­­­­­­­­Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до лабораторної роботи № 4-5

з дисципліни

*Прикладне програмування*

на тему:

**“ Комплексна лабораторна робота ”**

Виконав: студент ОІ-24

**Зубач Руслан**

Прийняв: асистент каф. АСУ

Чорненький В.Я.

**Львів – 2024**

*Лабораторна робота №* *4-5*

**Завдання лабораторної роботи**

Створити консольну програму на мові програмування Java згідно до індивідуального завдання (ваш номер варіанту, згенерований випадковим чином, можна знайти у файлі «Варіанти завдань.pdf»). Саме завдання описане досить поверхово, якщо ви претендуєте на високу оцінку – його потрібно розширити. Наприклад, у завданні може бути сказано про «створити кілька квіток, з наявних квіток сформувати букет». Цілком логічно буде додати можливість видалити квітку, поміняти параметри існуючої квітки тощо. Оцінка за лабораторну роботу залежатиме і від глибини пропрацювання предметної області.

Окрім цього, програма повинна відповідати таким додатковим вимогам:

• Використовувати можливості ООП: класи, успадкування, поліморфізм, інкапсуляцію.

• Кожний клас повинен мати назву, яка повністю описує його суть, і інформативний склад.

• Успадкування потрібно використовувати тільки тоді, коли воно має сенс.

• При записі програми потрібно використовувати домовленості щодо оформлення коду java code convention.

• Класи потрібно грамотно розкласти по пакетах.

• Для збереження параметрів ініціалізації потрібно використовувати файли.

• Повинне бути консольне меню для вибору різних дій, наприклад: «завантажити з файлу», «довідка», «вихід» тощо.

Етапи виконання завдання (кожен етап вважається окремою лабораторною роботою):

1. Уважно пропрацюйте предметну область згідно вашого варіанту завдання. Опишіть технологічне завдання (який функціонал матиме ваша програма). Створіть Use Case та Class діаграми для вашого завдання.

2. Реалізуйте консольне меню до вашого завдання з використанням паттерну «Команда». На цьому етапі не потрібно реалізовувати функціонал конкретних пунктів меню – тільки створити «кістяк» вашої програми.

3. Реалізуйте функціонал програми. Обов’язково зверніть увагу на додаткові вимоги до програми (описані вище).

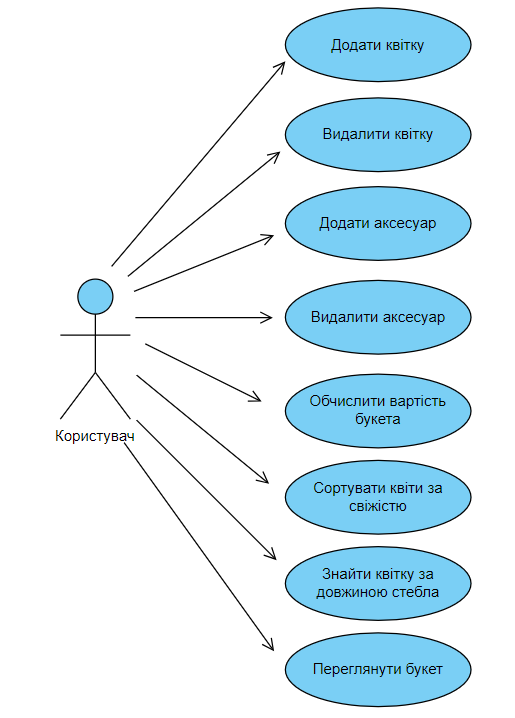
4. Покрийте вашу програму юніт тестами (coverage >=90%).

5. Додайте логгер з логуванням основних дій та виняткових ситуацій. При цьому далі логу мають зберігатися у файлі, а критичні помилки – додатково надсилатися на e-mail

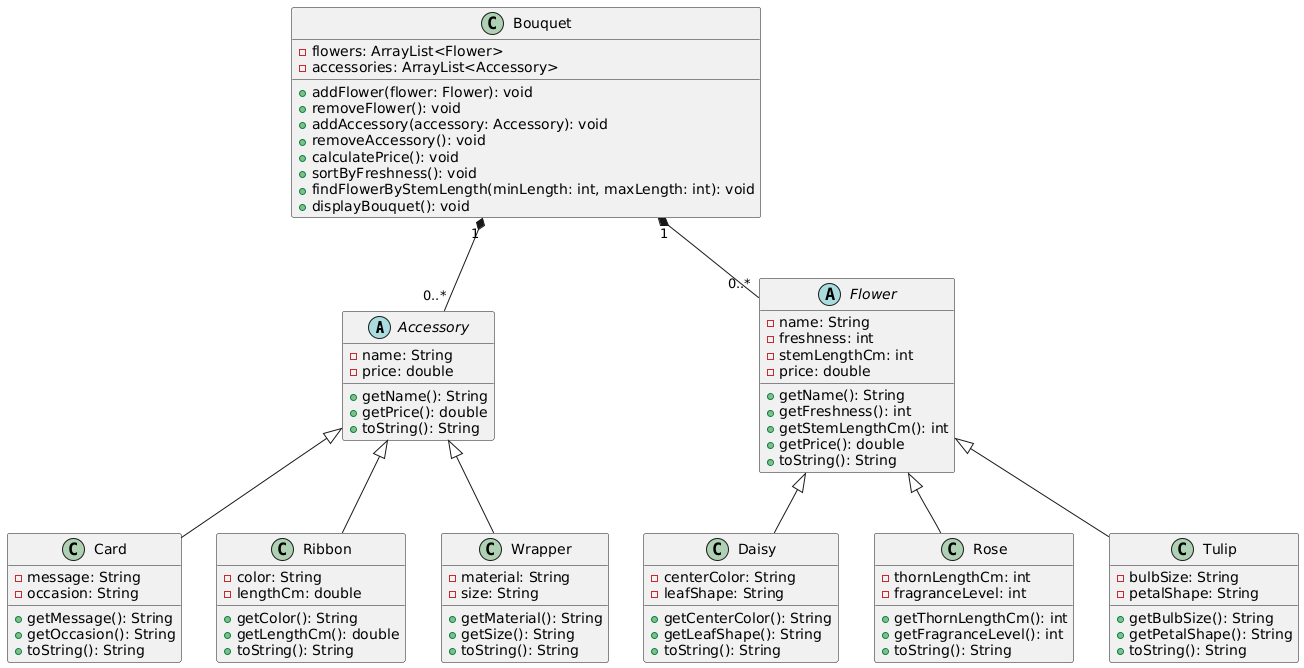
**Варіант 1**

Квіти. Визначити ієрархію квітів. Створити кілька об’єктів-квітів. Зібрати букет (використовуючи аксесуари) з визначенням його вартості. Здійснити сортування квітів у букеті на основі рівня свіжості. Знайти квітку в букеті, що відповідає заданому діапазону довжин стебел.

**Use Case діаграма**



**Class діаграма**

****

**Код програми**

**Command.java**

package command;  
  
public interface Command {  
 void execute();  
}

**Menu.java**

package menu;  
  
import command.Command;  
  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
public class Menu {  
 private Map<Integer, Command> commands = new HashMap<>();  
  
 public void addCommand(int option, Command command) {  
 commands.put(option, command);  
 }  
  
 public void showMenu() {  
 Scanner sc = new Scanner(System.*in*);  
 int option;  
  
 do{  
 System.*out*.println("==== Меню ====");  
 System.*out*.println("1. Додати/Видалити квітку до букета");  
 System.*out*.println("2. Додати/Видалити аксесуари");  
 System.*out*.println("3. Обчислити вартість букета");  
 System.*out*.println("4. Сортувати квіти за свіжістю");  
 System.*out*.println("5. Знайти квітку за довжиною стебла");  
 System.*out*.println("6. Переглянути букет");  
 System.*out*.println("7. Вийти");  
 System.*out*.print("Оберіть опцію: ");  
 option = sc.nextInt();  
 sc.nextLine();  
  
 Command cmd = commands.get(option);  
 if (cmd != null) {  
 cmd.execute();  
 } else if (option != 7) {  
 System.*out*.println("Невірний вибір. Спробуйте ще раз.");  
 }  
 } while (option != 7);  
  
 System.*out*.println("Програма завершена.");  
 sc.close();  
 }  
}

**Main.java**

package main;  
import bouquet.Bouquet;  
import command.\*;  
import menu.Menu;  
  
public class Main {  
 public static void main(String[] args) {  
 Bouquet bouquet = new Bouquet();  
  
 Menu menu = new Menu();  
  
 menu.addCommand(1, new AddRemoveFlowerCommand(bouquet));  
 menu.addCommand(2, new AddRemoveAccessoryCommand(bouquet));  
 menu.addCommand(3, new CalculateBouquetPriceCommand(bouquet));  
 menu.addCommand(4, new SortByFreshnessCommand(bouquet));  
 menu.addCommand(5, new FindFLowerByStemLengthCommand(bouquet));  
 menu.addCommand(6, new DisplayBouquetPriceCommand(bouquet));;  
  
 menu.showMenu();  
 }  
}

**Git репозиторій:** <https://github.com/wixieee/AP_LPNU/tree/master/LR4-8PP>

**Висновок**

На цій лабораторній роботі я ознайомився з Use Case діаграмами та Class діаграмами. Опанував паттерн команда