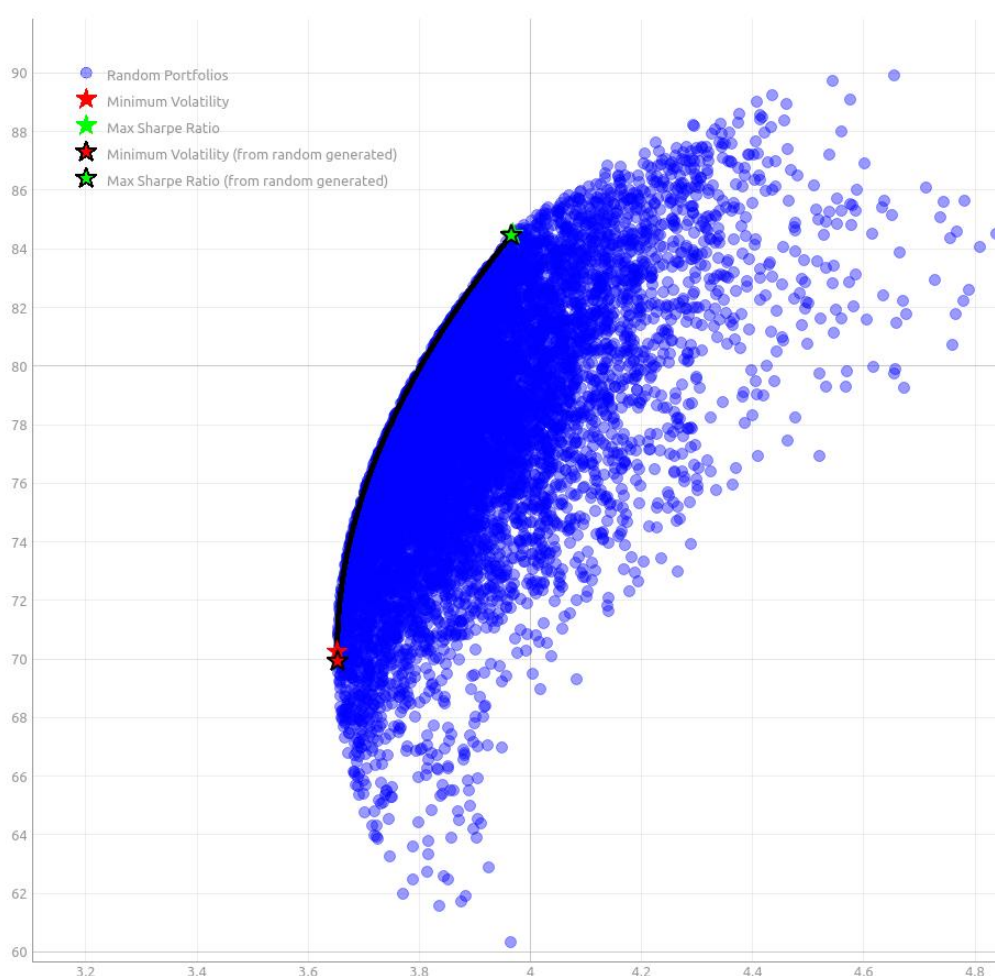


Рисунок 2.2 – Випадкова генерація портфелів

2.4.2 Оптимізація шляхом максимізації коефіцієнту Шарпа

Визначення коефіцієнту Шарпа пов'язано з концепцією віддачі з корекцією на ризик. Віддача, скоригована на ризик, уточнює віддачу інвестиції, вимірюючи, наскільки сильно ризик пов'язаний з її прибутком. Може існувати ціла низка різних методів вираження коригування ризику, і коефіцієнт Шарпа є одним із них [5]. Оптимізація шляхом максимізації коефіцієнту Шарпа полягає у тому, щоб максимізувати надлишкову віддачу, при цьому не сильно підвищуючи ризики. Оскільки функції оптимізації здебільшого саме на мінімізацію функції, для реалізації даного методу було використано негативний коефіцієнт Шарпа і мінімізовано його значення. Для того, щоб оптимізація спрацювала враховуючи обмеження предметної області, потрібно задати деякі лінійні обмеження. Єдиним обмеженням в нашому випадку є сума ваг усіх активів портфелю, її значення завжди має дорівнювати одиниці. Для оптимізації було використано функцію бібліотеки `scipy.optimize` з назвою `minimize`. Дана функція також дозволяє задати початкові значення для аргументів, які вона і буде оптимізувати [7]. В нашому випадку було встановлено рівний розподіл активів у портфелі. Також було проведено декілька тестів з використанням згенерованих портфелів. На рис. 2.3 зображено портфель з максимальним коефіцієнтом Шарпа серед випадково згенерованих портфелів, та портфель з оптимізованим максимальним коефіцієнт Шарпа [8].

2.4.3 Оптимізація шляхом мінімізації волатильності портфелю

Для оптимізації даної функції було накладено такі самі обмеження, як і на максимізацію коефіцієнту Шарпа. Сума ваг усіх активів має бути рівною одиниці. Для оптимізації так само було використано функцію `minimize` з бібліотеки `scipy.optimize`, тому можна регулювати початкові значення ваг активів у портфелі і встановлювати значення з випадково згенерованих портфелів. На рис. 2.4 зображено портфель з мінімальною волатильністю [7].

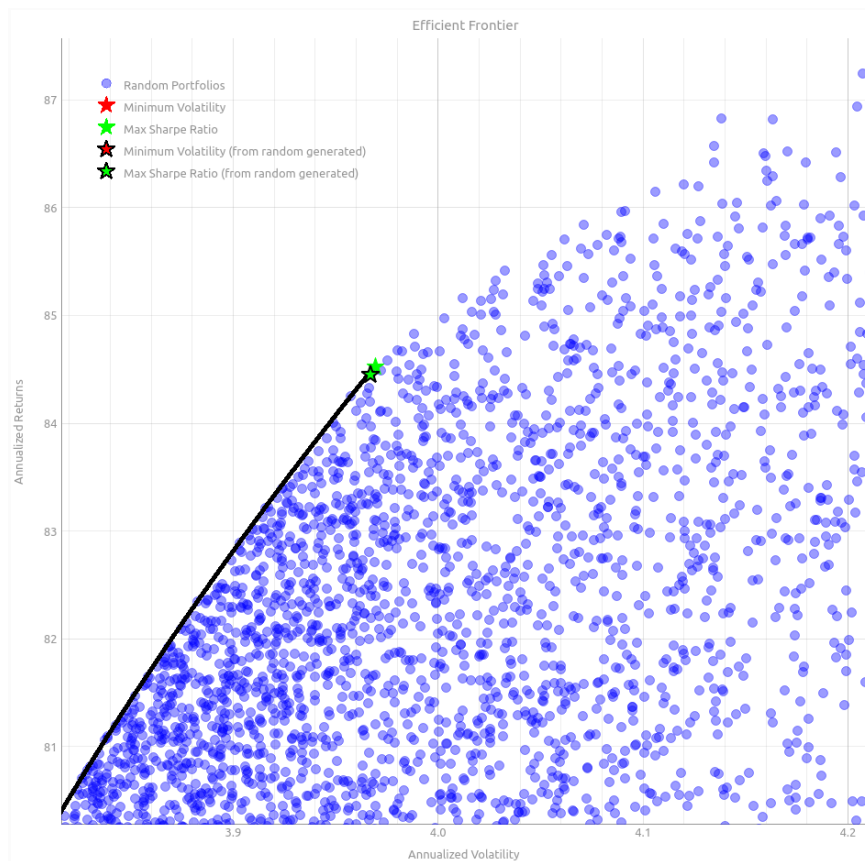


Рисунок 2.3 – Максимальний та оптимізований коефіцієнти Шарпа

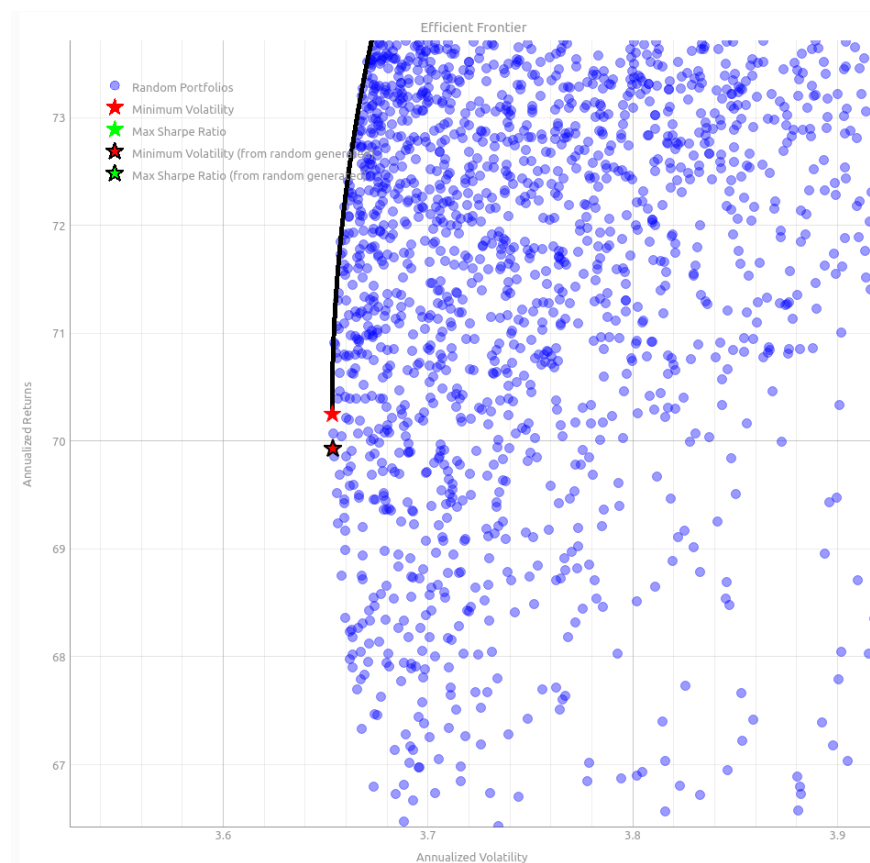


Рисунок 2.4 – Мінімальна та оптимізована волатильність портфелів

2.5 Оптимізація інвестиційного портфеля за методом Марковіца

Портфельна теорія Марковіца спрямована на оптимальний вибір активів, виходячи з необхідного співвідношення прибутковості до ризику [7]. Сформульовані ним ідеї складають основу сучасної портфельної теорії. Після проведеної Марковицем формалізації, з математичної точки зору завдання по формуванню оптимального портфеля представляла собою задачу квадратичної оптимізації при лінійних обмеженнях. Цей клас завдань, є одним з найбільш вивчених класів оптимізаційних задач, для яких існує велика кількість ефективних алгоритмів. Для побудови простору можливих портфелів Марковіц запропонував використовувати клас активів, вектор їх середніх очікуваних доходностей і матрицю коваріацій. На основі цих даних будується безліч можливих портфелів з різними співвідношеннями прибутковості-ризик. Приклад залежності показано на рис. 2.5.

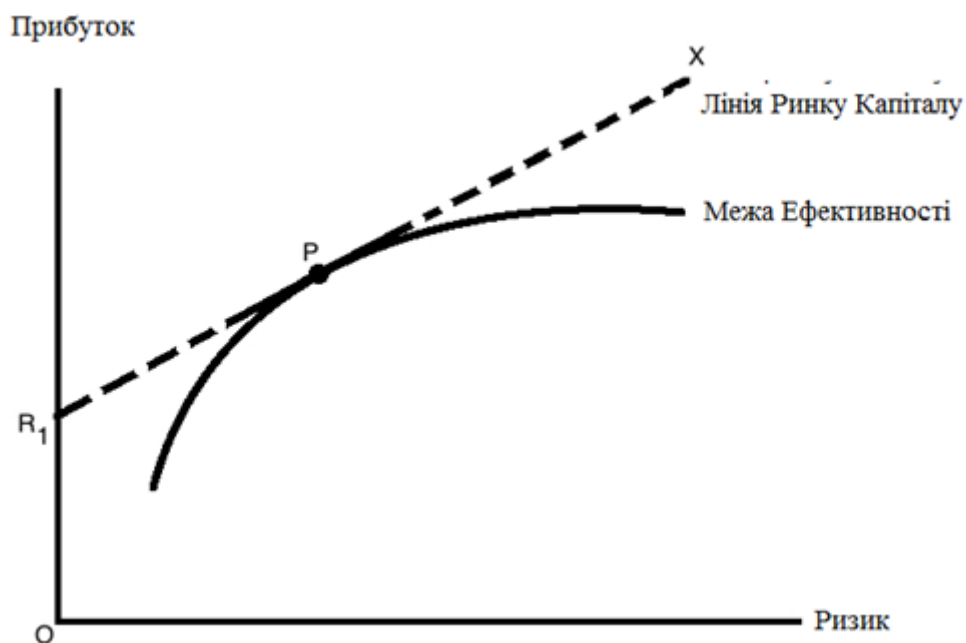


Рисунок 2.5 - Крива ризик-прибуток в моделі Марковіца

Так як в основі аналізу лежать два критерії, менеджер вибирає наступні портфелі [2]:

- пошук ефективних, або рішень, що неможливо покращити. У цьому випадку будь-яке інше рішення, краще знайдених по одному параметру обов'язково буде гірше за іншим;
- вибір головного критерію (наприклад, прибутковість повинна бути не нижче певної величини) інші використовуючи лише в якості критеріальних обмежень;
- задаючи суперкритерій, який є суперпозицією зазначених двох.

Перевагою даного методу є наочність отриманих результаті. Після проведення безлічі спроб скласти портфель, в результаті отримуємо досить гарний набір даних, з якого інвестор має змогу обрати саме той портфель, що задовольняє його потреби. Проте моделювання великої кількості даних вимагає певних витрат, як часу, так і ресурсі машини. Після отримання початкових даних, таких як вартість акції компанії p за певний період часу, далі за формулою (2.7) знадобляться відносні зміни до попереднього періоду [7].

$$P = \frac{t_i - t_{i-1}}{t_i}, \quad (2.7)$$

де P – різниця вартості акцій;

t – період.

Далі необхідно вирахувати математичне очікування (2.8), що є ознакою прибутку, дисперсію (2.9) та середньоквадратичне відхилення (2.10) для кожної компанії [8]:

$$M(x) = \frac{\sum_{t=1}^n x_t}{n}, \quad (2.8)$$

$$D(x) = \frac{\sum_{t=1}^n (M(x) - x_t)^2}{n - 1}, \quad (2.9)$$

$$\sigma(x) = \sqrt{D(x)} \quad (2.10)$$

Після цього необхідно розрахувати матрицю коваріацій (2.11).

$$\text{cov}(X, Y) = M(XY) - M(X)M(Y) \quad (2.11)$$

При чому за формулою (2.12) коваріація випадкової величини з самою собою є її дисперсією.

$$\text{cov}(X, X) = D(x) \quad (2.12)$$

Після знаходження усіх вищеперерахованих значень можна приступати до формування набору портфелів. Загальна прибутковість портфеля (2.13) буде являти собою зважену суму прибутковості кожного окремого фінансового інструменту (активу):

$$r_p = \sum_{i=1}^n w_i * r_i \quad (2.13)$$

де r_p – дохідність інвестиційного портфеля;

w_i – частка i -го фінансового інструмента в портфелі;

r_i – дохідність i -го фінансового інструмента.

У моделі Марковіца (рис. 2.6) ризик окремо взятого фінансового інструменту розраховується як стандартне відхилення доходностей. Для розрахунку загального ризику портфеля необхідно відобразити їх сукупну зміну і взаємний вплив через коваріацію (2.14):

$$\sigma_p = \sqrt{\sum_i \sum_j w_i w_j \text{cov}(i, j)} \quad (2.14)$$

де σ_p – ризик інвестиційного портфелю.

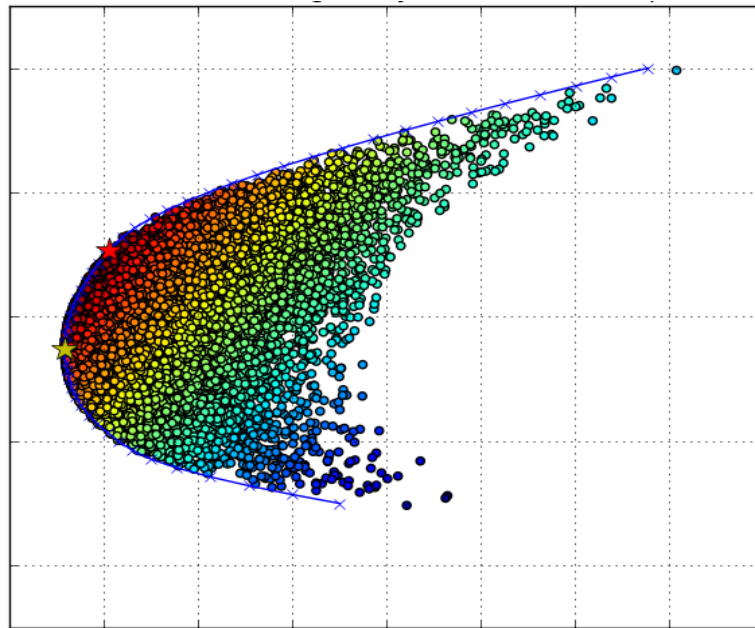


Рисунок 2.6 – Приклад формування набору портфелів за методом Марковіца

В результаті, окрім екстремумів, тобто максимальних (мінімальних) значень ризику та прибутковості, досить раціонально підібрати оптимальний варіант, як мав би найбільший показник співвідношення ціна-якість, тобто оптимальний прибуток з оптимальним ризиком. Один з варіантів, це використати коефіцієнт Шарпа. Даний коефіцієнт є класичним для оцінки портфелю, результатів фонду або торгівельної системи. Коефіцієнт був розроблений Вільямом Шарпом як частка відділення різниці прибутковості активу і безризикової ставки до стандартного відхилення активу. Формула має наступний вид (2.15) [8]:

$$Sharp = \frac{r_a - r_0}{\sigma}, \quad (2.15)$$

де r_a – прибутковість активу; r_0 – прибутковість безризикового активу; σ – стандартне відхилення активу.

Загалом, модель Марковіца має декілька недоліків [7]:

- модель була розроблена для ефективних ринків капіталу, на яких спостерігається постійне зростання вартості активів і відсутні різкі коливання курсів. Кореляція між акціями не постійна і змінюється з часом, в результаті в майбутньому це не зменшує систематичний ризик інвестиційного портфеля;

- майбутня прибутковість фінансових інструментів (акцій) визначається як середньоарифметичне. Даний прогноз ґрунтується тільки на історичному значенні доходностей акції і не включає вплив макроекономічних (рівень ВВП, інфляції, безробіття, галузеві індекси цін на сировину і матеріали) і мікроекономічних чинників (ліквідність, рентабельність, фінансова стійкість, ділова активність компанії);

- ризик фінансового інструменту оцінюється за допомогою заходів мінливості дохідності щодо середньоарифметичного, але зміна прибутковості вище не є ризиком, а являє собою надприбутковість акції.

2.6 Висновки до розділу

В даному розділі були проаналізовані сучасні методи портфельної оптимізації: випадкова генерація портфелів, максимізація коефіцієнту Шарпа, мінімізація шляхом мінімізації волатильності портфелю. Враховуючи основні принципи інвестиційної теорії, задачею портфельного інвестування є формування інвестиційного портфелю з відомими частками певних активів для максимізації віддачі та мінімізації ризику. Для вирішення даної проблеми було обрано методику Гарі Марковіца, бо у межах якої за допомогою статистичних методів здійснюють якомога вигідніший розподіл ризику портфеля цінних паперів і оцінювання доходу, складовими елементами її є оцінювання активів, прийняття інвестиційних рішень, оптимізація портфеля, оцінювання отриманих результатів.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

Проектування – це етап життєвого циклу розроблення програмних систем, наступний після інженерії вимог. Завданням цього етапу є перетворення побажань замовників системи у проєктні рішення, що забезпечать здійснення згаданих побажань у формі відповідної системи програмування. Таким чином, під час проєктування виконується трансформація простору вимог у простір проєктних рішень. В основі проєктування лежить парадигма подолання складності загального завдання шляхом декомпозиції цільового продукту на окремі його складові або компоненти [9].

3.1 UML-діаграми. Діаграма Use-Case

UML надає стандартний спосіб створення візуальних описів системи, включаючи опис бізнес-процесів, принципів функціонування, а також більш конкретні речі, такі як опис фрагментів коду мови програмування, схем баз даних та компонентів програмного забезпечення. Способів, якими можна використовувати UML, дуже багато. Сама мова UML не визначає конкретних методів або процедур опису. Вона просто надає стандартні типи діаграм UML для опису ваших програмних систем з тих чи інших сторін. UML може бути використана у складі вже відомих методологій розробки програмного забезпечення, наприклад, Rational Unified Process. Або просто використана за необхідністю більш зручно описати ту чи іншу сторону чи особливість програми, яку ви розробляєте, за допомогою конкретної діаграми [9].

Діаграма варіантів використання – це вид поведінкових діаграм UML, яка є найкорисніша при висвітленні функціональних особливостей програми для людей, що не мають глибоких знань в ІТ галузі. Наприклад, замовників софту. Ця діаграма використовується для опису цілей, які переслідує користувач вашої програми, будь це або людина, або інша програма, що користується системою. Діаграма варіантів використання застосовується для

опису функціональних вимог до програми, або її підсистем, або іншого об'єкта. Вона представляє загальну картину того, як ваша програма буде використовуватись. Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених межею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання. Суть діаграми прецедентів полягає в тому, що проєктована система подається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання [10]. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система під час діалогу з актором. На рис. 3.1 показано діаграму варіантів використання до проєкту.



Рисунок 3.1 – Діаграма варіантів використання

3.2 Розробка діаграми послідовності

Діаграма послідовності — відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень. Іншими словами, діаграма послідовностей відображає часові особливості передачі і прийому повідомлень об'єктами. Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертаються результати, які пов'язані з повідомленнями [10]. Діаграма послідовності побудована на рис. 3.2.

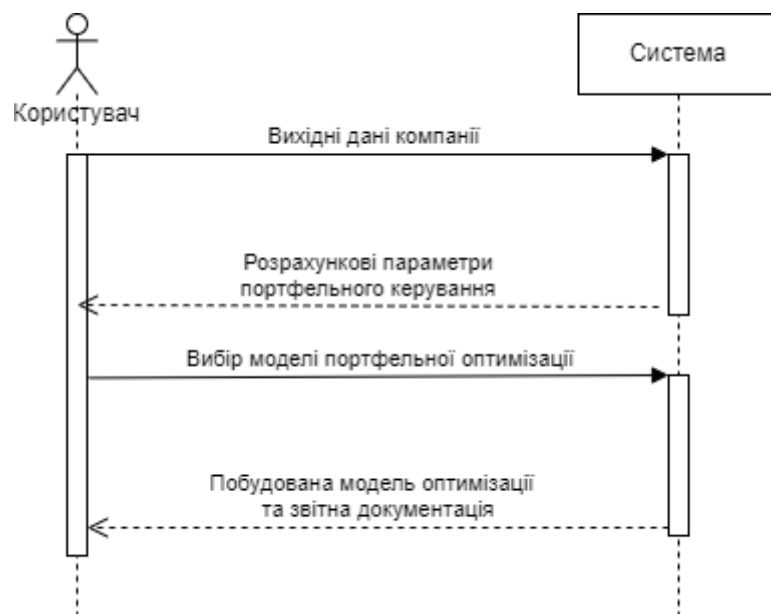


Рисунок 3.2 – Діаграма послідовності

3.3 Розробка діаграми розгортання

Діаграма розгортання — це діаграма, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час

роботи програми на таких діаграмах не відображаються. Діаграма розгортання в UML моделює фізичне розгортання артефактів на вузлах [10]. Побудована діаграма розгортання представлена на рис. 3.3.

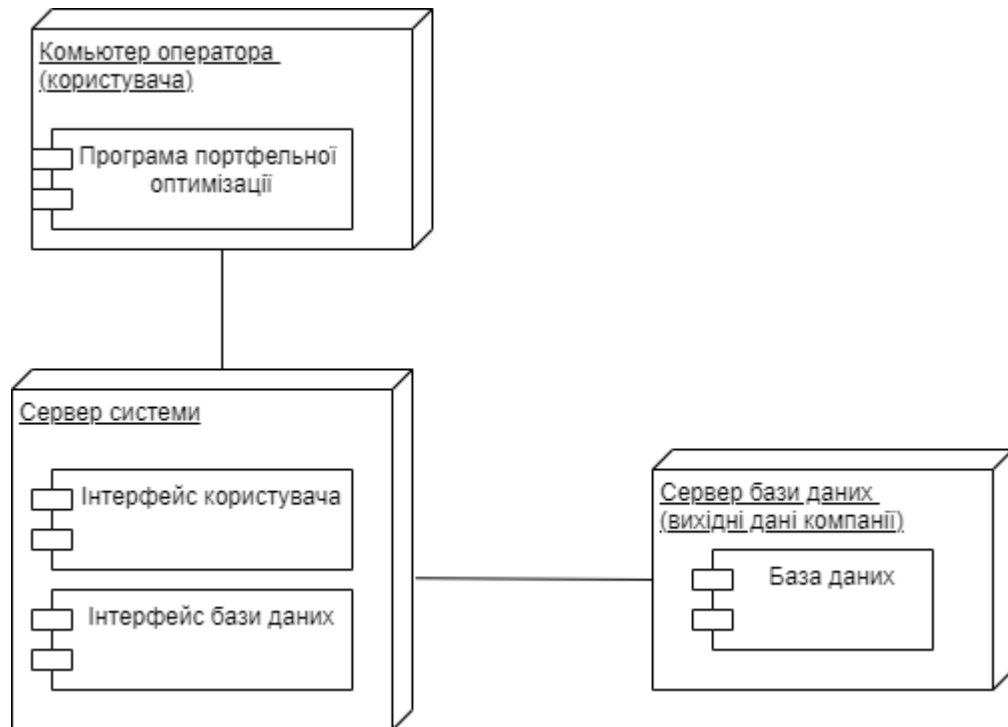


Рисунок 3.3 – Діаграма розгортання

3.4 Розробка діаграми діяльності

Діаграма діяльності — візуальне представлення графу, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Граф діяльності є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше

разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності [10]. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку. Діаграма діяльності показана на рис. 3.4.

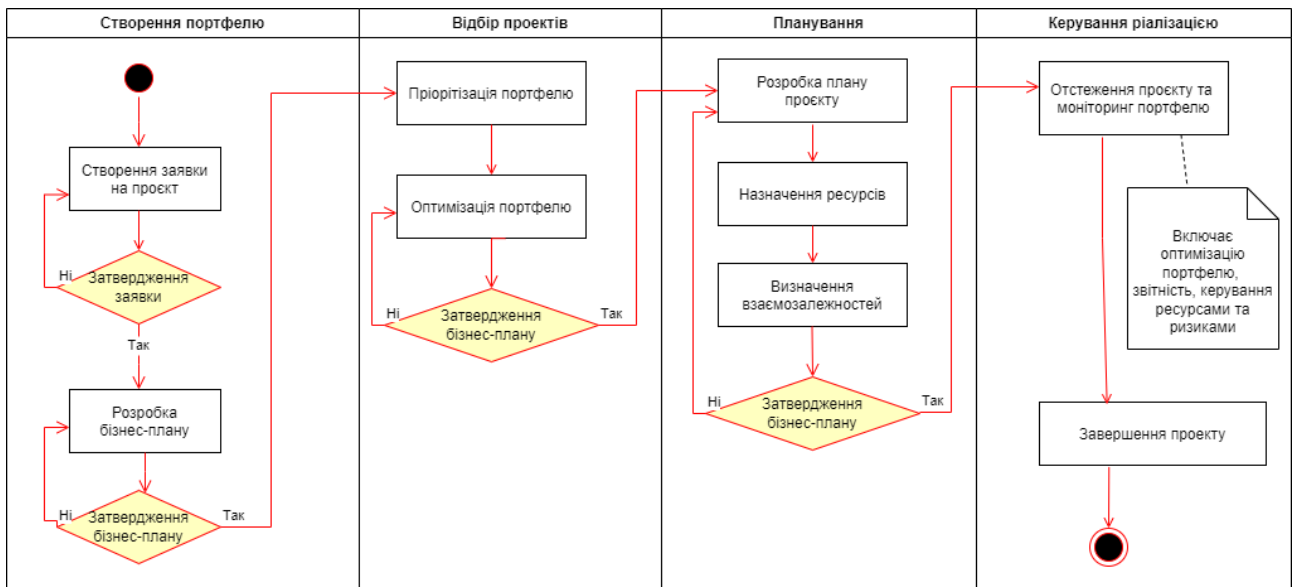


Рисунок 3.4 – Діаграма діяльності

3.5 Висновки до розділу

В даному розділі виконано проектування розробленого програмного продукту за допомогою UML специфікації, оскільки UML надає стандартний спосіб створення візуальних описів системи, включаючи опис бізнес-процесів, принципів функціонування, а також більш конкретні речі, такі як опис фрагментів коду мови програмування, схем баз даних та компонентів програмного забезпечення. Представлені діаграми: варіантів використання, послідовності, розгортання та діяльності.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Технічні засоби розробки

Програмний продукт розроблено за допомоги мови програмування C++ та фреймворку QT. QT – це крос-платформовий інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної операційної системи без зміни сирцевого коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Бібліотека дозволяє керувати потоками, працювати з мережею та забезпечує крос-платформовий доступ до файлів. Qt також може бути використаним у багатьох інших мовах програмування: C#, Java, Qt Jambi, Node.js, Pascal, Perl, PHP, Ruby, Python.

В даному проєкті застосовано Qt5, який примітний модульною структурою та зміщенням акценту в бік використання для написання застосунків засобів декларативного опису інтерфейсу з визначенням логіки взаємодії з користувачем мовою JavaScript, у той час як застосування C++ позиціонується для реалізації критичних до часу виконання або надмірно складних частин програми, а також для створення нових модульних бекендів для Qt Quick. Незважаючи на багато істотних поліпшень і змін, Qt5 зберігає базову зворотну сумісність із минулими випусками, підтримує повною мірою засоби для створення Qt-програм мовою C++ і містить майже всі компоненти Qt4 (припинена підтримка давно застарілих елементів), більшість модулів колишнього Qt Mobility й деякі експериментальні елементи з Qt Labs. До складу Q5 входять:

– базові бібліотеки – Qt Core, Qt Network, Qt Gui, Qt Sql, Qt Testlib, Qt Widgets, Qt QML, Qt Quick, Qt Multimedia, Qt WebKit, Qt WebKit Widgets;

– доповнення – Qt Xml, Qt XmlPatterns, Qt Svg, Qt Concurrent, Qt Printsupport, Qt DBus, Qt OpenGL, Qt ActiveQt, Qt Graphical Effects, Qt Script, Qt Declarative, Qt Image Formats;

– інструменти та супутні проєкту елементи – Qt Assistant, Qt Linguist, Qt Designer, Qt Translations, Qt Creator, Qt Documentation.

До комплекту Qt включене середовище розробки Qt Creator, яке містить у собі редактор коду, довідку, графічні засоби Qt Designer і можливість зневадження застосунків. Qt Creator може використовувати GCC або Microsoft VC++ як компілятори і GDB як зневаджувач. Для Windows-версій бібліотека комплектується компілятором, заголовними й об'єктними файлами MinGW. Підтримується розробка як програм мовою C++, так і використання мови QML для визначення сценаріїв, у якій використовується JavaScript, а структура й параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt комплектується візуальним середовищем розробки графічного інтерфейсу Qt Designer, що дозволяє створювати діалоги й форми мишею (рис. 4.1).

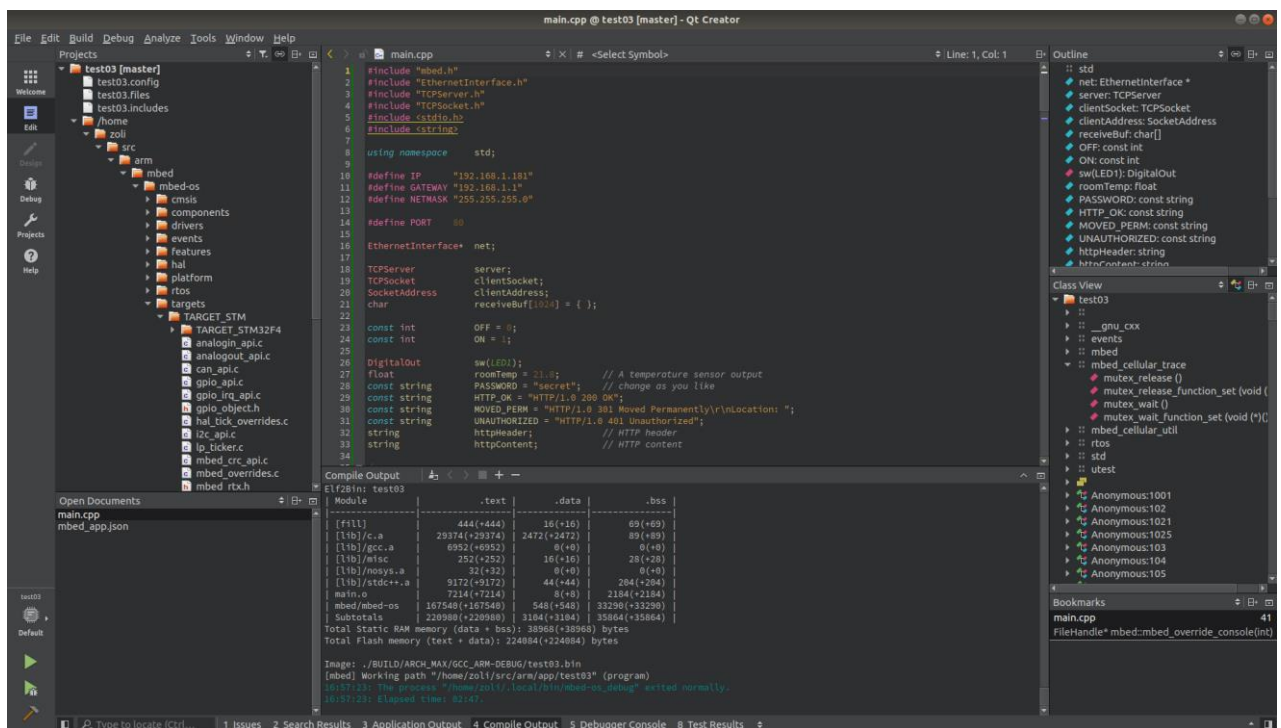


Рисунок 4.1 – Середовище розробки QT

4.2 Вихідні дані до проєкту

Відповідно завданню до дипломному проєкту було розроблено програмний продукт, що дозволяє аналізувати вихідні дані (акції металургійних підприємств за певний період), та на їх основі побудувати модель Марковіца. Відповідно моделі надаються варіанти зіставляння активів у портфелі, а програмний продукт надає результати портфелю з максимальною ефективністю, відповідно до коефіцієнту Шарпа.

4.3 Розробка інтерфейсу програмного продукту

Після запуску програмного продукту загальний зовнішній вид відповідає рис. 4.2.

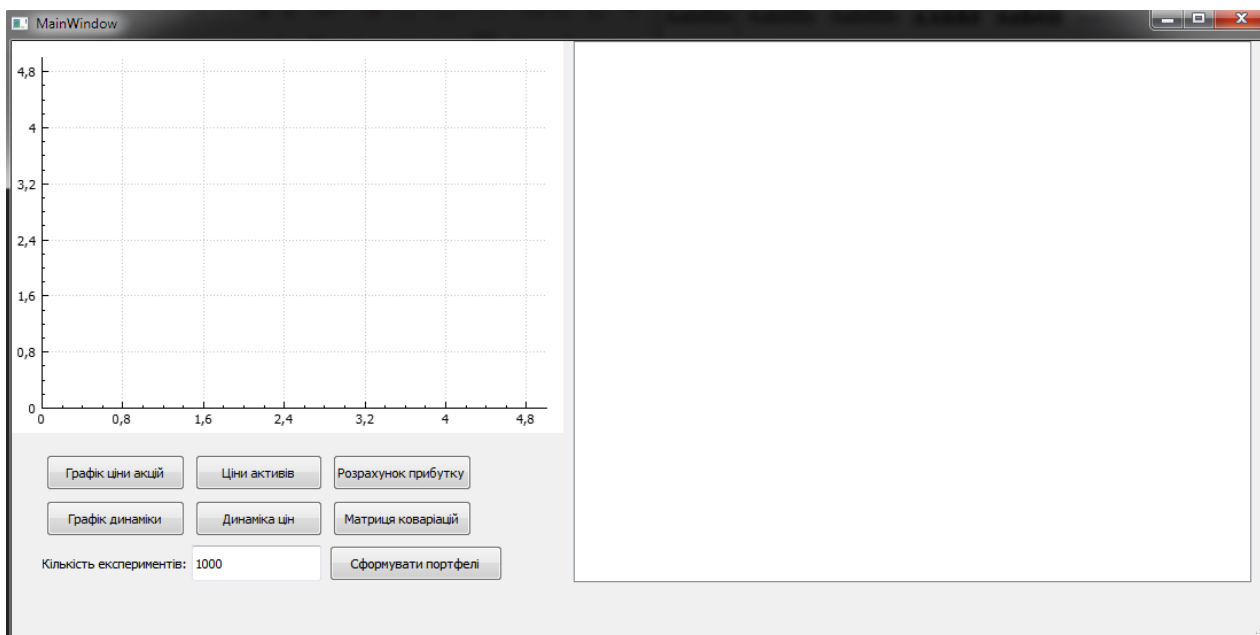


Рисунок 4.2 – Зовнішній вигляд програми

В даному описі програми завантажені тестові дані. Далі при натисненні на кнопки «Графік цін акцій» та «Графік динаміки» буде відображено відповідно вартості акцій певної компанії та динаміку зміни вартості. При повторному натисненні буде відображено графік для іншої компанії. Також,

кнопки «Ціни активів» та «Динаміка цін» відображають в таблиці ціни акцій за певні періоди та їх зміну за певний період відповідно. Приклади зображені на рис. 4.3. та рис. 4.4.

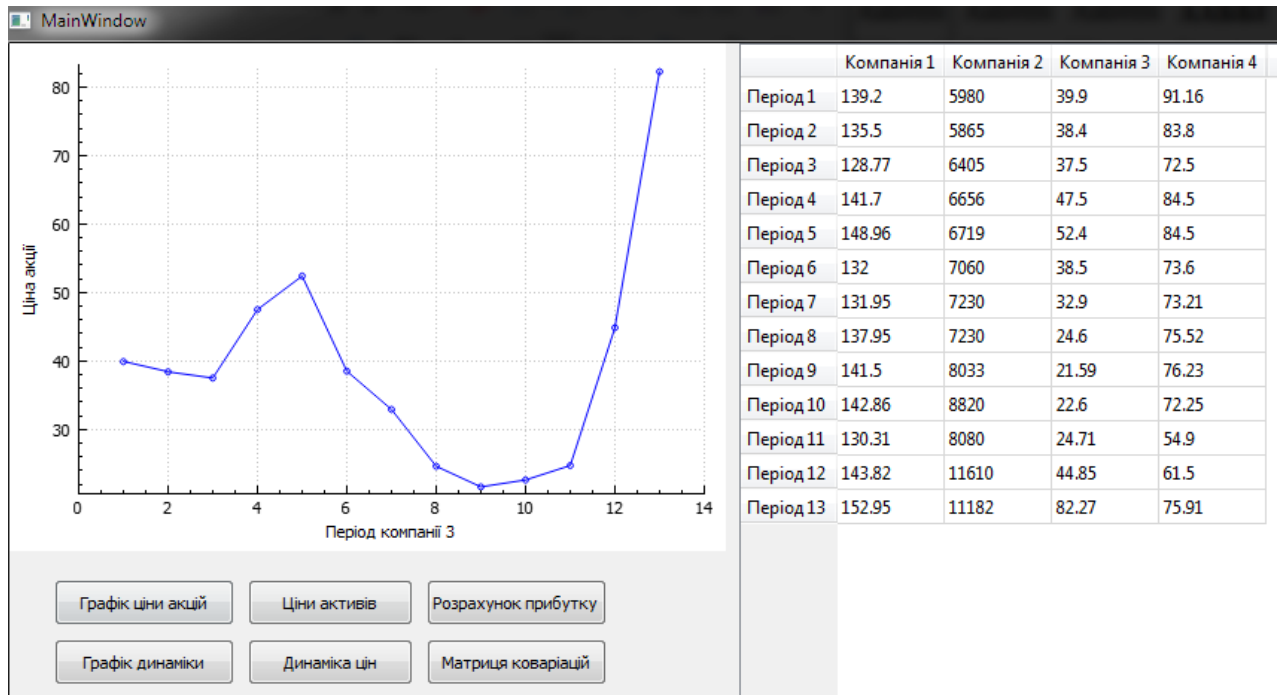


Рисунок 4.3 – Відображення графіку та таблиці цін активів

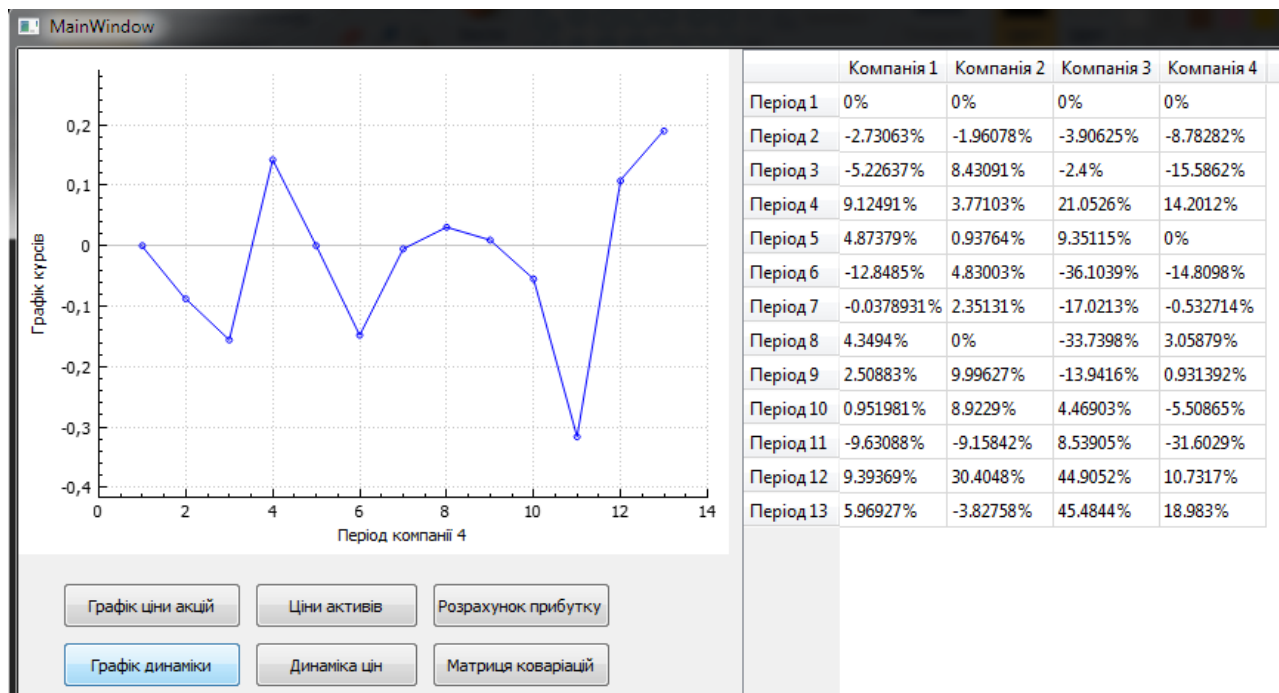


Рисунок 4.4 – Відображення графіку та таблиці динаміки цін активів

Натиснення на кнопку «Розрахунок прибутку» виведе таблицю з математичним очікуванням (прибутком), дисперсією та середньоквадратичним відхиленням (ризиком) усіх компаній, що зображено на рис. 4.5. Для перегляду матриці коваріацій необхідно натиснути на однойменну кнопку, що відобразить її у таблиці, як на рис. 4.6.

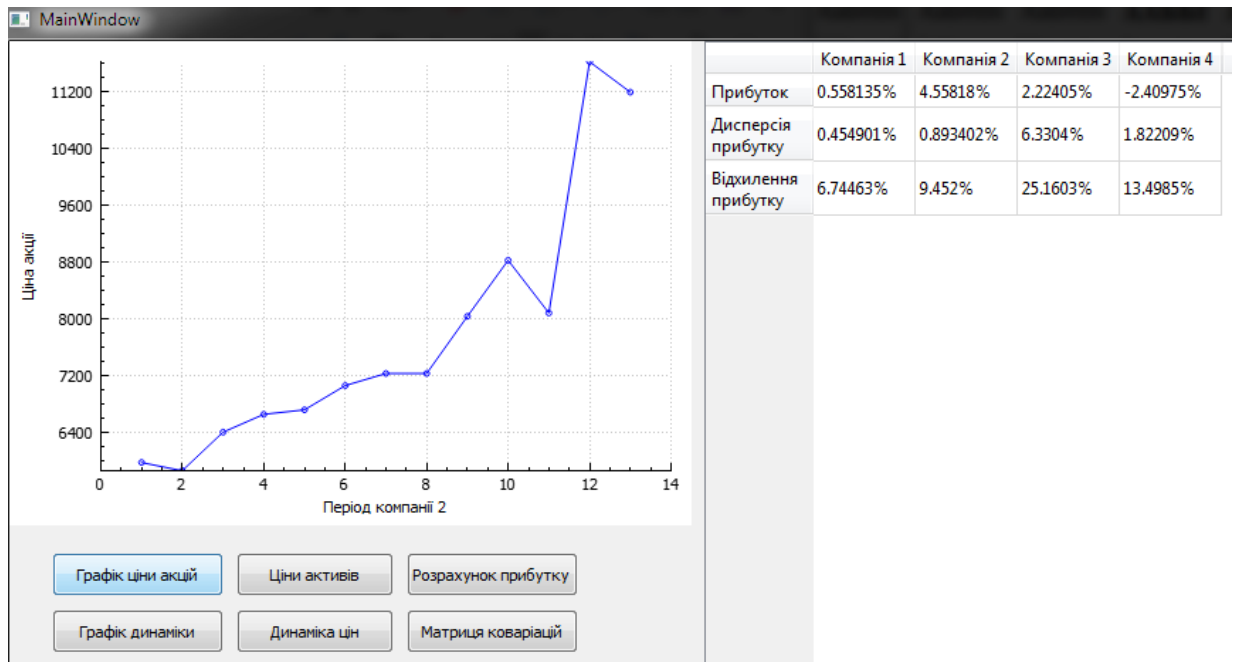


Рисунок 4.5 – Відображення таблиці ризику та прибутковості активів

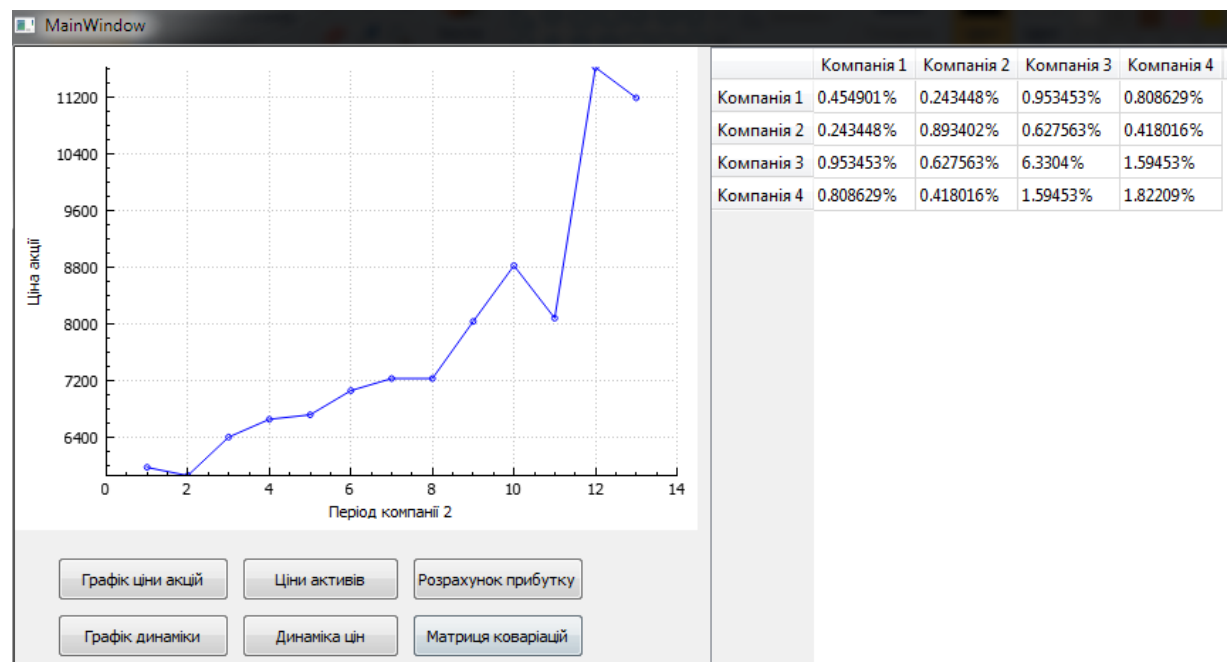


Рисунок 4.6 – Відображення матриці коваріацій

Кнопка «Сформувати портфелі» виконує побудову моделі Марковиця і відображення її на графіку прибуток/ризик на цьому вікні, та відкриє нове для більш детального його перегляду. Перед її натисненням, зліва неї є поле для вводу кількості портфелів, що будуть сформовані в моделі (рис. 4.7, рис. 4.8).

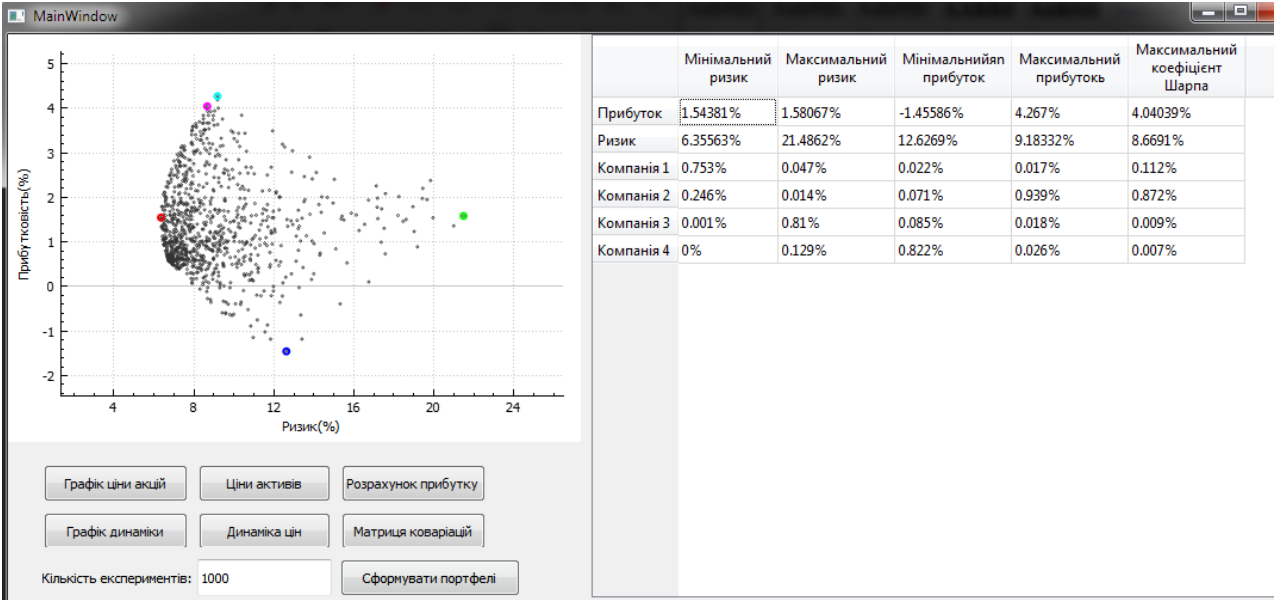


Рисунок 4.7 – Побудова моделі Марковиця

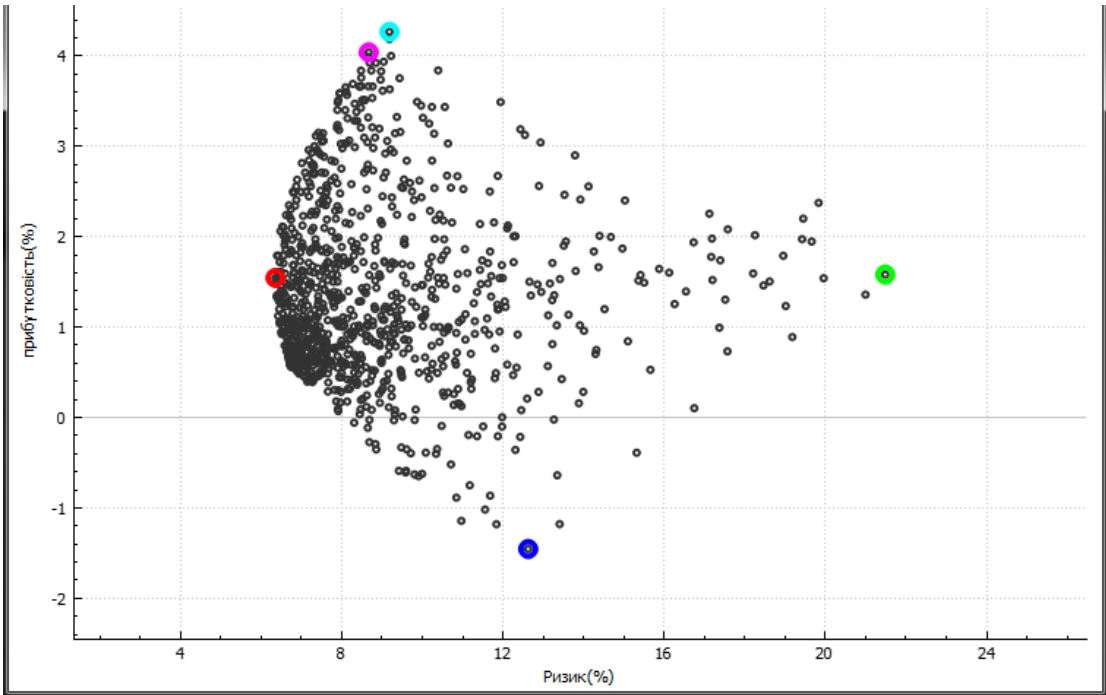


Рисунок 4.8 – Відображення моделі Марковиця у графічному вигляді

Також в таблиці буде відображено значення прибутку, ризику та долей усіх компаній для 5 портфель: з максимальним прибутком, з мінімальним прибутком, з максимальним ризиком, з мінімальним ризиком та портфель з найвищим коефіцієнтом Шарпа. Даний вивід дає досить наочний огляд та аналіз вхідних даних, а також дозволяє обрати бажаний набір активів. З попереднього рисунку можна побачити, що екстремуми моделі та коефіцієнт Шарпа відображені відповідним кольором. У вкладці «Stocks» (рис. 4.9) зображено графіки цін на акції обраних компаній у обраних часових межах.

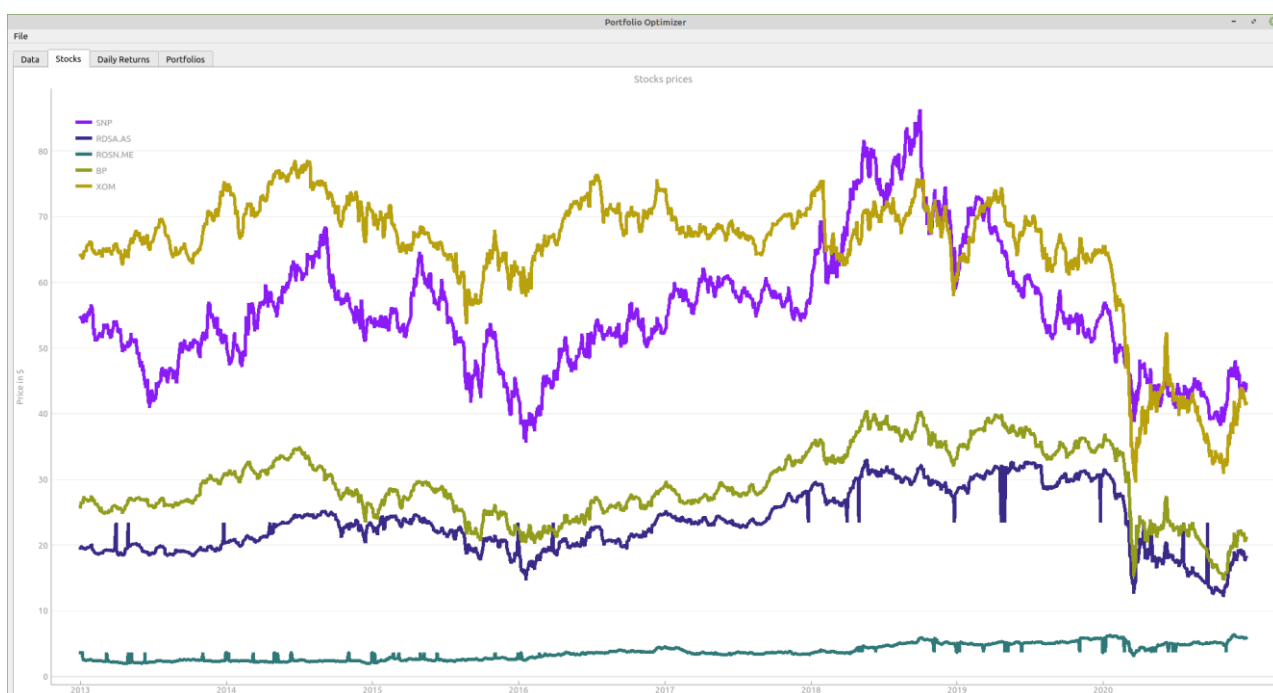


Рисунок 4.9 – Динаміка цін та акцій

На вкладці «Daily Returns» на рисунку 4.10 зображено графіки щоденної віддачі цін акцій обраних компаній. Для розрахування коефіцієнту Шарпа та мінімальної волатильності необхідно на вкладці «Portfolios» на рисунку 4.11 ввести Minimum Risk Rate та кількість портфельів у відповідні поля. Після натискання кнопки Generate у лівій частині екрану з'явиться графік з згенерованими портфелями, ефективним рубежем та з позначеними оптимальними портфелями, знайденими за допомогою різних методів

оптимізації. У правій частині екрану з'являться дві кругові діаграми з відсотковим відношенням акцій у оптимальних портфелях.

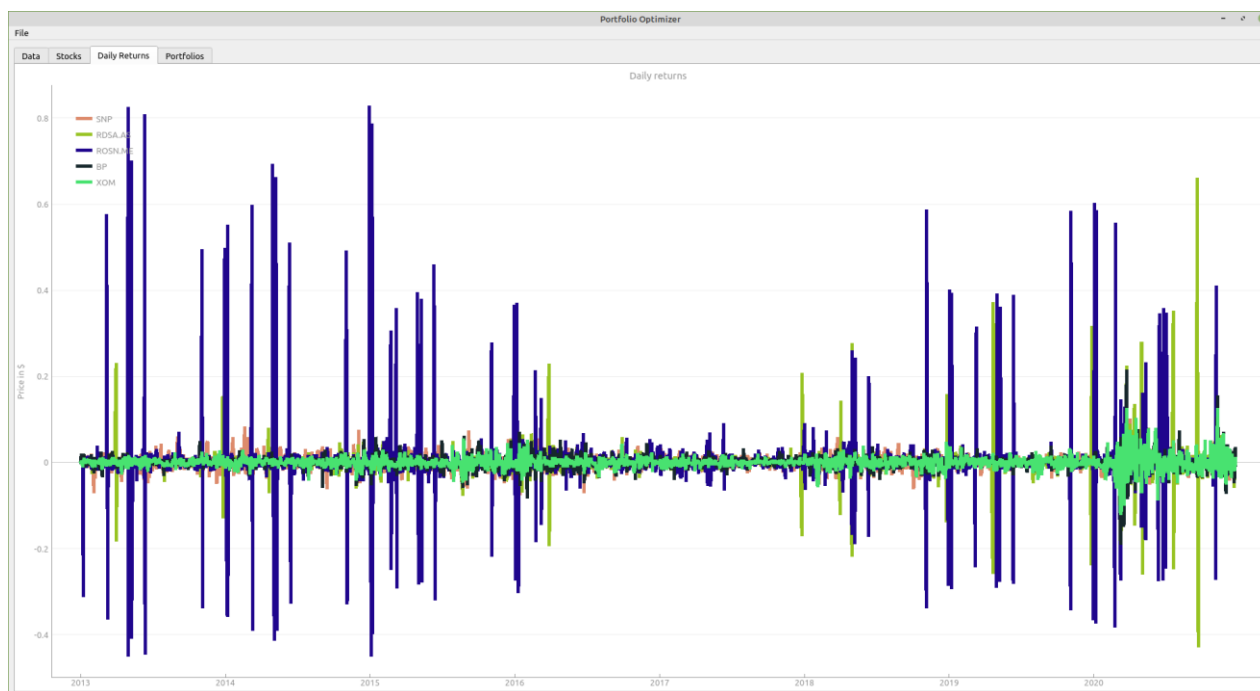


Рисунок 4.10 – Графіки щоденної віддачі цін акцій

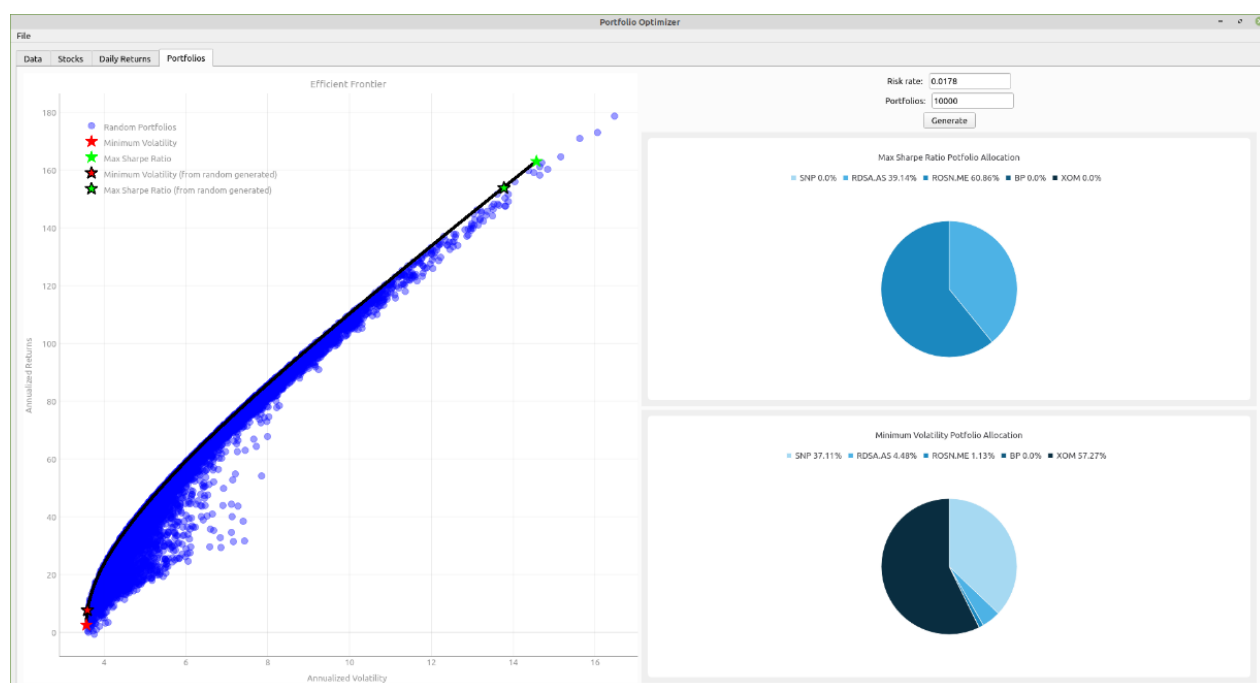


Рисунок 4.11 – Розрахунок результуючих параметрів згідно згенерованих портфелів

4.4 Висновки до розділу

В даному розділі представлено програмну реалізацію вирішення задачі реалізації процесів енергозбереження на металургійних підприємствах із використанням портфельної оптимізації та при застосуванні мови програмування C++ та фреймворку QT. QT – це крос-платформовий інструментарій розробки програмного забезпечення, що дозволяє запускати написане за його допомогою програмне забезпечення на більшості сучасних операційних систем, просто компілюючи текст програми для кожної операційної системи без зміни сирцевого коду.

Вихідними даними до програмного продукту – є дискретні масиви акцій компаній, що записані у цифрову вигляді у файлі на диску комп'ютеру. Програма дозволяє в табличному та графічному виглядах представити результати портфельної оптимізації, – для цього передбачені наступні пункти меню програми: «Графік цін акцій» та «Графік динаміки» (буде відображено відповідно вартості акцій певної компанії та динаміку зміни вартості), «Ціни активів» та «Динаміка цін» відображають в таблиці ціни акцій за певні періоди та їх зміну за певний період відповідно, «Сформувати портфелі» виконує побудову моделі Марковиця і відображення її на графіку прибуток/ризик на цьому вікні; також є можливість переглянути матрицю коваріацій, динаміку цін та акцій компанії, отримати графіки щоденної віддачі цін акцій, та після всіх вищезазначених розрахунків отримати комплексну інформацію та рекомендації у вікні «Розрахунок результуючих параметрів згідно згенерованих портфелів».

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Експериментальне дослідження програмного продукту

Відповідно до завдання було проаналізовано дані акцій компаній за певний період. Взято в умовних одиницях вартість закриття кожного місяця впродовж трьох років, 2017-2019 роки (рис. 5.1) [6].

	Apple	Microsoft	Intel	IBM	HP	Coca-Cola
2017 01 01	121.3	63.97	36.83	174.53	15.06	41.58
2017 02 01	136.98	65.87	36.195	179.83	17.37	41.955
2017 03 01	143.69	68.46	36.08	174.17	17.89	42.44
2017 04 01	143.65	69.86	36.14	160.29	18.82	43.14
2017 05 01	152.78	68.94	36.1	152.62	18.77	45.46
2017 06 01	144.04	72.72	33.74	153.89	17.47	44.855
2017 07 01	148.78	74.78	35.48	144.65	19.105	45.84
2017 08 01	164.05	74.44	35.07	143.04	19.085	45.53
2017 09 01	154.06	83.18	38.08	145.01	19.96	45
2017 10 01	169.04	84.09	45.49	154.09	21.54	45.98
2017 11 01	171.71	85.59	44.83	153.95	21.45	45.77
2017 12 01	169.27	95.2	46.17	153.45	21.01	45.95
2018 01 01	167.47	93.86	48.17	163.74	23.33	47.61
2018 02 01	178.1	91.27	49.29	155.81	23.39	43.22
2018 03 01	168.05	93.53	51.81	153.46	21.92	43.45
2018 04 01	165.27	98.86	51.63	144.98	21.5	43.21
2018 05 01	186.87	98.65	55.21	141.29	22.03	43.005
2018 06 01	182.91	106.1	49.71	139.72	22.69	43.86
2018 07 01	190.61	112.27	48.11	144.875	23.07	46.63
2018 08 01	227.51	111.18	48.13	148.18	21.55	41.58

Рисунок 5.1 - Дані акцій компаній

Також, на рис. 5.2 представлено дані відносної зміни вартості акцій у відсотках до попереднього періоду.

	Apple	Microsoft	Intel	IBM	HP	Coca-Cola
2017 01 01	121.3	63.97	36.83	174.53	15.06	41.58
2017 02 01	136.98	65.87	36.195	179.83	17.37	41.955
2017 03 01	143.69	68.46	36.08	174.17	17.89	42.44
2017 04 01	143.65	69.86	36.14	160.29	18.82	43.14
2017 05 01	152.78	68.94	36.1	152.62	18.77	45.46
2017 06 01	144.04	72.72	33.74	153.89	17.47	44.855
2017 07 01	148.78	74.78	35.48	144.65	19.105	45.84
2017 08 01	164.05	74.44	35.07	143.04	19.085	45.53
2017 09 01	154.06	83.18	38.08	145.01	19.96	45
2017 10 01	169.04	84.09	45.49	154.09	21.54	45.98
2017 11 01	171.71	85.59	44.83	153.95	21.45	45.77
2017 12 01	169.27	95.2	46.17	153.45	21.01	45.95
2018 01 01	167.47	93.86	48.17	163.74	23.33	47.61
2018 02 01	178.1	91.27	49.29	155.81	23.39	43.22
2018 03 01	168.05	93.53	51.81	153.46	21.92	43.45
2018 04 01	165.27	98.86	51.63	144.98	21.5	43.21
2018 05 01	186.87	98.65	55.21	141.29	22.03	43.005
2018 06 01	182.91	106.1	49.71	139.72	22.69	43.86
2018 07 01	190.61	112.27	48.11	144.875	23.07	46.63

Рисунок 5.2 – Відносні зміни вартості акцій

У роботі наведено графіки зміни вартості акцій відповідних компаній. Як видно з вищенаведених графіків на рис. 5.3-5.7, акції кожної з компаній змінюються нерівномірно, є піки, зростання та падіння.

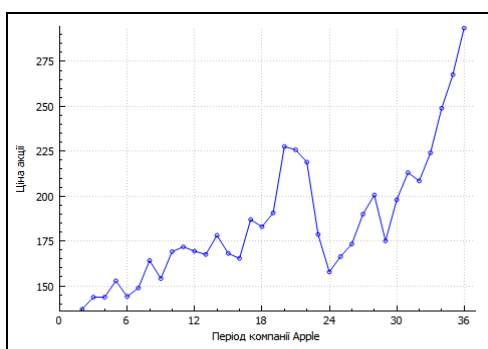


Рисунок 5.3 – Вартість акцій компанії Apple

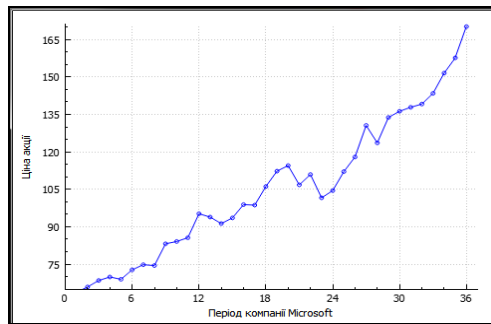


Рисунок 5.4 - Вартість акцій компанії Microsoft

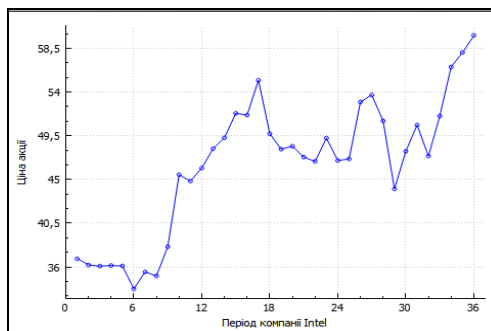


Рисунок 5.5 - Вартість акцій компанії Intel

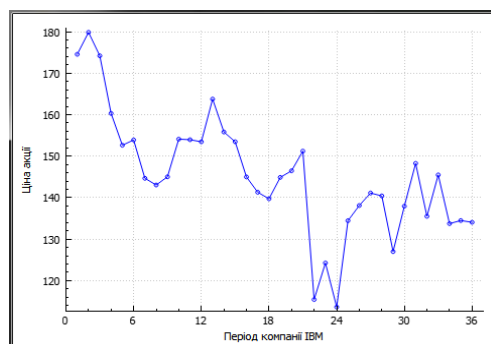


Рисунок 5.6 - Вартість акцій компанії IBM

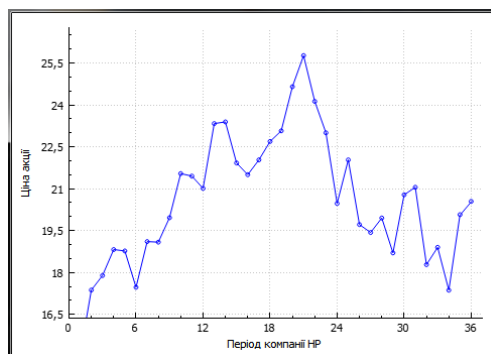


Рисунок 5.7 - Вартість акцій компанії HP

Як видно з вищенаведеного графіку на рис. 5.8, акція компанії змінюється нерівномірно, є піки, зростання та падіння. Також, на рисунках 5.9-5.14 представлено графічні дані відносної зміни вартості акцій компаній у відсотках до попереднього періоду.

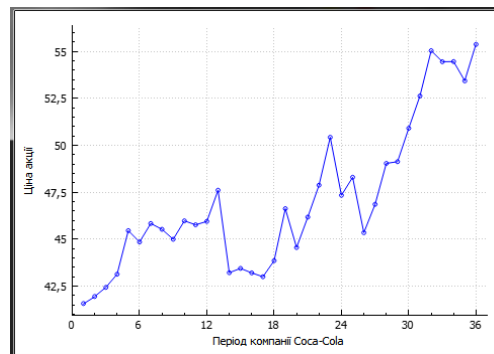


Рисунок 5.8 - Вартість акцій компанії Coca-Cola

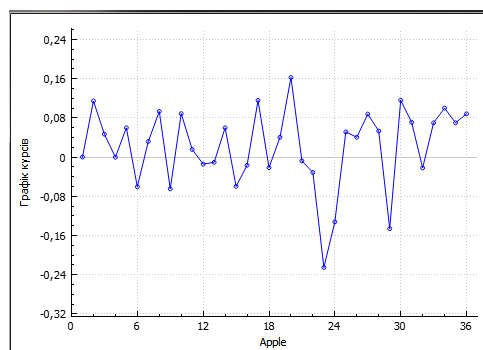


Рисунок 5.9 – Відносна зміна вартості акцій компанії Apple

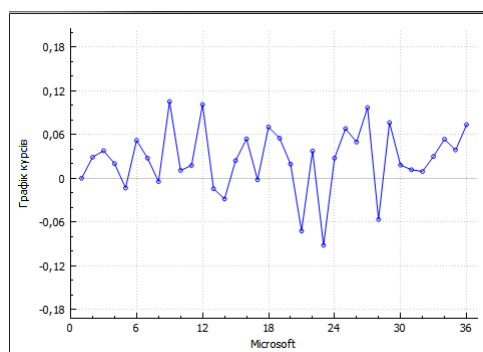


Рисунок 5.10 - Відносна зміна вартості акцій компанії Microsoft

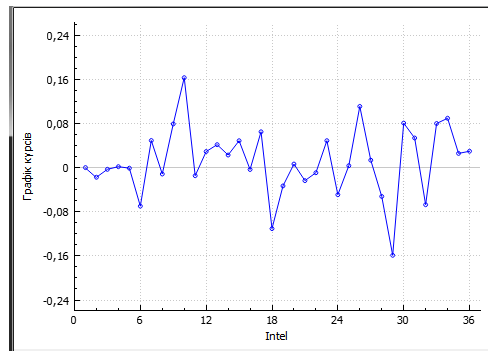


Рисунок 5.11 - Відносна зміна вартості акцій компанії Intel

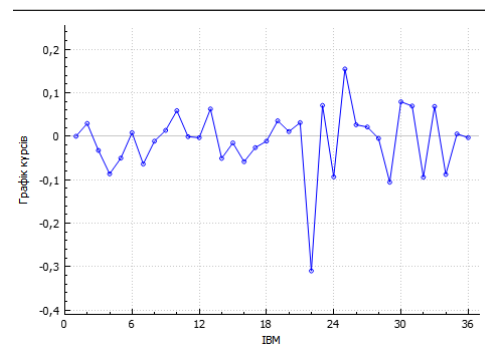


Рисунок 5.12 - Відносна зміна вартості акцій компанії IBM

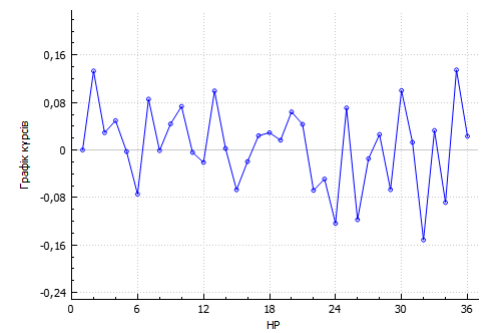


Рисунок 5.13 - Відносна зміна вартості акцій компанії HP

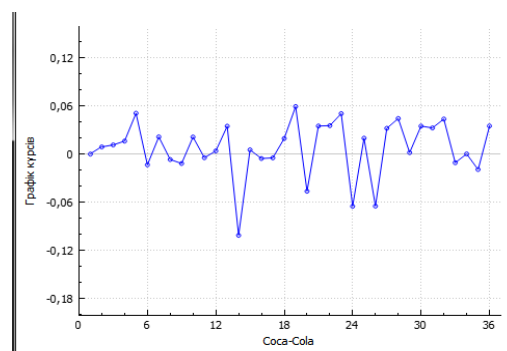


Рисунок 5.14 - Відносна зміна вартості акцій компанії Coca-Cola

Як видно з попередніх графіків, динаміка зміни цін мало дає корисних даних, адже ціни коливалися упродовж усього періоду спостереження.

На рисунку 5.15 позначено матрицю коваріацій, на головній діагоналі знаходяться дисперсії компанії саму на себе. Дана матриця буде використана при формуванні портфелів. На рисунку 5.16 зображено прибуток, його дисперсія та відхилення усіх компаній. Найкориснішими даними будуть значення прибутку та його відхилення.

	Apple	Microsoft	Intel	IBM	HP	Coca-Cola
Apple	0.655398%	0.0275781%	0.187471%	0.144184%	0.279764%	-0.014178%
Microsoft	0.0275781%	0.195971%	-0.0164322%	-0.0315274%	-0.0233723%	-0.0209887%
Intel	0.187471%	-0.0164322%	0.384614%	0.15802%	0.102585%	-0.0210575%
IBM	0.144184%	-0.0315274%	0.15802%	0.59428%	0.249854%	0.0252194%
HP	0.279764%	-0.0233723%	0.102585%	0.249854%	0.485149%	0.0433518%
Coca-Cola	-0.014178%	-0.0209887%	-0.0210575%	0.0252194%	0.0433518%	0.122764%

Рисунок 5.15 – Матриця коваріацій

	Apple	Microsoft	Intel	IBM	HP	Coca-Cola
Прибуток	2.17271%	2.65774%	1.18363%	-1.03512%	0.64151%	0.754944%
Дисперсія прибутку	0.655398%	0.195971%	0.384614%	0.59428%	0.485149%	0.122764%
Відхилення прибутку	8.09567%	4.42686%	6.20173%	7.70895%	6.96526%	3.50377%

Рисунок 5.16 – Розрахунок прибутку

Згідно тестовим даним, найкориснішими даними будуть значення прибутку та його відхилення. Аналізуючи дані з попереднього рисунку, можна побачити, що найбільші значення прибутку має компанія Microsoft, до того ж значення ризику майже найменше серед усіх компаній. Apple має наступне значення прибутку, проте відлякує відносно великим значенням ризику. Компанія IBM взагалі має від'ємне значення прибутку, до того ж також досить велике значення ризику. Зазвичай це означає виключення компанії зі списку

учасників портфелю, проте в якості прикладу її значення будуть використані в розрахунках. Найменше ж значення ризику має компанія Coca-Cola, проте вона має й невелике значення прибутку. Інші дві компанії не дуже виділяються серед інших, тому й не потребують детального опису. Після проведення вищенаведених розрахунків, необхідно сформувати модель Марковиця, та на її основі підібрати найбільш оптимальні значення портфелів. На рис. 5.17 зазначено модель Марковиця для 1000 екземплярів портфелів.

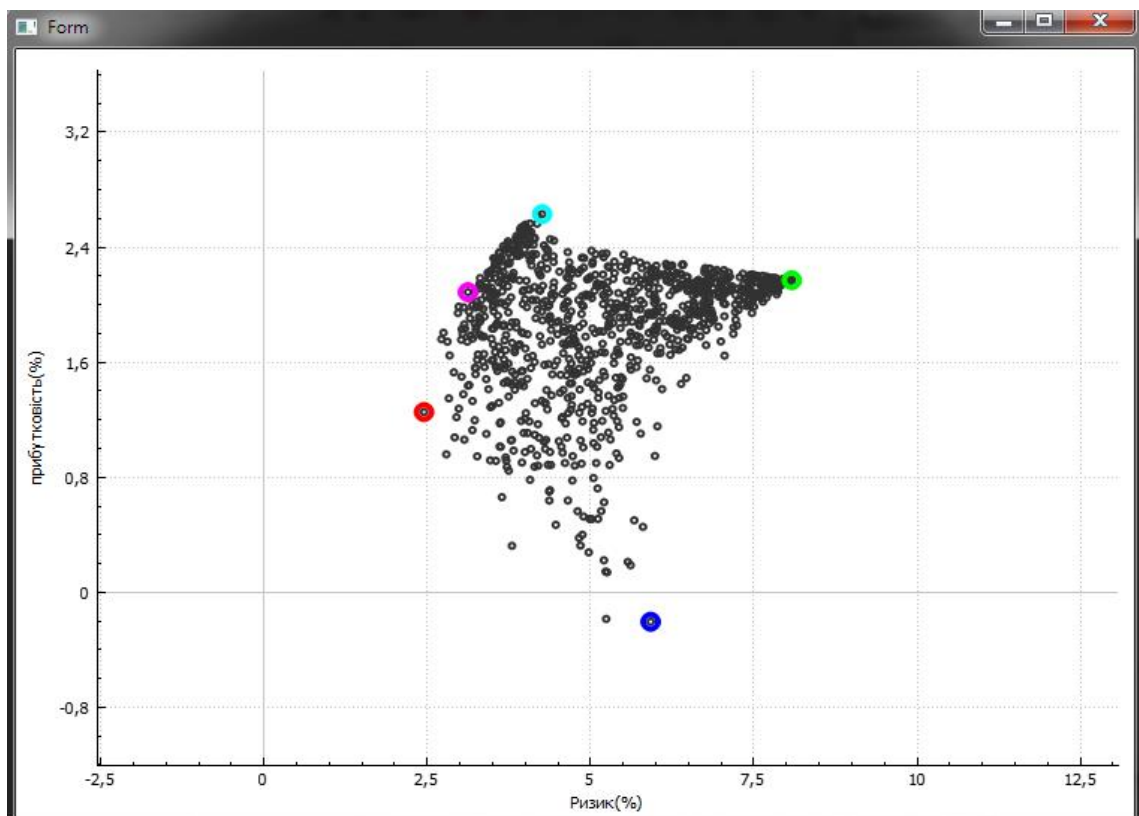


Рисунок 5.17 – Сформована модель Марковиця

Для більш детальної інформації бажано скористатися табличними даними, що наведено на рис. 5.18.

	Мінімальний ризик	Максимальний ризик	Мінімальний прибуток	Максимальний прибуток	Максимальний коефіцієнт Шарпа
Прибуток	1.25305%	2.16976%	-0.205329%	2.62949%	2.08682%
Ризик	2.45014%	8.08276%	5.92092%	4.25941%	3.12281%
Apple	0.047%	0.998%	0.074%	0.04%	0.052%
Microsoft	0.197%	0%	0.089%	0.954%	0.657%
Intel	0.259%	0%	0.031%	0.006%	0.148%
IBM	0.029%	0%	0.691%	0%	0.029%
HP	0.022%	0.001%	0.096%	0%	0.031%
Coca-Cola	0.446%	0.001%	0.019%	0%	0.083%

Рисунок 5.18 – Табличні дані сформованої моделі

Дана таблиця виображає 4 екстремуми та найоптимальніші варіанти портфель. Як видно, для портфелю з мінімальним ризиком, необхідно обрати близько 0.45% акцій Coca-Cola, та розділити більшість акцій між компаніями Microsoft та Intel. Проте при цьому прибуток складе всього близько 1.25%, а ризик всього 2.45%. Для тих, хто хоче ризикнути, а саме з ризиком 8%, можна обрати портфель, в якому майже повністю буде включено акції компанії Apple. При цьому й прибуток буде більше, а саме близько 2.2%. Для максимального прибутку необхідно покласти у портфель більше акцій компанії Microsoft. При цьому прибуток буде ще більшим, а саме 2,62%. Ризик буде сягати близько 4.26%. Окрім цих варіантів, найдоцільнішим буде використати коефіцієнт Шарпа, при цьому значення ризику та прибутку будуть найоптимальнішими, а саме 3.2% та 2.09% відповідно. Задля цього необхідно акції розподілити таким чином, як зображено на рис. 5.17. Невдалим буде варіант, при якому в портфель буде вкладено найбільше акцій компанії IBM. Хоча й значення ризику середне серед усіх представлених вище варіантів, проте значення прибутку буде від’ємним. Тому такий варіант не є доцільним.

5.2 Програмне тестування. Unit-tests

Для тестування програмного продукту виконано модульне тестування. Модульні тести, або unit-тести, розробляють в процесі розробки та їх зазвичай (unit-тести) застосовують для того, щоб упевнитися, що код відповідає вимогам архітектури та має очікувану поведінку.

Метою модульного тестування є ізоляція кожної частини програми та впевненість у тому, що кожна окрема частина є коректною. Модульний тест забезпечує жорсткий «контракт», за яким має працювати тестований код. Як результат, це надає деякі переваги. Модульне тестування допомагає знайти помилки раніше в циклі розробки ПЗ, що робить розробку дешевшою та швидшою.

Переваги модульного тестування:

- легкий рефакторинг – модульне тестування дозволяє програмісту, коли він буде змінювати код (проводити рефакторинг) бути впевненим, що модуль працює вірно. Оскільки модульне тестування вимагає написання тестів для всіх функцій та методів у програмі, помилки швидко локалізуються та виправляються;
- спрощене інтеграційне тестування – модульне тестування може бути застосоване в інтеграційному тестуванні: тестування окремих модулів та сукупності цих модулів робить інтеграційне тестування легшим. Однак модульне тестування знизу вгору не є інтеграційним тестуванням. Інтеграція з зовнішніми модулями має включатися до інтеграційних тестів, а не до модульних;
- документування – модульні тести являють собою специфічний вид документації до системи. Розробники можуть подивитися на модульний тест, щоб дізнатися про функції, що виконує модуль, та як його застосовувати;
- дизайн – кожен тест визначає потрібні класи та методи, їх очікувану поведінку;

– відокремлення інтерфейсу від реалізації – призначення модульного тесту – це тестування єдиного програмного модулю.

5.3 Висновки до розділу

В даному розділі розроблений програмний продукт був застосований для тестових даних акцій компаній. В розділі наведено графіки зміни вартості акцій, представлено графічні дані відносної зміни вартості акцій у відсотках до попереднього періоду, позначено матрицю коваріацій, зображено прибуток, його дисперсія та відхилення усіх компаній. Програмний продукт розроблено із використанням мови програмування C++ та фреймворку QT. Виконано модульне тестування за допомогою unit-tests та бібліотеки тестів Boots.

ВИСНОВКИ

В дипломній роботі виконано створення програмної системи для автоматизації процесів енергозбереження на металургійних підприємствах. Проаналізовані ризики при інвестуванні в об'єкт господарської власності, а також показана залежність між прибутком та ризиками. Виконано аналіз застосування сучасних методів портфельної оптимізації. Враховуючи основні принципи інвестиційної теорії, задачею портфельного інвестування є формування інвестиційного портфелю з відомими частками певних активів для максимізації віддачі та мінімізації ризику. Для вирішення даної проблеми було обрано методику Г. Марковіца.

В роботі виконано програмну реалізацію вирішення задачі аналізу процесів енергозбереження на металургійних підприємствах із використанням портфельної оптимізації та при застосуванні мови програмування C++ та фреймворку QT. Програма дозволяє в табличному та графічному виглядах представити результати портфельної оптимізації: відобразити динаміку вартості акцій певної компанії та динаміку зміни вартості, сформувати портфелі та виконати побудову моделі Марковиця і відображення її на графіку прибуток/ризик, переглянути матрицю коваріацій, отримати комплексну інформацію та відповідні рекомендації щодо розрахункових параметрів.

Виконано модульне тестування за допомогою Unit-tests та бібліотеки тестів Boots. Роботу програми також протестовано із використанням експериментальних тестових даних. Розроблений програмний продукт є повністю функціональним та готовим до експлуатації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мойсеєнко, І. П. Управління фінансово-економічною безпекою підприємства / І. П. Мойсеєнко, О. М. Марченко. – Львів : ЛьвДУВС, 2011. – 380 с.
2. Семенютіна, Т. В. Економічні ризики, небезпеки, загрози: сутність та взаємозв'язок / Т. В. Семенютіна. – Київ : Економічний простір. – 2012. – № 68. – С. 106–113.
3. Рудніченко, Є. М. Загроза, ризик, небезпека: сутність та взаємозв'язок із системою економічної безпеки підприємства / Є. М. Рудніченко. – Київ : Економіка. Менеджмент. Підприємництво. – 2018. – № 25 (1). – С. 188–195.
4. Вітлінський, В. В. Аналіз, моделювання та управління економічним ризиком : навч.-метод. посібник / В. В. Вітлінський, П. І. Верченко. – Київ : КНЕУ, 2000. – 292 с.
5. Камінський, А. Б. Моделювання фінансових ризиків: монографія / А. Б. Камінський. – Київ : ВПЦ «Київський університет», 2006. – 304 с.
6. Портфельна теорія Марковіца. Методика формування інвестиційного портфеля [Електрон. ресурс]. – 2021. – Режим доступу: <https://uk.campwaltblog.com/4291261-portfolio-theory-markowitz-the-method-of-forming-the-investment-portfolio>.
7. Markowitz H. M. Foundations of Portfolio Theory / H. M. Markowitz // Journal of Finance. – 1991. – Vol. 46. – Issue 2. – P. 469-77.
8. Піддубний, І. Д. Підходи до формування оптимального портфеля цінних паперів [Електрон. ресурс] / І. Д. Піддубний. – 2019. – Режим доступу: http://ekmair.ukma.edu.ua/bitstream/handle/123456789/15502/Piddubnyi_Pidkhody_do_formuvannia_optymalnoho.pdf.
9. Практичне керівництво створення UML-діаграм [Електрон. ресурс]. – Режим доступу: https://flexberry.github.io/ru/gpg_practical-guides-uml.html.
10. Unified Modeling Language tutorial [Electronic resource]. – Access mode: <https://www.tutorialspoint.com/uml/index>.

ДОДАТОК А
Текст програми

```

//-----
// drawplot.h
//-----

#ifndef DRAWPLOT_H
#define DRAWPLOT_H

#include <QWidget>

namespace Ui { class drawPlot; }

class drawPlot : public QWidget{
    Q_OBJECT
public:
    explicit drawPlot(QWidget *parent = nullptr);
    ~drawPlot();
    void printPlot(QVector<double> profits,QVector<double>risks);
private:
    Ui::drawPlot *ui;
};

#endif // DRAWPLOT_H
//-----

// mainwindow.h
//-----

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include "drawplot.h"

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow{
    Q_OBJECT
public:

```

```

MainWindow(QWidget *parent = nullptr);
~MainWindow();

private slots:
    void on_pushButton_clicked();
    void on_pushButton_2_clicked();
    void on_pushButton_3_clicked();
    void on_pushButton_4_clicked();
    void on_pushButton_5_clicked();
    void on_pushButton_6_clicked();
    void on_pushButton_7_clicked();

private:
    Ui::MainWindow *ui;
    drawPlot printPlot;
};

#endif // MAINWINDOW_H

//-----
// main.cpp
//-----

#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[]){
QApplication a(argc, argv);  MainWindow w;  w.show();  return a.exec(); }

//-----
// drawplot. cpp
#include "drawplot.h"
#include "ui_drawplot.h"

drawPlot::drawPlot(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::drawPlot)
{
    ui->setupUi(this); }

```

```

drawPlot::~drawPlot() { delete ui; }

void drawPlot::printPlot(QVector<double> profits,QVector<double>risks){
    double minY = std::numeric_limits<double>::max(), maxY =
std::numeric_limits<double>::min();

    double minX = minY, maxX = maxY;
    QVector<QVector<double>>minRisks = {{0},{0}};
    QVector<QVector<double>>maxRisks = {{0},{0}};
    QVector<QVector<double>>minProfit = {{0},{0}};
    QVector<QVector<double>>maxProfit = {{0},{0}};
    QVector<QVector<double>>maxCoefSharp = {{0},{0}};
    double maxCharp = maxY;
    for (int i=0; i<risks.size(); i++) {
        if (profits[i]<minY){
minY = profits[i]; minProfit[0][0] = risks[i]; minProfit[1][0] = profits[i]; }
        if (profits[i]>maxY){
maxY = profits[i]; maxProfit[0][0] = risks[i]; maxProfit[1][0] = profits[i]; }
        if (risks[i]<minX){
minX = risks[i]; minRisks[0][0] = risks[i]; minRisks[1][0] = profits[i]; }
        if (risks[i]>maxX){
maxX = risks[i]; maxRisks[0][0] = risks[i]; maxRisks[1][0] = profits[i]; }
        if((profits[i]/risks[i])>maxCharp){
maxCharp = (profits[i]/risks[i]); maxCoefSharp[0][0] = risks[i];
maxCoefSharp[1][0] = profits[i]; } }

    QPen penPlot;
    penPlot.setWidth(2);
    ui->widget->clearGraphs();
    ui->widget->addGraph();
    ui->widget->graph(0)->setData(risks, profits);
    penPlot.setColor(QColor(50, 50, 50, 255));
    ui->widget->graph(0)->setPen(penPlot);

```

```

    ui->widget->graph(0)->setLineStyle(QCPGraph::lsNone);
    ui->widget->graph(0)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 4));
    penPlot.setWidth(4);
    ui->widget->addGraph();
    penPlot.setColor(QColor(250, 0, 0, 255));
    ui->widget->graph(1)->setPen(penPlot);
    ui->widget->graph(1)->setData(minRisks[0], minRisks[1]);
    ui->widget->graph(1)->setLineStyle(QCPGraph::lsNone);
    ui->widget->graph(1)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 10));
    ui->widget->addGraph();
    penPlot.setColor(QColor(0, 250, 0, 255));
    ui->widget->graph(2)->setPen(penPlot);
    ui->widget->graph(2)->setData(maxRisks[0], maxRisks[1]);
    ui->widget->graph(2)->setLineStyle(QCPGraph::lsNone);
    ui->widget->graph(2)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 10));
    ui->widget->addGraph();
    penPlot.setColor(QColor(0, 0, 250, 255));
    ui->widget->graph(3)->setPen(penPlot);
    ui->widget->graph(3)->setData(minProfit[0], minProfit[1]);
    ui->widget->graph(3)->setLineStyle(QCPGraph::lsNone);
    ui->widget->graph(3)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 10));
    ui->widget->addGraph();
    penPlot.setColor(QColor(0, 250, 250, 255));
    ui->widget->graph(4)->setPen(penPlot);
    ui->widget->graph(4)->setData(maxProfit[0], maxProfit[1]);
    ui->widget->graph(4)->setLineStyle(QCPGraph::lsNone);

```

```

        ui->widget->graph(4)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 10));
        ui->widget->addGraph();
        penPlot.setColor(QColor(250, 0, 250, 255));
        ui->widget->graph(5)->setPen(penPlot);
        ui->widget->graph(5)->setData(maxCoefSharp[0], maxCoefSharp[1]);
        ui->widget->graph(5)->setLineStyle(QCPGraph::lsNone);
        ui->widget->graph(5)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 10));
        ui->widget->xAxis->setLabel("Ризик(%)");
        ui->widget->yAxis->setLabel("прибутковість(%)");
        ui->widget->xAxis->setRange(minX-5, maxX+5);
        ui->widget->yAxis->setRange(minY-1, maxY+1);
        ui->widget->replot();
    }
//-----
//mainwindow.cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"
 QVector<QVector<double>> shares, dynamics, covMatrix;
 QVector<QString> periods, companies;
 QVector<double> expProfits, expVarProfits, expStDevProfits;
 int plotMode = -1;
 MainWindow::MainWindow(QWidget *parent)
 : QMainWindow(parent) , ui(new Ui::MainWindow){
    periods, companies; QVector<double> date; shares.append(date);
    for(int i = 0; i < shares.size(); i++){
        QVector<double> dynamic;
        dynamic.append(0);
        for(int j = 1; j < shares[0].size(); j++){

```

```

        dynamic.append((shares[i][j] - shares[i][j-1])/shares[i][j]);    }
    dynamics.append(dynamic);    }
for(int i = 0 ; i < dynamics.size(); i++){
    double expectedValue = 0;
    for(int j = 1; j < dynamics[0].size(); j++){
        expectedValue+=dynamics[i][j];    }
    expectedValue = expectedValue / (dynamics[0].size() - 1);
    expProfits.append(expectedValue);    }
for(int i = 0 ; i < dynamics.size(); i++){
    double expectedValue = 0;
    for(int j = 1; j < dynamics[0].size(); j++){
        expectedValue+=pow(dynamics[i][j] - expProfits[i],2);    }
    expectedValue = expectedValue/ (dynamics[0].size() - 1);
    expVarProfits.append(expectedValue);
    expStDevProfits.append(sqrt(expectedValue));    }
for(int i = 0 ; i < shares.size();i++){
    QVector<double> line;
    for(int j = 0 ; j < shares.size();j++){
        line.append(0);    }
    covMatrix.append(line);    }
for(int i = 0 ; i < shares.size();i++){
    for(int j = i ; j < shares.size();j++){
        double expTemp = 0;
        for(int k = 0 ; k < shares[0].size();k++){
            expTemp+= dynamics[i][k] * dynamics[j][k];    }
        expTemp = expTemp / (shares[0].size() -1);
        covMatrix[i][j] = expTemp - (expProfits[i]*expProfits[j]);
        covMatrix[j][i] = covMatrix[i][j];    }    }
ui->setupUi(this);
}

```

```

MainWindow::~MainWindow() { delete ui; }

void MainWindow::on_pushButton_clicked() {
    plotMode++;
    if(plotMode == shares.size()) plotMode = 0;
    QVector<double> xRange;
    for(int i = 0 ; i < shares[0].size(); i++)
        xRange.append(i+1);
    ui->widget->clearGraphs();
    ui->widget->addGraph();
    ui->widget->graph(0)->setData(xRange, shares[plotMode]);
    ui->widget->graph(0)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 4));
    ui->widget->yAxis->setLabel("Ціна акції");
    ui->widget->xAxis->setLabel("Період компанії " +companies[plotMode] );
    ui->widget->xAxis->setRange(xRange[0]-1, xRange[xRange.size()-1]+1);
    double minY = std::numeric_limits<double>::max(), maxY =
std::numeric_limits<double>::min();
    for (int i=1; i<shares[0].size(); i++) {
        if (shares[plotMode][i]<minY) minY = shares[plotMode][i];
        if (shares[plotMode][i]>maxY) maxY = shares[plotMode][i];
    }
    ui->widget->yAxis->setRange(minY-1, maxY+1);
    ui->widget->replot();}

void MainWindow::on_pushButton_2_clicked() {
    QStandardItemModel *model = new QStandardItemModel;
    QStandardItem *item;
    QStringList horizontalHeader, verticalHeader
    for(int i = 0; i < shares.size(); i++){
        horizontalHeader.append(companies[i]);
        for(int j = 0; j < shares[0].size(); j++){
            item = new QStandardItem(QString::number(shares[i][j]));

```

```

        model->setItem(j, i, item);
        if(i ==0) verticalHeader.append( periods[j]);
        //if(i ==0) verticalHeader.append("Період " + QString::number(j+1));
    } }

model->setHorizontalHeaderLabels(horizontalHeader);
model->setVerticalHeaderLabels(verticalHeader);
ui->tableView->setModel(model);
ui->tableView->resizeRowsToContents();
ui->tableView->resizeColumnsToContents();}

void MainWindow::on_pushButton_3_clicked(){
    QStandardItemModel *model = new QStandardItemModel;
    QStandardItem *item;
    QStringList horizontalHeader;
    QStringList verticalHeader;
    for(int i = 0; i < dynamics.size(); i++){
        //horizontalHeader.append("Компанія " + QString::number(i+1));
        horizontalHeader.append(companies[i]);
        for(int j = 0; j < dynamics[0].size(); j++){
            item = new QStandardItem(QString::number(dynamics[i][j]*100) + "%");
            model->setItem(j, i, item);
            if(i ==0) verticalHeader.append( periods[j]);    } }
    model->setHorizontalHeaderLabels(horizontalHeader);
    model->setVerticalHeaderLabels(verticalHeader);
    ui->tableView->setModel(model);
    ui->tableView->resizeRowsToContents();
    ui->tableView->resizeColumnsToContents();}

void MainWindow::on_pushButton_4_clicked(){
    QStandardItemModel *model = new QStandardItemModel;
    QStandardItem *item;
    QStringList horizontalHeader;

```

```

QStringList verticalHeader;
verticalHeader.append("Прибуток");
verticalHeader.append("Дисперсія\nприбутку");
verticalHeader.append("Відхилення\nприбутку");
for(int i = 0; i < dynamics.size(); i++){
    horizontalHeader.append(companies[i]);
    item = new QStandardItem(QString::number(expProfits[i] *100) + "%");
    model->setItem(0, i, item);
    item = new QStandardItem(QString::number(expVarProfits[i]*100) + "%");
    model->setItem(1, i, item);
    item = new QStandardItem(QString::number(expStDevProfits[i]*100) + "%");
    model->setItem(2, i, item);
}
model->setHorizontalHeaderLabels(horizontalHeader);
model->setVerticalHeaderLabels(verticalHeader);
ui->tableView->setModel(model);
ui->tableView->resizeRowsToContents();
ui->tableView->resizeColumnsToContents();
}

void MainWindow::on_pushButton_5_clicked(){
    plotMode++;
    if(plotMode == dynamics.size()) plotMode = 0;
    QVector<double> xRange;
    for(int i = 0 ; i < dynamics[0].size(); i++)
        xRange.append(i+1);
    ui->widget->clearGraphs();
    ui->widget->addGraph();
    ui->widget->graph(0)->setData(xRange, dynamics[plotMode]);
    ui->widget->graph(0)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 4));

```

```

ui->widget->xAxis->setLabel(companies[plotMode]);
ui->widget->yAxis->setLabel("Гпафiк кyпciв");
ui->widget->xAxis->setRange(xRange[0]-1, xRange[xRange.size()-1]+1);
double   minY   =   std::numeric_limits<double>::max(),   maxY   =
std::numeric_limits<double>::min();

for (int i=1; i<dynamics[0].size(); i++)   {
    if (dynamics[plotMode][i]<minY) minY = dynamics[plotMode][i];
    if (dynamics[plotMode][i]>maxY) maxY = dynamics[plotMode][i];
}
ui->widget->yAxis->setRange(minY-0.1, maxY+0.1);
ui->widget->replot(); }

void MainWindow::on_pushButton_6_clicked(){
    QStandardItemModel *model = new QStandardItemModel;
    QStandardItem *item;
    QStringList horizontalHeader;
    QStringList verticalHeader;
    for(int i = 0; i < shares.size(); i++){
        horizontalHeader.append(companies[i]);
        verticalHeader.append(companies[i]);
        for(int j = 0; j < shares.size(); j++){
            item = new QStandardItem(QString::number(covMatrix[i][j] * 100) + "%");
            model->setItem(j, i, item);        }    }
    model->setHorizontalHeaderLabels(horizontalHeader);
    model->setVerticalHeaderLabels(verticalHeader);
    ui->tableView->setModel(model);
    ui->tableView->resizeRowsToContents();
    ui->tableView->resizeColumnsToContents();
}

void MainWindow::on_pushButton_7_clicked() {
    int maxValue = 1000, countExperiments = ui->lineEdit->text().toInt();

```

```

if(countExperiments<=0){ countExperiments = 1000;
    ui->lineEdit->setText("1000");    }
QVector<double> profits, risks;
QVector<QVector<double>> coefProfit;
for(int k = 0; k < countExperiments; k++){
    QVector<double> randCoef;
    double sumCoef = 0;
    randCoef.append((rand() % maxVal) / (maxVal * 1.0));
    sumCoef+=randCoef[0];
    for (int i = 1; i< shares.size()-1;i++ ) {
        double qq = (maxVal - int(maxVal*sumCoef));
        if(qq==0)            randCoef.append(0);
        else                  randCoef.append((rand() % (maxVal -
int(maxVal*sumCoef))) / (maxVal * 1.0));
        sumCoef+=randCoef[i];    }
    randCoef.append(1.0 - sumCoef);
    coefProfit.append(randCoef);
    sumCoef+=0;    }
for(int k = 0; k < countExperiments; k++){
    double sumProfit = 0;
    for (int i = 0; i< shares.size();i++ ){
        sumProfit+= expProfits[i]*coefProfit[k][i];    }
    profits.append(sumProfit*100);    }
for(int k = 0; k < countExperiments; k++){
    double sumRisk = 0;

    for(int i = 0 ; i < covMatrix.size(); i++){
        for (int j = i;j < covMatrix.size() ; j++) {
            if(i == j){
                sumRisk+= covMatrix[i][i] * pow(coefProfit[k][i],2);    }

```

```

        else{
            sumRisk+= 2 * coefProfit[k][i] * coefProfit[k][j] * covMatrix[i][j];
        }
    }

    sumRisk = sqrt(sumRisk);
    risks.append(sumRisk*100);

    double minY = std::numeric_limits<double>::max(), maxY =
std::numeric_limits<double>::min();

    double minX = minY, maxX = maxY;
    QVector<QVector<double>>minRisks = {{0},{0}};
    QVector<QVector<double>>maxRisks = {{0},{0}};
    QVector<QVector<double>>minProfit = {{0},{0}};
    QVector<QVector<double>>maxProfit = {{0},{0}};
    QVector<QVector<double>>maxCoefSharp = {{0},{0}};
    double maxCharp = maxY;
    int iMinR, iMaxR, iMinP, iMaxP, iSharp;
    for (int i=0; i<risks.size(); i++) {
        if (profits[i]<minY){
minY = profits[i]; minProfit[0][0] = risks[i]; minProfit[1][0] = profits[i];
iMinP = i;
        }
        if (profits[i]>maxY){
maxY = profits[i]; maxProfit[0][0] = risks[i]; maxProfit[1][0] = profits[i];
iMaxP = i;
        }
        if (risks[i]<minX){
minX = risks[i]; minRisks[0][0] = risks[i]; minRisks[1][0] = profits[i];
iMinR = i;
        }
        if (risks[i]>maxX){
maxX = risks[i]; maxRisks[0][0] = risks[i]; maxRisks[1][0] = profits[i];
iMaxR = i;
        }
        if((profits[i]/risks[i])>maxCharp){
maxCharp = (profits[i]/risks[i]); maxCoefSharp[0][0] = risks[i];

```

```

        maxCoefSharp[1][0] = profits[i];        iSharp = i;        }    }

QPen penPlot;
penPlot.setWidth(1);
ui->widget->clearGraphs();
ui->widget->addGraph();
ui->widget->graph(0)->setData(risks, profits);
penPlot.setColor(QColor(50, 50, 50, 255));
ui->widget->graph(0)->setPen(penPlot);
ui->widget->graph(0)->setLineStyle(QCPGraph::lsNone);
ui->widget->graph(0)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 2));
    penPlot.setWidth(2);
    ui->widget->addGraph();
    penPlot.setColor(QColor(250, 0, 0, 255));
    ui->widget->graph(1)->setPen(penPlot);
    ui->widget->graph(1)->setData(minRisks[0], minRisks[1]);
    ui->widget->graph(1)->setLineStyle(QCPGraph::lsNone);
    ui->widget->graph(1)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 5));
        ui->widget->addGraph();
        penPlot.setColor(QColor(0, 250, 0, 255));
        ui->widget->graph(2)->setPen(penPlot);
        ui->widget->graph(2)->setData(maxRisks[0], maxRisks[1]);
        ui->widget->graph(2)->setLineStyle(QCPGraph::lsNone);
        ui->widget->graph(2)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 5));
            ui->widget->addGraph();
            penPlot.setColor(QColor(0, 0, 250, 255));
            ui->widget->graph(3)->setPen(penPlot);
            ui->widget->graph(3)->setData(minProfit[0], minProfit[1]);

```

```

ui->widget->graph(3)->setLineStyle(QCPGraph::lsNone);
ui->widget->graph(3)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 5));
ui->widget->addGraph();
penPlot.setColor(QColor(0, 250, 250, 255));
ui->widget->graph(4)->setPen(penPlot);
ui->widget->graph(4)->setData(maxProfit[0], maxProfit[1]);
ui->widget->graph(4)->setLineStyle(QCPGraph::lsNone);
ui->widget->graph(4)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 5));
ui->widget->addGraph();
penPlot.setColor(QColor(250, 0, 250, 255));
ui->widget->graph(5)->setPen(penPlot);
ui->widget->graph(5)->setData(maxCoefSharp[0], maxCoefSharp[1]);
ui->widget->graph(5)->setLineStyle(QCPGraph::lsNone);
ui->widget->graph(5)-
>setScatterStyle(QCPScatterStyle(QCPScatterStyle::ssCircle, 5));
ui->widget->xAxis->setLabel("Ризик(%)");
ui->widget->yAxis->setLabel("Прибутковість(%)");
ui->widget->xAxis->setRange(minX-5, maxX+5);
ui->widget->yAxis->setRange(minY-1, maxY+1);
ui->widget->replot();
QStandardItemModel *model = new QStandardItemModel;
QStandardItem *item;
QStringList horizontalHeader , QStringList verticalHeader;
horizontalHeader.append("Мінімальний\нризик");
horizontalHeader.append("Максимальний\нризик");
horizontalHeader.append("Мінімальнийяп\нприбуток");
horizontalHeader.append("Максимальний\нприбуток");
horizontalHeader.append("Максимальний\нкоефіцієнт\нШарпа");

```

```

verticalHeader.append("Прибыток");
verticalHeader.append("Ризик");
for(int i = 0; i < shares.size(); i++){
    verticalHeader.append(companies[i]);    }
 QVector<int> indexTable;
indexTable.append(iMinR);
indexTable.append(iMaxR);
indexTable.append(iMinP);
indexTable.append(iMaxP);
indexTable.append(iSharp);
for(int k = 0; k < indexTable.size(); k++){
item = new QStandardItem(QString::number(profits[indexTable[k]]) + "%");
    model->setItem(0, k, item);
    item = new QStandardItem(QString::number(risks[indexTable[k]]) + "%");
    model->setItem(1, k, item);
    for(int i = 0 ; i < shares.size(); i++){
item = new QStandardItem(QString::number(coefProfit[indexTable[k]][i]) + "%");
        model->setItem(i+2,k , item);    }    }
model->setHorizontalHeaderLabels(horizontalHeader);
model->setVerticalHeaderLabels(verticalHeader);
ui->tableView->setModel(model);
ui->tableView->resizeRowsToContents();
ui->tableView->resizeColumnsToContents();
printPlot.printPlot(profits, risks);
printPlot.show();
}

```

ДОДАТОК Б
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕСІВ ЕНЕРГОЗБЕРЕЖЕННЯ НА МЕТАЛУРГІЙНИХ ПІДПРИЄМСТВАХ

Виконав студент групи КНТ-210м
Ільонюк Д.Г.

Керівник канд.техн.наук, проф.
Дубровін В.І.

2021

1

Рисунок Б.1– Дослідження та програмна реалізація процесів енергозбереження на металургійних підприємствах

Актуальність

Металургійні підприємства є великими споживачами електричної і теплової енергії, тому вирішення завдання енергозбереження неможливо без розроблення комплексної програми й портфеля проєктів за основними напрямками енергозбереження з обов'язковою координацією їх з програмою розвитку основного виробництва. Гострота вирішення цих питань для металургійної галузі України спричинена необхідністю поліпшення економічної стабільності підприємств, підвищення конкурентоспроможності продукції та зменшення залежності від постачальників енергоресурсів.

Оптимізація інвестиційного портфеля є одним із поширених, типових і значимих фінансових завдань, яке виникає в практиці ресурсного забезпечення, енергозбереження, інвестування. Його рішення дозволяє знайти найбільш ефективний спосіб вкладення інвестором свого капіталу в акції декількох компаній.

Таким чином, розробка програмного додатку для дослідження процесів енергозбереження на металургійних підприємствах є актуальною проблемою, вирішення якої допоможе підприємствам оптимізувати свою роботу.

2

Рисунок Б.2– Актуальність проєкту

Постановка мети роботи

- Об'єкт дослідження – програмні засоби моніторингу процесів енергозбереження на металургійних підприємствах.
- Предмет дослідження – процеси енергозбереження на металургійних підприємствах.
- Мета роботи – створення програмної системи для автоматизації процесів енергозбереження на металургійних підприємствах.
- Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мови програмування C++ та фреймворк QT, персональний комп'ютер з процесором Intel Core i5 під управлінням операційної системи Microsoft Windows 10.

3

Рисунок Б.3 – Мета проєкту, об'єкт та предмет дослідження

Основні принципи портфельної оптимізації

До ключових елементів та стратегій створення ефективного інвестиційного портфелю відносять :

- диверсифікація;
- ребалансування;
- активне управління портфелем;
- пасивне управління портфелем



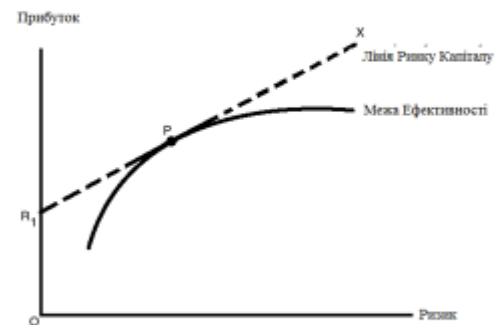
4

Рисунок Б.4 – Основні принципи портфельної оптимізації

Оптимізація інвестиційного портфеля за методом Марковіца

У моделі Марковіца ризик окремо взятого фінансового інструменту розраховується як стандартне відхилення доходів. В основі аналізу лежать наступні критерії:

- пошук ефективних рішень, що неможливо покращити. У цьому випадку будь-яке інше рішення, краще знайдених по одному параметру обов'язково буде гірше за іншим;
- вибір головного критерію (наприклад, прибутковість повинна бути не нижче певної величини) інші використовуючи лише в якості критеріальних обмежень;
- задаючи суперкритерій, який є суперпозицією зазначених двох



5

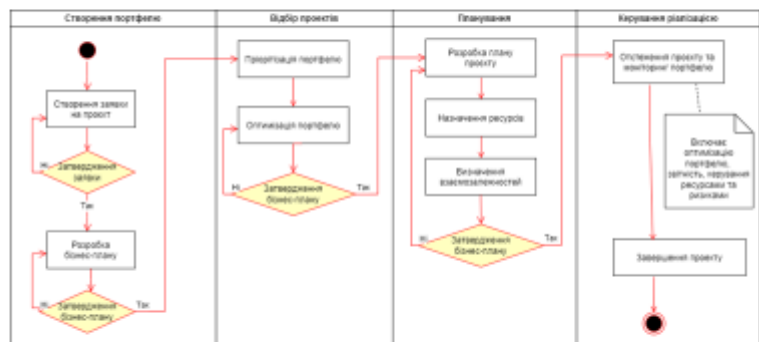
Рисунок Б.5 – Оптимізація інвестиційного портфеля

Проектування програмного додатку

Діаграма варіантів використання:



Діаграма діяльності:



6

Рисунок Б.6 – Проектування програмного додатку

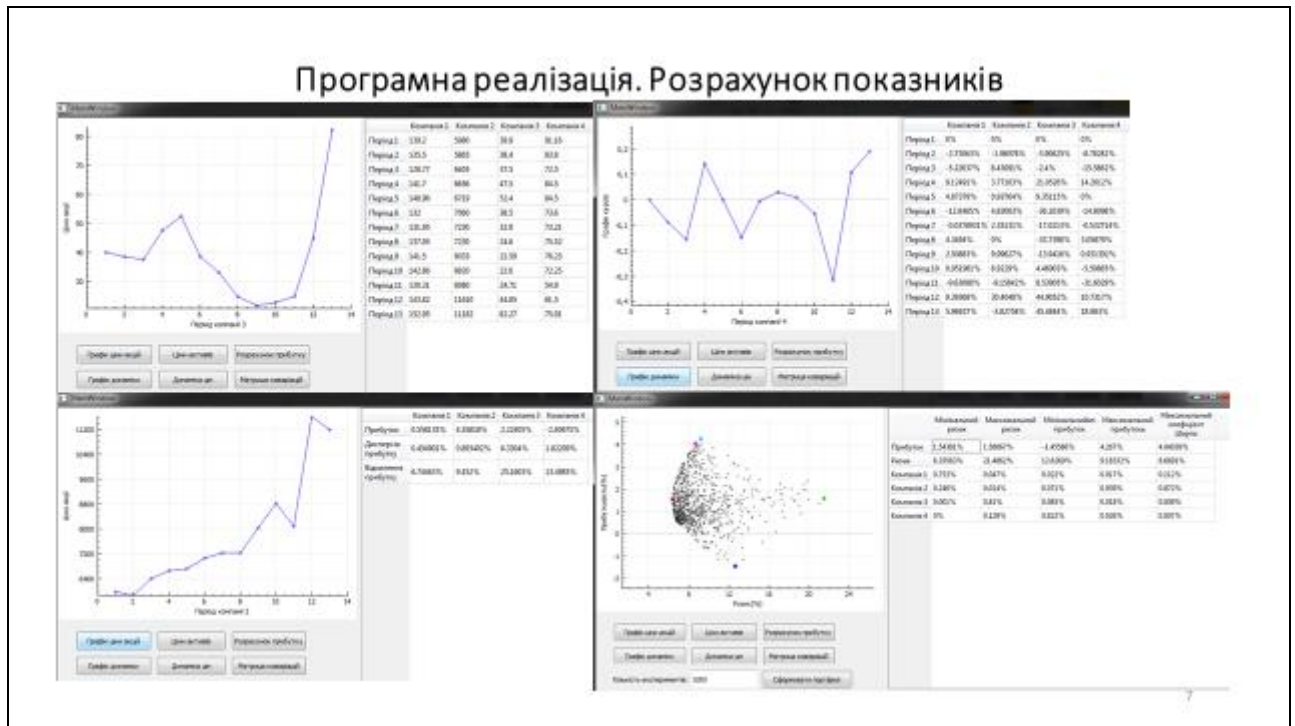


Рисунок Б.7 – Програмна реалізація

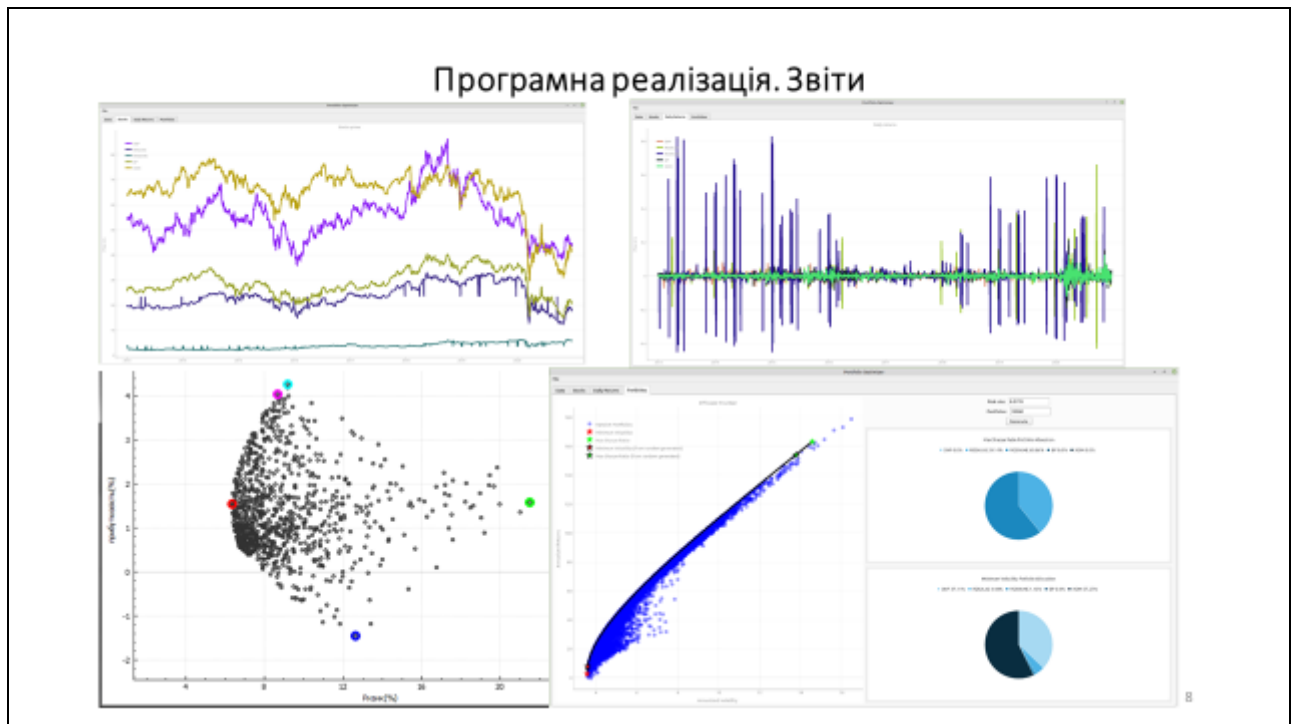


Рисунок Б.8 – Програмна реалізація

Експериментальне дослідження

Матриця коваріацій:

	Apple	Microsoft	Intel	IBM	HP	Coca-Cola
Apple	0.655308%	0.0275781%	0.187471%	0.144184%	0.270764%	-0.014178%
Microsoft	0.0275781%	0.195971%	-0.0164322%	-0.0315274%	-0.0233723%	-0.0209887%
Intel	0.187471%	-0.0164322%	0.384614%	0.13802%	0.102585%	-0.0210575%
IBM	0.144184%	-0.0315274%	0.13802%	0.59428%	0.248854%	0.0252194%
HP	0.270764%	-0.0233723%	0.102585%	0.248854%	0.485149%	0.0433518%
Coca-Cola	-0.014178%	-0.0209887%	-0.0210575%	0.0252194%	0.0433518%	0.122764%

Параметри портфельної оптимізації:

	Мінімальний ризик	Максимальний ризик	Мінімальний прибуток	Максимальний прибуток	Максимальний коефіцієнт Шарпа
Прибуток	1.25305%	2.10970%	-0.205329%	2.62949%	2.09082%
Ризик	2.45014%	8.08276%	5.92092%	4.25941%	3.12281%
Apple	0.047%	0.998%	0.074%	0.04%	0.052%
Microsoft	0.197%	0%	0.089%	0.954%	0.657%
Intel	0.259%	0%	0.031%	0.000%	0.148%
IBM	0.029%	0%	0.691%	0%	0.029%
HP	0.022%	0.001%	0.096%	0%	0.031%
Coca-Cola	0.446%	0.001%	0.019%	0%	0.083%

Розрахунок прибутку:

	Apple	Microsoft	Intel	IBM	HP	Coca-Cola
Прибуток	2.17271%	2.65774%	1.18363%	-1.03512%	0.64151%	0.754044%
Дисперсія прибутку	0.655308%	0.195971%	0.384614%	0.59428%	0.485149%	0.122764%
Відхилення прибутку	8.09567%	4.42686%	6.20173%	7.70805%	6.96526%	3.50377%

Сформована модель Марковиця:

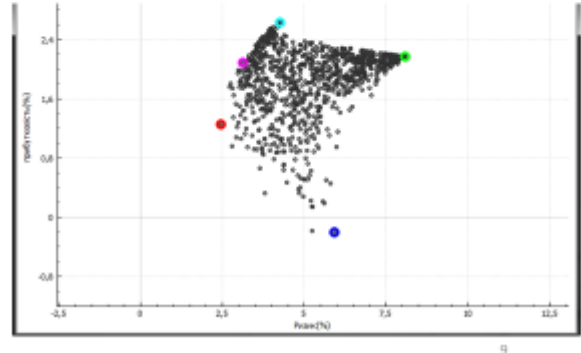


Рисунок Б.9 – Експериментальне дослідження

Висновки

У результаті виконання роботи було створено програмну систему для автоматизації процесів енергозбереження на металургійних підприємствах. Проаналізовані ризики при інвестуванні в об'єкт господарської власності, а також показана залежність між прибутком та ризиками. Виконано аналіз застосування сучасних методів портфельної оптимізації. Враховуючи основні принципи інвестиційної теорії, задачею портфельного інвестування є формування інвестиційного портфелю з відомими частками певних активів для максимізації віддачі та мінімізації ризику. Для вирішення даної проблеми було обрано методику Г. Марковиця.

В роботі виконано програмну реалізацію вирішення задачі аналізу процесів енергозбереження на металургійних підприємствах із використанням портфельної оптимізації та при застосуванні мови програмування C++ та фреймворку QT. Програма дозволяє в табличному та графічному виглядах представити результати портфельної оптимізації: відобразити динаміку вартості акцій певної компанії та динаміку зміни вартості, сформувати портфелі та виконати побудову моделі Марковиця і відображення її на графіку прибуток/ризик, переглянути матрицю коваріацій, отримати комплексну інформацію та відповідні рекомендації щодо розрахункових параметрів.

Виконано модульне тестування за допомогою Unit-tests та бібліотеки тестів Boots. Роботу програми також протестовано із використанням експериментальних тестових даних. Розроблений програмний продукт є повністю функціональним та готовим до експлуатації.

Рисунок Б.10 – Висновки