Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 2

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «Класи та пакети»

**Виконав:**

студент групи *КІ-306*

*Рудь Володимир*

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

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

**Завдання ( Варіант 18 «Сканер» ):**

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.

**Вихідний код програми**

**Результат виконання програми**

Рис.1 – Скріншот документу згенерованого програмою

**Фрагмент згенерованої документації**

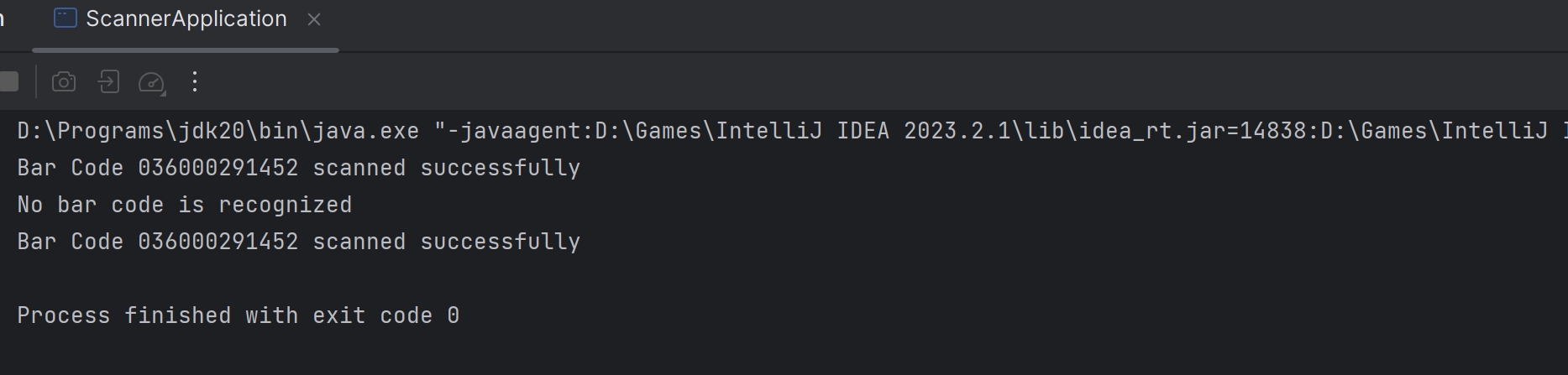
*/\*\*  
 \* Bar Code Scanner Application class implements main method for BarCodeScanner  
 class possibilities demonstration  
 \* @author Rud Volodymyr  
 \* @version 1.0  
 \*/*public class ScannerApplication {  
 */\*\*  
 \* @param args  
 \*/* public static void main(String[] args) {  
 Display tablet = new Display();  
 Camera codeReader = new Camera(1080);  
 BarCodeScanner scannerBar = new BarCodeScanner(tablet, codeReader);  
 scannerBar.ScanBarCode("036000291452");  
 scannerBar.ScanBarCode("000001000000");  
 scannerBar.ScanBarCode("036000291452");  
  
 }  
}

import java.io.File;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.Arrays;  
  
*/\*\*  
 \* Class <code>Display</code> implements display and AutoCloseable  
 \* @author Volodymyr Rud  
 \* @version 1.0  
 \*/*public class Display implements AutoCloseable {  
 //file to write  
 private PrintWriter fout;  
  
 */\*\*  
 \* Constructor initializes file writter variable  
 \*/* Display() {  
 try {  
 fout = new PrintWriter(new File("DisplayFile.txt"));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* Method print in file and in console result of calculating of bar code  
 \*/* public void WriteBarCodeScanResult(boolean res, String code) {  
 if (res) {  
 System.*out*.println("Bar Code " + code + " scanned successfully");  
 fout.println("Bar Code " + code + " scanned successfully");  
 fout.flush();  
 } else {  
 System.*out*.println("Bar Code " + code + " is incorrect");  
 fout.println("Bar Code " + code + " is incorrect");  
 fout.flush();  
 }  
 }  
  
 */\*\*  
 \* Method closes the file  
 \* @throws Exception  
 \*/* @Override  
 public void close() throws Exception {  
 if (fout != null) {  
 fout.flush();  
 fout.close();  
 }  
 }  
  
 */\*\*  
 \* Method prints that bar code is not recognized  
 \*/* public void noCodeRecognise() {  
 System.*out*.println("No bar code is recognized");  
 fout.println("No bar code is recognized");  
 fout.flush();  
 }  
}

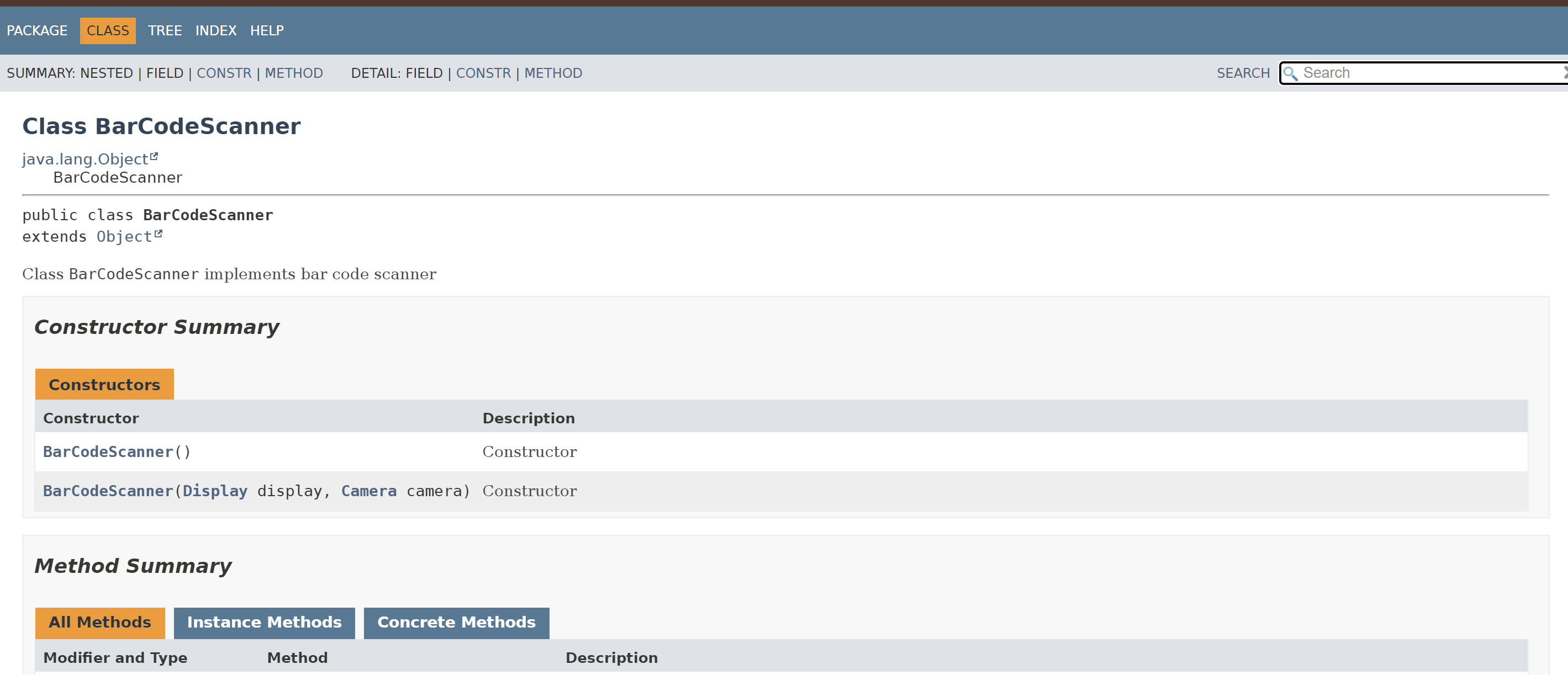
import java.util.List;  
import java.util.Random;  
  
*/\*\*  
 \* Class <code>Camera</code> implements camera  
 \* @author Volodymyr Rud  
 \* @version 1.0  
 \*/*public class Camera {  
 //resolution of camera  
 private int resolution;  
  
 */\*\*  
 \* Constructor  
 \* @param resolution  
 \*/* public Camera(int resolution) {  
 setResolution(resolution);  
  
 }  
 */\*\*  
 \* Method that checks on correctness of parameters to set resolution  
 \* @param resolution  
 \*/* public void setResolution(int resolution) {  
 if (resolution > 0) {  
 this.resolution = resolution;  
 }  
 }  
  
 */\*\*  
 \* Method that returns resolution of camera  
 \* @return resolution of camera  
 \*/* public int getResolution() {  
 return resolution;  
 }  
  
 */\*\*  
 \* Method that returns true 7 in 10 chance  
 \* @return true 7 in 10 chance  
 \*/* public boolean captureImage() {  
 Random random = new Random();  
 int randomNumber = random.nextInt(10);  
  
 return randomNumber < 8;  
 }  
}

import java.lang.reflect.Array;  
import java.util.Arrays;  
  
*/\*\*  
 \* Class <code>BarCode</code> implements bar code  
 \* @author Volodymyr Rud  
 \* @version 1.0  
 \*/*public class BarCode {  
 //array of digits of bar code  
 private int[] barCode;  
  
 */\*\*  
 \* Constructor by default. initialize empty array  
 \*/* BarCode() {  
 barCode = new int[12];  
 }  
  
 */\*\*  
 \*  
 \* @param barCode  
 \* Constuctor  
 \*/* public BarCode(int [] barCode) {  
 setBarCode(barCode);  
 }  
  
 */\*\*  
 \* Method returns array of numbers barCode  
 \* @return array of numbers barCode  
 \*/* public int[] getBarCode() {  
 return barCode;  
 }  
  
 */\*\*  
 \* Method that returns true or false if bar code is calculated correct  
 \* @param code  
 \* @return true or false if bar code is calculated correct  
 \*/* public boolean setBarCode(int[] code) {  
 int sumOdd = 0;  
 int sumEven = 0;  
  
 for (int i = 0; i < 12; i++) {  
 int digit = code[i];  
  
 if ((i + 1) % 2 == 0) {  
 sumEven += digit;  
 } else {  
 sumOdd += digit;  
 }  
 }  
  
 int totalSum = sumOdd + (sumEven \* 3);  
 int calcResult = (10 - (totalSum % 10)) % 10;  
 int checkDigit = code[11];  
  
 if (calcResult == checkDigit) {  
 this.barCode = Arrays.*copyOf*(code, 12);  
 return true;  
 } else  
 return false;  
 }  
}

import java.util.ArrayList;  
import java.util.List;  
  
  
*/\*\*  
 \* Class <code>BarCodeScanner</code> implements bar code scanner  
 \* @author Volodymyr Rud  
 \* @version 1.0  
 \*/*public class BarCodeScanner {  
 private BarCode scannedBarCode;  
 private final Display display;  
 private final Camera camera;  
  
 */\*\*  
 \* Constructor  
 \*/* public BarCodeScanner()  
 {  
 scannedBarCode = new BarCode();  
 display = new Display();  
 camera = new Camera(360);  
 }  
  
 */\*\*  
 \* Constructor  
 \* @param display <Display>display</Display>  
 \* @param camera <Camera>camera</Camera>  
 \*/* public BarCodeScanner(Display display, Camera camera)  
 {  
 this.scannedBarCode = new BarCode();  
 this.display = display;  
 this.camera = camera;  
 }  
  
 */\*\*  
 \* Method that returns scannedBarCode  
 \* @return scannedBarCode  
 \*/* public BarCode getScannedBarCode()  
 {  
 return scannedBarCode;  
 }  
  
 */\*\*  
 \* Method that returns true if process of scanning bar code has executed without exceptions  
 \* @param code  
 \* @return true if process of scanning bar code has executed without exceptions  
 \*/* public boolean ScanBarCode(String code)  
 {  
 int[] temp = new int[12];  
 if(camera.captureImage()) {  
 if (code.length() != 12) {  
 display.noCodeRecognise();  
 return false;  
 }  
 for (int i = 0; i < 12; i++) {  
 temp[i] = Character.*getNumericValue*(code.charAt(i));  
 }  
 scannedBarCode = new BarCode();  
 if (!scannedBarCode.setBarCode(temp)) {  
 display.WriteBarCodeScanResult(false, code);  
 return false;  
 }  
 else{  
 display.WriteBarCodeScanResult(true, code);  
 return true;  
 }  
  
 }  
 else {  
 display.noCodeRecognise();  
 return false;  
 }  
 }  
}

****

**Рис.1 Результат виконання програми**

****

**Рис.2 Згенерована документація**

**Відповіді на контрольні запитання**

1. Синтаксис визначення класу.

public class ClassName {

// Код класу тут

}

1. Синтаксис визначення методу.

returnType methodName(parameterType parameterName) {

// Тіло методу

}

1. Синтаксис оголошення поля.

dataType fieldName;

1. Як оголосити та ініціалізувати константне поле?

final dataType fieldName = value;

1. Які є способи ініціалізації полів?

конструктор, ініціалізація при оголошенні, блоки ініціалізації.

1. Синтаксис визначення конструктора.

public ClassName(constructorParameters) {

// Тіло конструктора

}

1. Синтаксис оголошення пакету.

package packageName;

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

import packageName.ClassName;

1. В чому суть статичного імпорту пакетів?

Статичний імпорт пакетів дозволяє використовувати статичні члени класу без префіксу

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

Вимоги до файлів і каталогів у використанні пакетів: файл класу повинен зберігатися в каталозі, що відповідає імені пакету, і шлях до нього повинен відповідати його імені простору імен.

**Висновок**

Ознайомився з роботою класів та пакетів у мові програмування Java. Розробив клас який реалізує предметну область «Локомотив», та програму драйвер яка перевіряє роботу класу.