|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ «ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ  НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ  «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ» | | | | | | |
| Циклова комісія дисциплін програмної інженерії | | | | | | |
| **ЗВІТ** з навчальної практики | | | | | | |
| «Практика з об’єктно-орієнтованого програмування»  Розробка автоматизованої системи керування апаратами | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | Виконав: здобувач освіти 3 курсу, | | | | | |
|  | групи | 35 | |  | | |
|  | напрям підготовки: 121 «Інженерія | | | | | |
|  | програмного забезпечення» | | | | | |
|  | спеціалізації «Розробка програмного | | | | | |
|  | забезпечення» | | | | | |
|  | Грисенко А.О. | | | | | |
|  | (прізвище та ініціали) | | | | | |
|  | Керівник | |  | |  | Олійник В.В. |
|  |  | | (підпис) | |  | (прізвище та ініціали) |
|  | | | | | | |
| Полтава – 2024 | | | | | | |

**ЗМІСТ**

[ВСТУП 3](#_Toc161822144)

[1. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №1 4](#_Toc161822145)

[1.1. Постановка завдання 4](#_Toc161822146)

[1.3. Текст програми з коментарями 5](#_Toc161822147)

[1.4. Результати роботи програми 8](#_Toc161822148)

[2. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №2 9](#_Toc161822149)

[2.1. Постановка завдання 9](#_Toc161822150)

[2.2. Текст програми з коментарями. 9](#_Toc161822151)

[2.3. Результати роботи програми 10](#_Toc161822152)

[3. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №3 11](#_Toc161822153)

[3.1. Постановка завдання 11](#_Toc161822154)

[3.2. Текст програми з коментарями 11](#_Toc161822155)

[3.3. Результати роботи програми 12](#_Toc161822156)

[4. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №4 13](#_Toc161822157)

[4.1. Постановка завдання 13](#_Toc161822158)

[4.2. Текст програми з коментарями 13](#_Toc161822159)

[4.3. Результати роботи програми 15](#_Toc161822160)

[5. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №5 16](#_Toc161822161)

[5.1. Постановка завдання 16](#_Toc161822162)

[5.2. Текст програми з коментарями 16](#_Toc161822163)

[5.3. Результати роботи програми 17](#_Toc161822164)

[6. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №6 18](#_Toc161822165)

[6.1. Постановка завдання 18](#_Toc161822166)

[6.2. Текст програми з коментарями 18](#_Toc161822167)

[6.3. Результати роботи програми 19](#_Toc161822168)

[ВИСНОВКИ 20](#_Toc161822169)

[ДОДАТОК А. РЕПОЗИТОРІЙ ПРОГРАМИ 22](#_Toc161822170)

# ВСТУП

Практика з об'єктно-орієнтованого програмування є ключовим етапом у формуванні та розвитку професійних навичок студентів, які обрали спеціальність у сфері інформаційних технологій. Метою даної практики є не лише закріплення теоретичних знань, але й набуття практичних навичок та досвіду, що відповідають вимогам і особливостям сучасної організації професійної діяльності в галузі програмування.

Ця практика спрямована на вироблення у студентів ряду ключових компетенцій, які стануть основою для успішної реалізації їхньої майбутньої професійної діяльності. Серед цих компетенцій основними є:

* вивчення об'єктно-орієнтованого програмування, теорії алгоритмів і структур даних, технологій розробки програмного забезпечення та інших спеціальних дисциплін;
* отримання можливості відпрацьовувати свої навички під час розробки реального програмного продукту, використовуючи принципи об'єктно-орієнтованого програмування;
* засвоєння методів аналізу об'єктно-орієнтованого програмного коду, виявлення та вирішення можливих проблем та покращення ефективності;
* ознайомлення з усіма етапами життєвого циклу розробки програмного продукту, включаючи планування, реалізацію та тестування;
* розробка комплекту проектної документації, яка відображатиме основні результати практики та вміння студента;
* створення функціонального та ефективного об'єктно-орієнтованого програмного коду.

Ці завдання практики допоможуть студентам сформувати в собі високий рівень професійної підготовки, готовності до вирішення завдань у реальних умовах роботи та розвитку вмінь самостійно поновлювати свої знання та творчо застосовувати їх у практичній діяльності.

# 1. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №1

## 1.1. Постановка завдання

Створити програму на мові програмування Java в якій потрібно:

* розробити клас, що серіалізується, для зберігання параметрів і результатів обчислень. Використовуючи агрегування, розробити клас для знаходження рішення задачі;
* розробити клас для демонстрації в діалоговому режимі збереження та відновлення стану об'єкта, використовуючи серіалізацію. Показати особливості використання transient полів;
* розробити клас для тестування коректності результатів обчислень та серіалізації/десеріалізації. Використовувати докладні коментарі для автоматичної генерації документації засобами javadoc;
* індивідуальне завдання: визначити суму площ рівностороннього трикутника та рівностороннього прямокутника за заданою довжиною сторони у двійковій системі числення.

## 1.3. Текст програми з коментарями

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

Рисунок 1.1 – Реалізація серіалізації.

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

Рисунок 1.2 – Реалізація обчислення.

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 1.3 – Реалізація серіалізації та десеріалізації.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Рисунок 1.4 – Індивідуальне завдання.

## 1.4. Результати роботи програми

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Рисунок 1.5 – Відображення працездатності програми.

Зображення, що містить знімок екрана, текст

Автоматично згенерований опис

Рисунок 1.6 – Відображення працездатності програми.

# 2. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №2

## 2.1. Постановка завдання

* як основа використовувати вихідний текст проекту попередньої лабораторної роботи. Забезпечити розміщення результатів обчислень у колекції з можливістю збереження/відновлення;
* використовуючи шаблон проектування Factory Method (Virtual Constructor), розробити ієрархію, що передбачає розширення рахунок додавання нових відображуваних класів;
* розширити ієрархію інтерфейсом "фабрикованих" об'єктів, що представляє набір методів для відображення результатів обчислень;
* реалізувати ці методи виведення результатів у текстовому виді;
* розробити тареалізувати інтерфейс для "фабрикуючого" методу.

## 2.2. Текст програми з коментарями.

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 2.1 – Реалізація серіалізації та десеріалізації.

## 2.3. Результати роботи програми

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Рисунок 2.2 – Відображення працездатності програми.

# 3. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №3

## 3.1. Постановка завдання

* за основу використовувати вихідний текст проекту попередньої лабораторної роботи. Використовуючи шаблон проектування Factory Method (Virtual Constructor), розширити ієрархію похідними класами, реалізують методи для подання результатів у вигляді текстової таблиці. Параметри відображення таблиці мають визначатися користувачем;
* продемонструвати заміщення (перевизначення, overriding), поєднання (перевантаження, overloading), динамічне призначення методів (Пізнє зв'язування, поліморфізм, dynamic method dispatch);
* забезпечити діалоговий інтерфейс із користувачем;
* розробити клас для тестування основної функціональності;
* використати коментарі для автоматичної генерації документації засобами javadoc

## 3.2. Текст програми з коментарями

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 3.1 – Перевірка функціональності.

## 3.3. Результати роботи програми

**Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис**

Рисунок 3.2 – Відображення працездатності програми.

4. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №4

## 4.1. **Постановка завдання**

* реалізувати можливість скасування (undo) операцій (команд);
* продемонструвати поняття "макрокоманда";
* при розробці програми використовувати шаблон Singletone;
* забезпечити діалоговий інтерфейс із користувачем;
* розробити клас для тестування функціональності програми.

## 4.2. Текст програми з коментарями

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 4.1 – Реалізації команд для виконання та скасування операцій обчислення.

Зображення, що містить текст, знімок екрана, дизайн

Автоматично згенерований опис

Рисунок 4.2 – Управління виконанням та скасуванням команд.

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

Рисунок 4.3 – Реалізація макрокоманди.

## 4.3. Результати роботи програми

Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

Рисунок 4.4 – Відображення працездатності програми.

# 5. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №5

## 5.1. **Постановка завдання**

* продемонструвати можливість паралельної обробки елементів колекції (пошук мінімуму, максимуму, обчислення середнього значення, відбір за критерієм, статистична обробка тощо);
* управління чергою завдань (команд) реалізувати за допомогою шаблону Worker Thread.

## 5.2. Текст програми з коментарями

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 5.1 – Тестування розроблених класів.

## 5.3. Результати роботи програми

**Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис**

Рисунок 5.2 – Виконання роботи програми.

# 6. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ №6

## 6.1. **Постановка завдання**

* розробити ієрархію класів відповідно до шаблону Observer (java) та продемонструвати можливість обслуговування розробленої раніше колекції (об'єкт, що спостерігається, Observable) різними (не менше двох) спостерігачами (Observers) – відстеження змін, упорядкування, висновок, відображення і т.д.;
* при реалізації ієрархії класів використати інструкції (Annotation). Відзначити особливості різних політик утримання анотацій (annotation retention policies). Продемонструвати підтримку класів концепції рефлексії (Reflection);
* використовуючи раніше створені класи, розробити додаток, що відображає результати обробки колекції об'єктів у графічному вигляді;
* забезпечити діалоговий інтерфейс з користувачем та перемальовування графіка під час зміни значень елементів колекції.

## 6.2. Текст програми з коментарями

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 6.1 – Реалізація графічного інтерфейсу.

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рисунок 6.2 – Реалізація Observer.

## 6.3. Результати роботи програми

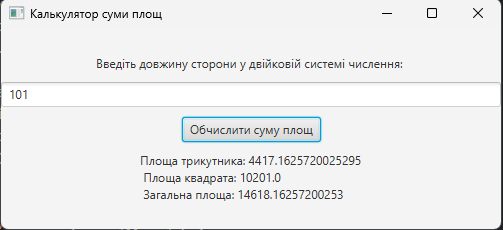


Рисунок 6.3 – Виконання роботи програми.

# ВИСНОВКИ

Протягом практики було здійснено глибоке вивчення об’єктно-орієнтованого програмування, теорії алгоритмів, структур даних та сучасних технологій розробки програмного забезпечення. Я отримав цінний досвід, відпрацювавши навички на реальному програмному продукті, що дозволило застосувати теоретичні знання на практиці. Було освоєно методи аналізу програмного коду, ідентифікації та вирішення проблем, що сприяло підвищенню ефективності розробки. Знання про життєвий цикл програмного продукту було поглиблено, включаючи всі етапи від планування до тестування. Розроблено комплект проектної документації, який відображає ключові досягнення практики та демонструє професійні вміння. В результаті, я набув здатності створювати функціональний та ефективний об’єктно-орієнтований програмний код.СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. МЕТОДИЧНІ ВКАЗІВКИ ДО ПРОХОДЖЕННЯ ПРАКТИКИ З «ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ» ДЛЯ ЗДОБУВАЧІВ ОСВІТИ НАПРЯМУ ПІДГОТОВКИ 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ» СПЕЦІАЛІЗАЦІЇ «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ» [Текст]: МЕТОДИЧНІ ВКАЗІВКИ / Відокремлений структурний підрозділ «Полтавський політехнічний фаховий коледж Національного технічного університету «Харківський політехнічний інститут»; [уклад.: В.В.Олійник, А.О.Зелінська]. – Полтава: ВСП «ППФК НТУ «ХПІ», 2022. – 27 с.
2. Effective Java / за ред. Joshua Bloch. United States, 2017. 416 с.
3. Java: A Beginner's Guide / за ред. Herbert Schildt. United States, 2018. 648 с.
4. Java Concurrency in Practice / за ред. Brian Goetz . United States, 2006. 432 с.
5. Clean Code: A Handbook of Agile Software Craftsmanship / за ред. Robert C. Martin . United States, 2008. 464 с.
6. Oracle Java Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.oracle.com/en/java/> – Назва з екрану.
7. Java Code Geeks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javacodegeeks.com/> – Назва з екрану.
8. Stack Overflow [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/> – Назва з екрану.

# ДОДАТОК А. **РЕПОЗИТОРІЙ ПРОГРАМИ**



Рисунок А.1 – QR код на повний лістинг програми