

Міністерство освіти і науки України
Національний університет „Львівська політехніка”



Звіт з лабораторної роботи №3
з дисципліни «Кросплатформні засоби програмування»
на тему: «КЛАСИ ТА ПАКЕТИ»

Виконав: ст.групи КІ-36

Борисов І. С.

Прийняв, та перевірів :

Іванов Ю. С.

Львів 2022

Мета: ознайомитися з процесом розробки класів та пакетів мовою Java.

Завдання

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту. Програма має задовольняти наступним вимогам:

- програма має розміщуватися в пакеті Група.Прізвище.Lab3;
- клас має містити мінімум 3 поля, що є об'єктами класів, які описують складові частини предметної області;
- клас має містити кілька конструкторів та мінімум 10 методів;
- для тестування і демонстрації роботи розробленого класу розробити клас драйвер;
- методи класу мають вести протокол своєї діяльності, що записується у файл;
- розробити механізм коректного завершення роботи з файлом (не надіятися на метод `finalize()`);
 - програма має володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

2. Автоматично згенерувати документацію до розробленого пакету.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

Варіант №13 Телефон.

Хід роботи:

Код програми:

Main.java

```
public class Main {
    public static void main(String[] args) {
        Phone apps = new Phone();

        apps.getCurrentApp();
        apps.turnOn();
        apps.setScreenResolution(1200, 800);
        apps.addApp("Viber", "Telegram", "Safari", "Google");
        apps.removeApp("Viber");
        apps.setCurrentApp("Telegram");
        apps.getApp();
        apps.removeApp("Google");
        apps.setCurrentApp("Safari");
        apps.getCurrentApp();
        apps.getApp();
        apps.settings();

        Phone settin = new Phone(apps);

        settin.setCurrentApp("Telegram");
        settin.clearAllApps();
        settin.getCurrentApp();
        settin.settings();

        Phone setting = new Phone(1080, 1920, 144);
        setting.turnOn();
        setting.setScreenResolution(1080, 1920);
        setting.getScreenResolution();
        setting.setPhoneFrameRate(144);
        setting.settings();
        setting.turnOff();
        setting.isWorking();

    }
}
```

Apps.java

```
import lombok.NoArgsConstructor;

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;

/** зберігання додатків, які відкриті або існують */
@NoArgsConstructor
public class Apps {
    private final List<String> apps = new ArrayList<>();

    public Apps(Collection<String> apps) {
        this.apps.addAll(apps);
    }

    public Apps(Apps apps) {
        this.apps.addAll(apps.apps);
    }

    public List<String> getApps() {
```

```

        return apps;
    }

    public void addApp(String app) {
        if (apps.contains(app)) {
            throw new IllegalArgumentException("This app already at list");
        }
        apps.add(app);
    }

    public void removeApp(String app) {
        if (apps.contains(app)) {
            apps.remove(app);
        } else {
            throw new IllegalArgumentException("No such app at list");
        }
    }

    public boolean hasApp(String app) {
        return apps.contains(app);
    }

    public String clearAll() {
        apps.clear();
        return "No apps";
    }
}

```

Desktop.java

```

public class Desktop {
    /** оголошення полів */
    private int width;
    private int height;
    private int FPS;

    /** конструктор без параметрів */
    public Desktop() {
        width = 100;
        height = 100;
        FPS = 60;
    }

    /** конструктор з параметрами */
    public Desktop(int width, int height, int frameRate) {
        if (width < 0 || height < 0 || frameRate < 0) {
            throw new IllegalArgumentException("Desktop width and height must be positive number");
        }
        /**звертаємось до змінних класів */
        this.width = width;
        this.height = height;
        this.FPS = frameRate;
    }

    /** конструктор для копіювання об'єкту */
    public Desktop(Desktop other) {
        this.width = other.width;
        this.height = other.height;
        this.FPS = other.FPS;
    }

    /** методи get i set для задання розміру екрану */
    public int getWidth() {

```

```

        return width;
    }

    public int getHeight() {
        return height;
    }

    public int getFPS() {
        return FPS;
    }

    /** умови задання розмірів екрану з перевіркою */

    public void setWidth(int width) {
        if (width < 0) {
            throw new IllegalArgumentException("Turn off the iPhone");
        }
        this.width = width;
    }

    public void setHeight(int height) {
        if (height < 0) {
            throw new IllegalArgumentException("Desktop width and height must be positive number");
        }
        this.height = height;
    }

    public void setFPS(int FPS) {
        if (FPS < 0) {
            throw new IllegalArgumentException("Frame rate must be bigger than 0");
        }
        this.FPS = FPS;
    }

    /** об'єднання даних і методів роботи з цими даними */
    public void setResolution(int width, int height) {
        setHeight(height);
        setWidth(width);
    }
}

```

Phone.java

```

import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;

/** клас самого об'єкту, який буде використовувати інші класи */
public class Phone {
    private final Desktop desktop;
    private final Apps apps;
    private String currentApp;
    private boolean working;

    /** зміни які додають кольори в консоль */
    public static final String ANSI_RESET = "\u001B[0m";
    public static final String ANSI_GREEN = "\u001B[32m";
    public static final String ANSI_YELLOW = "\u001B[43m";
    public static final String ANSI_BLUE = "\u001B[44m";
    public static final String ANSI_PURPLE = "\u001B[35m";
}

```

```

public static final String ANSI_CYAN = "\u001B[36m";

private static final String file = "Lab_03.txt";

/** конструктор по замовчуванню */
public Phone() {
    this.desktop = new Desktop();
    this.working = true;
    this.apps = new Apps();
}

/** конструктор із заданням параметрів */
public Phone(int width, int height, int frameRate, Collection<String> apps)
{
    this.desktop = new Desktop(width, height, frameRate);
    this.apps = new Apps(apps);
    this.working = false;
}

/** конструктор із заданням параметрів */
public Phone(int width, int height, int frameRate) {
    this(width, height, frameRate, Collections.emptyList());
}

/** конструктор для копіювання об'єкта */
public Phone(Phone Phone) {
    this.apps = new Apps(Phone.apps);
    this.desktop = new Desktop(Phone.desktop);
    this.currentApp = Phone.currentApp;
    this.working = Phone.working;
}

/** функція яка нічого не повертає */
public void turnOn() {
    filler("Turn on the Phone");
    System.out.println(ANSI_CYAN + "Turn on the Phone" + ANSI_RESET);
    working = true;
}

public void turnOff() {
    filler("Turn off the iPhone");
    System.out.println(ANSI_CYAN + "Turn off the iPhone" + ANSI_RESET);
    working = false;
}

public void isWorking() {
    filler("Is Phone working: " + working);
}

public void settings() {
    filler("
                                Current settings
                                ");
    System.out.println(ANSI_YELLOW + "
                                Current settings
" + ANSI_RESET);
    getScreenResolution();
    getScreenFrameRate();
    getCurrentApp();
    getApps();
    filler("
                                ");
    System.out.println(ANSI_BLUE + "
" + ANSI_RESET);
}

/** запис тексту в файл */
private static void filler(String text) {
    try {
        Files.write(Paths.get(file), (text + "\n").getBytes(),

```

```

StandardOpenOption.APPEND);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void addApp(String... apps) {
    filler("Add programs: " + String.join(", ", apps));
    System.out.println("Add programs: " + String.join(", ", apps));
    Arrays.stream(apps).forEach(this.apps::addApp);
}

public void removeApp(String... apps) {
    filler("Remove programs: " + String.join(", ", apps));
    System.out.println("Remove programs: " + String.join(", ", apps));
    Arrays.stream(apps).forEach(this.apps::removeApp);
}

public void getApps() {
    filler("Current list of programs: " + String.join(", ",
apps.getApps()));
    System.out.println("Current list of programs: " + String.join(", ",
apps.getApps()));
}

public void setCurrentApp(String currentApp) {
    if (!working) {
        System.err.println("You must turn on Phone first");
    } else if (!apps.hasApp(currentApp)) {
        System.err.println("No such program");
    } else {
        filler("Set program to " + currentApp);
        System.out.println("Set program to " + currentApp);
        this.currentApp = currentApp;
    }
}

public void getCurrentApp() {
    filler("Get the current program: ");
    System.out.println(ANSI_GREEN + "Get the current program: " +
ANSI_RESET);
    if (!working) {
        System.err.println("You must turn on Phone first");
    } else {
        filler("Current program is " + currentApp);
        System.out.println("Current program is " + currentApp);
    }
}

public void clearAllApps() {
    if (!working) {
        System.out.println("You must turn on Phone first");
    } else {
        filler("Go to next app" + apps.clearAll());
        System.out.println("Go to next app" + apps.clearAll());
    }
}

public void getScreenResolution() {
    filler("Get desctop resolution: " + "Height: " + desctop.getHeight() + "
Width: " + desctop.getWidth());
    System.out.println(ANSI_PURPLE + "Get desctop resolution: " + "Height: "
+ desctop.getHeight() + " Width: " + desctop.getWidth() + ANSI_RESET);
}

```

```

        public void getScreenFrameRate() {
            filler("Get desktop frame rate: " + desctop.getFPS());
            System.out.println(ANSI_PURPLE + "Get desktop frame rate: " +
desctop.getFPS() + ANSI_RESET);
        }

        public void setScreenResolution(int width, int height) {
            filler("Set desktop resolution to: " + height + "x" + width);
            System.out.println(ANSI_PURPLE + "Set desktop resolution to: " + height
+ "x" + width + ANSI_RESET);
            if (width < 0 || height < 0) {
                System.out.println("Width and height must be bigger than 0");
            } else {
                desctop.setResolution(width, height);
            }
        }

        public void setPhoneFrameRate(int frameRate) {
            desctop.setFPS(frameRate);
        }
    }
}

```

Результат виконання:

```

Get the current program:
Current program is null
Turn on the Phone
Set desktop resolution to: 800x1200
Add programs: Viber, Telegram, Safari, Google
Remove programs: Viber
Set program to Telegram
Current list of programs: Telegram, Safari, Google
Remove programs: Google
Set program to Safari
Get the current program:
Current program is Safari
Current list of programs: Telegram, Safari
Current settings
Get desktop resolution: Height: 800 Width: 1200
Get desktop frame rate: 60
Get the current program:
Current program is Safari
Current list of programs: Telegram, Safari
Set program to Telegram
Go to next appNo apps
Get the current program:
Current program is Telegram
Current settings
Get desktop resolution: Height: 800 Width: 1200
Get desktop frame rate: 60

```



```
Current program is Telegram
Current list of programs:
[redacted]
Turn on the Phone
Set desktop resolution to: 1920x1080
Get desktop resolution: Height: 1920 Width: 1080
[redacted]
Current settings
Get desktop resolution: Height: 1920 Width: 1080
Get desktop frame rate: 144
Get the current program:
Current program is null
Current list of programs:
[redacted]
Turn off the iPhone

Process finished with exit code 0
```

Контрольні питання

1. Синтаксис визначення класу.

Синтаксис оголошення простого класу в мові Java має наступний вигляд:

```
[public] class НазваКласу
{
    [конструктори]
    [методи]
    [поля]
}
```

2. Синтаксис визначення методу.

Синтаксис оголошення методу наступний:

```
[СпецифікаторДоступу] [static] [final] Тип назваМетоду([параметри]) [throws класи]
{
    [Тіло методу]
    [return [значення]];
}
```

3. Синтаксис оголошення поля.

Синтаксис оголошення поля наступний:

[СпецифікаторДоступу] [static] [final] Тип НазваПоля [= ПочатковеЗначення];

4. **Як оголосити та ініціалізувати константне поле?**

Оголосити константне поле можна за допомогою ключового слова final:

[СпецифікаторДоступу] [static] final Тип НазваПоля [= ПочатковеЗначення];

```
private final int i = 5;
```

5. **Які є способи ініціалізації полів?**

Ініціалізацію полів при створенні об'єкту можна здійснювати трьома способами:

- у конструкторі;
- явно при оголошенні поля;
- у блоці ініціалізації (виконується перед виконанням конструктора).

6. **Синтаксис визначення конструктора.**

Синтаксис оголошення конструктора:

[СпецифікаторДоступу] НазваКласу([параметри])

{

Тіло конструктора

}

7. **Синтаксис оголошення пакету.**

Синтаксис оператора package:

```
package НазваПакету{.НазваПідпакету};
```

8. **Як підключити до програми класи, що визначені в зовнішніх пакетах?**

Для підключення одного загальнодоступного класу пакету необхідно за допомогою оператора import через крапку вказати повну ієрархію пакету та назву класу, який має бути імпортовано, наприклад,

```
import java.util.Date
```

```
Date today = new Date();
```

Для підключення всіх загальнодоступних класів пакету необхідно за допомогою оператора `import` через крапку вказати повну ієрархію пакету та символ зірочка (*), наприклад,

```
import java.util.*
```

```
Date today = new Date();
```

9. В чому суть статичного імпорту пакетів?

Статичний імпорт дозволяє не вживати явно назву класу при звертанні до статичного поля або методу класу.

10. Які вимоги ставляться до файлів і каталогів при використанні пакетів?

Використання пакетів вимагає, щоб файли і каталоги проекту та їх ієрархія були строго структурованими. Так назви пакету і його підпакетів мають співпадати з назвами каталогів, де вони розміщуються. Назви загальнодоступних класів мають співпадати з назвами файлів, де вони розміщуються. Ієрархія каталогів і файлів проекту має співпадати з ієрархією пакетів. Після компіляції ієрархія каталогів, де містяться файли класів, співпадає з ієрархією каталогів проекту.

Висновок: на даній лабораторній роботі я ознайомився з процесом розробки класів та пакетів мовою Java.