

Міністерство освіти і науки України
Національний університет «Львівська Політехніка»

Кафедра ЕОМ



ЗВІТ

з лабораторної роботи №3

з дисципліни: “Кросплатформенні засоби програмування”

на тему: «Спадкування та інтерфейси»

Варіант - 19

Виконав:

ст. гр. КІ-305

Лихограй А. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

Мета роботи: ознайомитися з спадкуванням та інтерфейсами у мові Java.

Завдання:

1. Написати та налагодити програму на мові Java, що розширює клас «Принтер» з лабораторної роботи №2, для реалізації предметної області «Багатофункціональний пристрій». Суперклас «Принтер», 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. Дати відповіді на контрольні запитання:
 - синтаксис реалізації спадкування.
 - що таке суперклас та підклас?
 - як звернутися до членів суперкласу з підкласу?
 - коли використовується статичне зв'язування при виклику методу?
 - як відбувається динамічне зв'язування при виклику методу?
 - що таке абстрактний клас та як його реалізувати?
 - для чого використовується ключове слово instanceof?
 - як перевірити чи клас є підкласом іншого класу?
 - що таке інтерфейс?
 - як оголосити та застосувати інтерфейс?

Вихідний код програми:

До розробленого коду з лабораторної роботи №2 додав файл «*MultiFunction.java*» та змінив «*PrinterDriver.java*».

MultiFunction.java:

```
package KI305.lykhohrai.Lab3;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

interface ScanInterface {
    String scanDocument(String document);
    void printDocument();
};

public class MultifunctionDevice extends Printer implements
ScanInterface{
    public MultifunctionDevice() {
```

```

        super(new InkCartridge(), new PaperTray(), new
NetworkConnection());
    };

    public MultifunctionDevice(String model, InkCartridge inkCartridge,
PaperTray paperTray, NetworkConnection networkConnection, int resource) {
        super(model, inkCartridge, paperTray, networkConnection,
resource);
    };
    @Override public String scanDocument(String document){
        System.out.print("Розпочато сканування " + document + "\n\n");
        try (BufferedReader reader = new BufferedReader(new
FileReader(document))) {
            StringBuilder text = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
                text.append(line).append("\n");
            }
            System.out.print(text.toString());
            return text.toString();
        } catch (IOException e) {
            e.printStackTrace();
            System.out.print( "Помилки при скануванні документу:" +
e.getMessage());
            return e.getMessage();
        }
    };
    public void printDocument(String text) {
        writeLogs("Print document.\n");
        Scanner in = new Scanner(System.in);

        System.out.print("Введіть кількість копій: ");
        int count = in.nextInt();

        if (!isWork() || (count == 0)) {
            System.out.print("Пристрій не ввімкнено!!!\n");
            writeLogs("Printer has problem!\n");
        } else {
            System.out.print("Розпочато друк відсканованого документу:
\n");
            if (GetPaper() >= count && GetInk() > 10) {
                for (int i = 1; i <= count; i++) {
                    System.out.print("\nДрук " + i + " сторінки. \n");
                    System.out.print(text);
                    useInkPaper(text);
                }
            } else
                System.out.print("Замало чорнила або паперу!\n");
        }
    }
}

```

PrinterDriver.java

```
package KI305.lykhohrai.Lab3;
import java.io.*;

public class PrinterDriver {
    public static void main(String[] args) throws IOException {
        //MultifunctionDevice myDevice = new MultifunctionDevice("HP3",
        new InkCartridge(100), new PaperTray(30), new NetworkConnection(), 110);
        MultifunctionDevice myDevice = new MultifunctionDevice();
        myDevice.powerOn();
        myDevice.printDocument(myDevice.scanDocument("MyFile.txt"));
        myDevice.AboutPrinter();
        myDevice.printDocument();
        myDevice.AboutPrinter();
        myDevice.powerOff();
    }
};
```

Результат виконання програми:

```
Введіть IP адрес: 178.212.111.75
Введіть порт: 1024
Пристрій ввімкнено!
Розпочато сканування MyFile.txt

Ввід з текстового файлу
Для введення інформації з файлу необхідно підключити пакет java.io та
створити об'єкт класу Scanner з об'єкту File:
Scanner fin = new Scanner(File("MyFile.txt"));
При компіляції цього рядка коду може виникнути виключна ситуація неіснування
файлу з якого має проходити ввід даних (FileNotFoundException), яка розглядається
компілятором серйозніше, ніж, скажімо, ділення на нуль. Тому, компілятор не дасть нам
просто так скомпільувати цей рядок коду. Для коректної компіляції цього рядку коду
необхідно або вказати, що наш метод може генерувати це виключення, або реалізувати
код для перехоплення і обробки виключення. Для того, щоб вказати, що наш метод може
генерувати це виключення, у оголошенні методу, де відбувається створення об'єкту класу
Scanner з файлу, необхідно додати наступний код:
throws FileNotFoundException
наприклад,
public static void main(String[] args) throws FileNotFoundException
Пошук файлу відбувається у директорії з якої була запущена на виконання
програма. Після відкриття файлу інформацію з нього можна читати використовуючи
методи класу Scanner.
Введіть кількість копій: 1
Розпочато друк відсканованого документу:

Друк 1 сторінки.
Ввід з текстового файлу
```

Рис. 1. Результат роботи програми.

Фрагмент згенерованої документації:

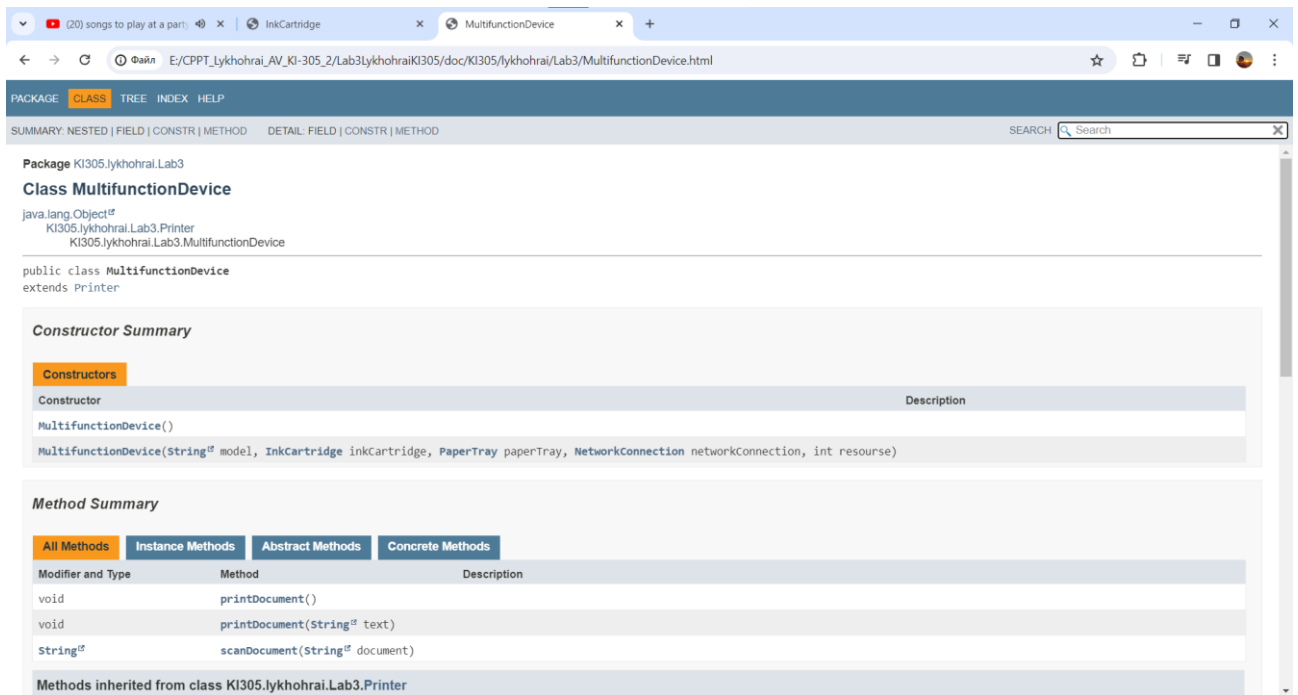


Рис. 2. Фрагмент документації.

Відповіді на контрольні запитання:

1. Синтаксис реалізації спадкування:

```
```java
class Subclass extends Superclass {
 // тіло підкласу
}
```
```

2. Суперклас та підклас:

- Суперклас (батьківський клас): клас, від якого успадковуються властивості та методи.
- Підклас (дочірній клас): клас, який успадковує властивості та методи з іншого класу (суперкласу).

3. Звернення до членів суперкласу з підкласу:

- З використанням ключового слова `super`:

```
```java
super.methodName(); // виклик методу суперкласу
```
```

4. Статичне зв'язування при виклику методу:

- Статичне зв'язування відбувається під час компіляції. Визначення, який метод викликати, відбувається на етапі компіляції.

5. Динамічне зв'язування при виклику методу:

- Динамічне зв'язування відбувається під час виконання програми. Реалізується через механізм поліморфізму.

6. Абстрактний клас та його реалізація:

- Абстрактний клас не може бути інстанційований і може містити абстрактні методи.

- Синтаксис:

```
```java
abstract class AbstractClass {
 abstract void abstractMethod();
}
```
```

7. Використання `instanceof`:

- `instanceof` використовується для перевірки, чи об'єкт є екземпляром певного класу або його підкласу.

8. Перевірка підкласу:

- Використовуючи `instanceof`:

```
```java
if (object instanceof Superclass) {
 // об'єкт є екземпляром суперкласу
}
```
```

9. Інтерфейс:

- Інтерфейс визначає набір методів, але не містить їх реалізації.

- Синтаксис:

```
```java
interface MyInterface {
 void method1();
 void method2();
}
```
```

10. Оголошення та застосування інтерфейсу:

- Оголошення:

```
```java
class MyClass implements MyInterface {
 // реалізація методів інтерфейсу
}
```
```

- Застосування:

```
```java
MyInterface obj = new MyClass();
obj.method1();
```
```

Висновок: під час виконання лабораторної роботи я ознайомився з спадкуванням та інтерфейсами в мові Java. Навчився працювати з ними. Розширив програму з лабораторної роботи №2, додавши клас «MultifunctionDevice», який наслідується від класу «Printer».