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

**Тріщук Максим**

Прийняв: Пелех Ю.М.

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.

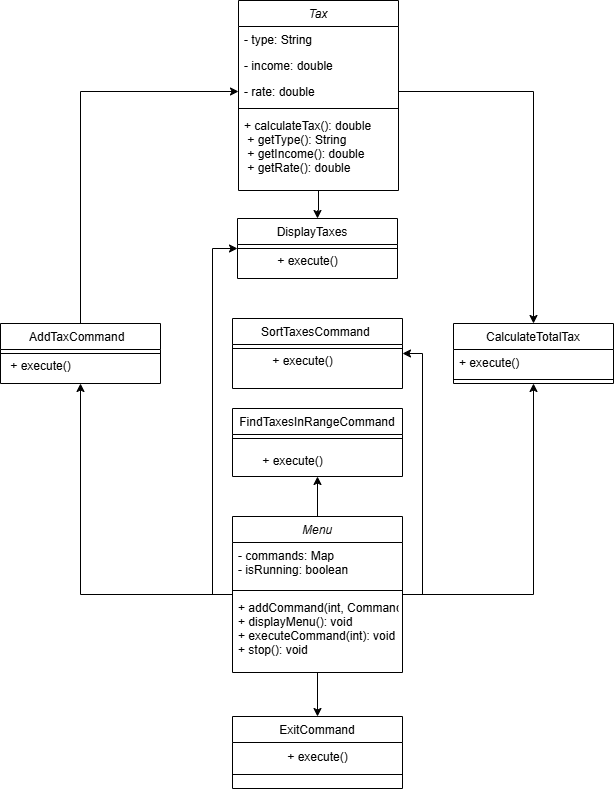
• Для збереження параметрів ініціалізації потрібно використовувати файли.

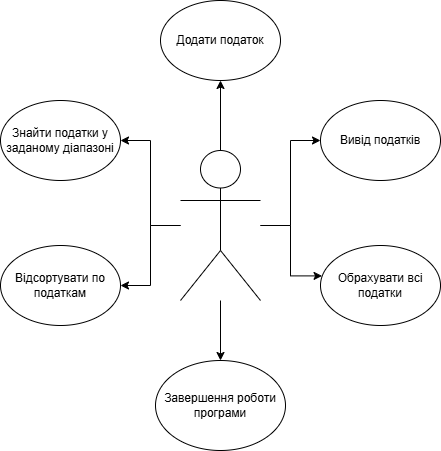
• Повинне бути консольне меню для вибору різних дій, наприклад: «завантажити з файлу», «довідка», «вихід» тощо.

**Варіант 15**

**Податки.** Визначити множину і суму податкових виплат фізичної особи за рік із врахуванням доходів з основного і додаткового місць роботи, авторських винагород, продажу майна, отримання в подарунок грошових сум і майна, переказів з-за кордону, пільг на дітей і матеріальну допомогу. Здійснити сортування податків по сумі. Знайти для особи податки, що відповідають заданому діапазону параметрів.

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.

class Menu {

private Map<Integer, Command> commands = new HashMap<>();

private boolean isRunning = true;

public void addCommand(int option, Command command) {

commands.put(option, command);

}

public void displayMenu() {

System.out.println("\n=== Tax Management Menu ===");

System.out.println("1. Add a new tax");

System.out.println("2. Display all taxes");

System.out.println("3. Calculate total tax");

System.out.println("4. Sort taxes by amount");

System.out.println("5. Find taxes in a range");

System.out.println("6. Exit");

System.out.print("Choose an option: ");

}

public void executeCommand(int option) {

Command command = commands.get(option);

if (command != null) {

command.execute();

} else {

System.out.println("Invalid option. Please try again.");

}

}

public void stop() {

isRunning = false;

}

public boolean isRunning() {

return isRunning;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Tax> taxes = new ArrayList<>();

Menu menu = new Menu();

// Додаємо команди до меню

menu.addCommand(1, new AddTaxCommand(taxes, scanner));

menu.addCommand(2, new DisplayTaxesCommand(taxes));

menu.addCommand(3, new CalculateTotalTaxCommand(taxes));

menu.addCommand(4, new SortTaxesCommand(taxes));

menu.addCommand(5, new FindTaxesInRangeCommand(taxes, scanner));

menu.addCommand(6, new ExitCommand(menu));

while (menu.isRunning()) {

menu.displayMenu();

int option;

try {

option = Integer.parseInt(scanner.nextLine());

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please enter a number.");

continue;

}

menu.executeCommand(option);

}

scanner.close();

}

}

**Висновок**

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

https://github.com/itimodzi/Trishchuk\_oi-24\_2024.git