Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Звіт з лабораторної роботи №4

з дисципліни «Кросплатформні засоби програмування»

на тему: «СПАДКУВАННЯ ТА ІНТЕРФЕЙСИ»

Виконав: ст.групи КІ-36

Борисов І. С.

Прийняв, та перевірив :

Іванов Ю. С.

Львів 2022

Мета: ознайомитися з спадкуванням та інтерфейсами у мові Java.

Завдання

1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у лабораторній роботі №3, для реалізації предметної області заданої варіантом. Суперклас, що реалізований у лабораторній роботі №3, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті Група.Прізвище.Lab4 та 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.getApps();
        apps.removeApp("Google");
        apps.setCurrentApp("Safari");
        apps.getCurrentApp();
        apps.getApps();
        apps.settings();
```

```
        Phone settin = new Phone();

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

        Phone setting = new Phone(1080, 1920, 140);
        setting.turnOn();
        setting.setScreenResolution(1080, 1920);
        setting.getScreenResolution();
        setting.setPhoneFrameRate(144);
        setting.settings();
        setting.turnOff();
        setting.isWorking();

        IMag iMag = new IMag();
        iMag.turnOn();
        iMag.addApp("Viber", "Telegram", "Safari", "Google");
        iMag.addAppsFromAppStore();
        iMag.getApps();
        iMag.turnOff();

    }
}
```

Apps.java

```
public class Apps{
    private final List<String> apps = new ArrayList<>();

    public Apps() {
    }

    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";
    }
}
```

Desctop.java

```
public class Desctop {

    private int width;
    private int height;
    private int frameRate;

    public Desctop() {
        width = 100;
        height = 100;
        frameRate = 60;
    }
    /**конструктор з параметрами */
    public Desctop(int width, int height, int frameRate) {
        if (width < 0 || height < 0 || frameRate < 0) {
            throw new IllegalArgumentException("Desctop width and height must be
positive number");
        }
    /**звертаємось до змінних класів */
        this.width = width;
        this.height = height;
        this.frameRate = frameRate;
    }
    /**конструктор для копіювання об'єкту */
    public Desctop(Desctop other) {
        this.width = other.width;
        this.height = other.height;
        this.frameRate = other.frameRate;
    }
    /**методи get і set для задання розміру екрану */
    public int getWidth() { return width; }

    public int getHeight() {
        return height;
    }

    public int getFrameRate() { return frameRate; }
    /**умови задання розмірів екрану з перевіркою */

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

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

    public void setFrameRate(int frameRate) {
        if (frameRate < 0) {
```

```
            throw new IllegalArgumentException("Frame rate must be bigger than
0");
        }
        this.frameRate = frameRate;
    }
    /**об'єднання даних і методів роботи з цими даними*/
    public void setResolution(int width, int height) {
        setHeight(height);
        setWidth(width);
    }
}
```

Phone.java

```
public class Phone extends Telefon{
    private final Desctop desctop;
    private final Apps apps;
    private String currentApp;
    private boolean isWorking;

    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_04.txt";
    /**конструктор по замовчуванню */
    public Phone() {
        this.desctop = new Desctop();
        this.isWorking = true;
        this.apps = new Apps();
    }

    public Phone(int width, int height, int frameRate, Collection<String> apps)
{
        this.desctop = new Desctop(width, height, frameRate);
        this.apps = new Apps(apps);
        this.isWorking = 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.desctop = new Desctop(phone.desctop);
        this.currentApp = phone.currentApp;
        this.isWorking = phone.isWorking;
    }
    /**функція яка нічого не повертає */
    public void turnOn() {
        filler("Turn on Telefon");
        System.out.println(ANSI_CYAN + "Turn on the Telefon" + ANSI_RESET);
        isWorking = true;
    }

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

```
    public void isWorking() {
        filler("Is Telefon working: " + isWorking);
        System.out.println("Is Telefon working: " + isWorking);
    }

    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 */
    protected 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 apps: " + 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 apps: " + 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 apps: " + String.join(", ", apps.getApps()));
        System.out.println("Current list of programs: " + String.join(", ",
apps.getApps()));
    }

    public void setCurrentApp(String currentApp) {
        if (!isWorking) {
            System.err.println("You must turn on Telefon first");
        } else if (!apps.hasApp(currentApp)) {
            System.err.println("");
        } 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 (!isWorking) {
            System.err.println("You must turn on Telefon first");
        } else {
            filler("Current program is " + currentApp);
            System.out.println("Current program is " + currentApp);
        }
```

```
    }

    public void clearAllApps() {
        if (!isWorking) {
            System.out.println("You must turn on Telefon 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 desctop frame rate: " + desctop.getFrameRate());
        System.out.println(ANSI_PURPLE + "Get desctop frame rate: " +
desctop.getFrameRate() + ANSI_RESET);
    }

    public void setScreenResolution(int width, int height) {
        filler("Set desctop resolution to: " + height + "x" + width);
        System.out.println(ANSI_PURPLE + "Set desctop 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.setFrameRate(frameRate);
    }
}
```

Telefon.java

```
public abstract class Telefon {

    public abstract void turnOn();
    public abstract void turnOff();
    public abstract void isWorking();
    public abstract void settings();
    public abstract void addApp(String... apps);
    public abstract void removeApp(String... apps);
    public abstract void getApps();
    public abstract void setCurrentApp(String currentApp) ;
    public abstract void getCurrentApp();
    public abstract void clearAllApps();
    public abstract void getScreenResolution();
    public abstract void getScreenFrameRate();
    public abstract void setScreenResolution(int width, int height);
    public abstract void setPhoneFrameRate(int frameRate);
}
```

IMag.java

```java
public class IMag extends Phone implements iMagInterface {
    public IMag() {
        new Desctop(4608, 1152, 1440);
    }

    public IMag(int width, int height, int frameRate) {
        new Desctop(width, height, frameRate);
    }

    public void addAppsFromAppStore() {
        filler("Add apps from AppStore");
        System.out.println("Add apps from AppStore");
        addApp("Xcode", "Photoshop", "Teams");
    }

}
```

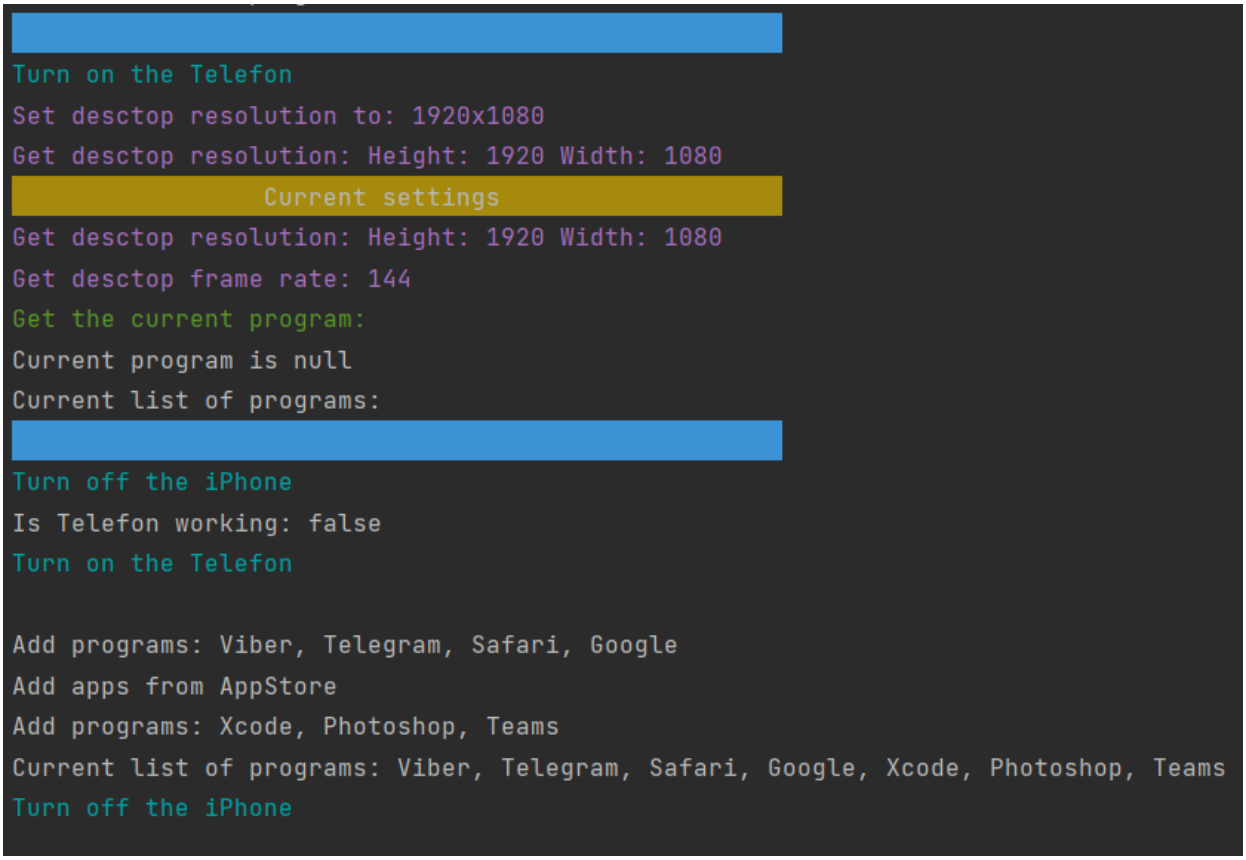
iMagInterface.java

```java
public interface iMagInterface {
    void addAppsFromAppStore();
}
```

Результат виконання:

```
Get the current program:
Current program is null
Turn on the Telefon
Set desctop 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 desctop resolution: Height: 800 Width: 1200
Get desctop frame rate: 60
Get the current program:
Current program is Safari
Current list of programs: Telegram, Safari

Go to next appNo apps
Get the current program:
Current program is null
                Current settings
Get desctop resolution: Height: 100 Width: 100
Get desctop frame rate: 60
Get the current program:
Current program is null
Current list of programs:
```



Контрольні питання

1. Синтаксис реалізації спадкування.

class Підклас extends Суперклас {

Додаткові поля і методи }

2. Що таке суперклас та підклас?

Суперклас – це базовий клас, а підклас – це похідний клас від суперкласу.

3. Як звернутися до членів суперкласу з підкласу?

Для звернення до методів чи полів суперкласу з підкласу потрібно використати ключове слово super.

super.назваМетоду([параметри]); // виклик методу суперкласу

super.назваПоля // звертання до поля суперкласу

4. Коли використовується статичне зв’язування при виклику методу?

Механізм статичного зв’язування передбачає визначення методу, який необхідно викликати, на етапі компіляції.

5. Як відбувається динамічне зв’язування при виклику методу?

Поліморфізм реалізується за допомогою механізму динамічного зв’язування, який полягає у тому, що вибір методу, який необхідно викликати, відбувається не на етапі компіляції, а під час виконання програми.

6. Що таке абстрактний клас та як його реалізувати?

Абстрактні класи призначені бути основою для розробки ієрархій класів та не дозволяють створювати об’єкти свого класу. Вони реалізуються за допомогою ключового слова abstract. На відміну від звичайних класів абстрактні класи можуть містити абстрактні методи (а можуть і не містити).

7. Для чого використовується ключове слово instanceof?

Оператор instanceof дозволяє перевірити приналежність об'єкта до зазначеного класу з урахуванням успадкування.

8. Як перевірити чи клас є підкласом іншого класу?

1) Можна перевірити за допомогою ключового слова instanceof.

2) Перевірити за допомогою метода isAssignableFrom(), наприклад:

```
public class Test {
public class A {}
public class B extends A {}
public class C {}
public static void main(String[] args) {
System.out.println("B extends A : " + A.class.isAssignableFrom(B.class));
System.out.println("C extends A : " + A.class.isAssignableFrom(C.class));
}
}
```

Результат:

B extends A : true

C extends A : false

9. Що таке інтерфейс?

Інтерфейси вказують що повинен робити клас не вказуючи як саме він це повинен робити. Інтерфейси покликані компенсувати відсутність множинного спадкування у мові Java та гарантують визначення у класах оголошених у собі прототипів методів.

10. Як оголосити та застосувати інтерфейс?

Синтаксис оголошення інтерфейсів:

[public] interface НазваІнтерфейсу {

Прототипи методів та оголошення констант інтерфейсу

}

Для застосування інтерфейсу потрібно оголосити за допомогою ключового слова implements, що клас реалізує інтерфейс. Якщо клас реалізує кілька інтерфейсів, то вони перелічуються через кому після ключового слова implements.

Висновок: на даній лабораторній роботі я ознайомився з спадкуванням та інтерфейсами у мові Java.