

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Звіт

до лабораторної роботи №3

З дисципліни: «Кросплатформенні засоби програмування»

На тему: « СПАДКУВАННЯ ТА ІНТЕРФЕЙСИ »

Виконав:

Студент групи КІ-301

Осадчук В.В.

Прийняв:

Майдан М.В.

Львів 2023

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 №2, для реалізації предметної області заданої варіантом. Суперклас, що реалізований у лабораторній роботі №2, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті Група.Прізвище.Lab3 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
5. Дати відповідь на контрольні запитання.

18 Варіант - Копіювальний апарат

CopyApp.java

```
package KI301_Osadchuk_Lab3;

import java.io.FileNotFoundException;

public class CopyApp {
    public static void main(String[] args) throws FileNotFoundException {
        Copier copier = new Copier();
        copier.connectPowerCordConnector();
        copier.turnOnStartButton();
        copier.turnOnColoredScan();
        copier.putSmthOnTable();
        copier.set_copy(true);
        copier.startCopying(true);
        copier.turnOffColoredScan();
        copier.startCopying(true);
        copier.dispose();
    }
}
```

Copy.java

```
package KI301_Osadchuk_Lab3;

public interface Copy {
    boolean isCopied();
    void startCopying(boolean s_c);
}
```

Copier.java

```
package KI301_Osadchuk_Lab3;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

public class Copier extends Scanner implements Copy{
    private boolean turnOnCopy;
    private boolean isCopied;
    private PrintWriter fout1;

    public Copier() throws FileNotFoundException {
        super();
        fout1 = new PrintWriter(new File("Log.txt"));
    }

    /**
     * Method releases used resources
     */
    public void dispose(){
        fout1.close();
    }

    void set_copy(boolean s_cpy){
        this.turnOnCopy = s_cpy;
    }

    boolean get_copy(){
        return this.turnOnCopy;
    }

    void set_coppeid(boolean s_c) {
        this.isCopied = s_c;
    }

    @Override
    public boolean isCopied() {
        return isCopied;
    }

    @Override
    public void startCopying(boolean s_c) {
        scanning();
        if(get_copy()){
            System.out.print("Start copying...\n");
            System.out.print("Copied\n");
            set_coppeid(s_c);
            fout1.print("Start copying...\n");
            fout1.print("Copied\n");
            fout1.flush();
        }
    }
}
```

```
<terminated> CopyApp [Java Applicatio
The Power Cord is connected
The start button is pressed
The colored scan is on
Something put on table
Colored scanning ...
Scanned
Start copying...
Copied
The colored scan isn't on
White\Black scanning ...
Scanned
Start copying...
Copied
```

Відповіді на КЗ:

1. class Підклас extends Суперклас { Додаткові поля і методи }
2. Найчастіше супер-клас – це базовий клас, а підклас – це похідний клас від суперкласу.
3. 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. Статичне зв'язування використовується при поліморфізмі. (компіляція). Лише тоді, коли метод є приватним, статичним або конструктором.
5. Поліморфізм реалізується за допомогою механізму динамічного (пізнього) зв'язування, який полягає у тому, що вибір методу, який необхідно викликати, відбувається не на етапі компіляції, а під час виконання програми.
6. 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. Використовується для визначення типу об'єкта в момент виконання програми. Посилання на базовий клас.
8. При наслідуванні у Java дозволяється перевизначення (перевантаження) методів та полів. При цьому область видимості методу, що перевизначається, має бути не меншою, ніж область видимості цього методу у суперкласі, інакше компілятор видасть повідомлення, про обмеження привілеїв доступу до даних. Перевизначення методу полягає у визначенні у підкласі методу з сигнатурою методу суперкласу. При виклику такого методу з-під об'єкта підкласу викличеться метод цього підкласу. Якщо ж у підкласі немає визначеного методу, що викликається, то викличеться метод суперкласу. Якщо ж у суперкласі даний метод також відсутній, то згенерується повідомлення про помилку.

9. 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. Синтаксис оголошення інтерфейсів:

```
[public] interface НазваІнтерфейсу {
```

```
Прототипи методів та оголошення констант інтерфейсу
```

```
}
```

Висновок: на цій лабораторній роботі я ознайомився з інтерфейсами, абстрактними класами та реалізував наслідування у мові Java.