Національний університет «Львівська Політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Про виконання лабораторної роботи №2

### З дисципліни «Кросплатформні засоби програмування»

**Виконав:**

студент групи КІ-304

Слободян Я. Р.

**Прийняв:**

викладач

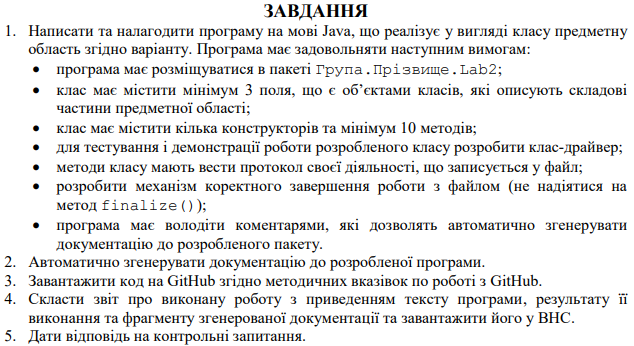
Олексів М. В.

Львів – 2023

**Тема:** класи та пакети.

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

Варіант №19

****

**19. Принтер**

**Виконання**

**Код програми:**

App.java

public class App {

public static void main(String[] args) throws Exception {

Printer printer = new Printer();

printer.turnOn();

printer.connectPrinterToDaInternet("", "", 2);

printer.hastyFillUp();

printer.getInkInfo();

printer.getInternetInfo();

printer.getPaperTrayInfo();

printer.printViaInternet(5);

printer.printViaFlsh(100);

printer.turnOff();

printer.dispose();

}

}

Printer.java

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

/\*\*

\* Class <code>Printer</code> implements the printer

\*/

public class Printer {

private Ink ink;

private InternetAdapter internetAdapter;

private PaperTray paperTray;

private PrintWriter file;

public boolean isEnabled;

/\*\*

\* Constructor

\*/

public Printer() throws FileNotFoundException{

this.ink = new Ink();

this.internetAdapter = new InternetAdapter();

this.paperTray = new PaperTray();

this.isEnabled = false;

this.file = new PrintWriter(new File("Log.txt"));

}

/\*\*

\* Constructor

\*@param <code>redP</code> The persentage of red ink

\*@param <code>greenP</code> The persentage of red ink

\*@param <code>blueP</code> The persentage of red ink

\*@param <code>CType</code> index of the connection type

\*@param <code>IsC</code> bool, if connected is true

\*@param <code>CStrength</code> strength of the connection

\*@param <code>ssid</code> ssid, if WiFi

\*@param <code>pass</code> password, if WiFi

\*@param <code>ptIndex</code> index of the paper type

\*@param <code>pAmount</code> amount of paper in the tray

\*@param <code>maxPAmount</code> maximal amount of paper in the tray

\*/

public Printer(float redP, float greenP, float blueP, int CType, boolean IsC, int CStrength, String ssid, String pass, int ptIndex, int pAmount, int maxPAmount) throws FileNotFoundException{

this.ink = new Ink(redP, greenP, blueP);

this.internetAdapter = new InternetAdapter(CType, IsC, CStrength, ssid, pass);

this.paperTray = new PaperTray(ptIndex, pAmount, maxPAmount);

this.isEnabled = false;

this.file = new PrintWriter(new File("Log.txt"));

}

/\*\*

\* Method turns the printer on

\*/

public void turnOn() {

isEnabled = true;

file.println("You succesfully turned your printeron. Great job, champ!");

file.println();

file.flush();

}

/\*\*

\* Method turns the printer off

\*/

public void turnOff() {

isEnabled = false;

file.println(internetAdapter.disconnect());

file.println("You succesfully turned your printer off.");

file.println();

file.flush();

}

/\*\*

\* Method gets information about inks of the printer

\*/

public void getInkInfo(){

file.println("Ink levels of your printer are:");

file.println("Red: " + ink.checkRedInkLevel()+"%");

file.println("Green: "+ ink.checkGreenInkLevel()+"%");

file.println("Blue: "+ ink.checkBlueInkLevel()+"%");

file.println();

file.flush();

}

/\*\*

\* Method gets information about internet connection of the printer

\*/

public void getInternetInfo(){

if(internetAdapter.checkIfConnected()){

file.println("You are connected to the internet via " + internetAdapter.checkConnectionType()+".");

if(internetAdapter.checkWifiSSID() != ""){

file.println("SSID: "+internetAdapter.checkWifiSSID());

file.println("Password: "+internetAdapter.checkWifiPassword());

}

file.println("Connection strength: "+internetAdapter.checkConnectionStrength());

}

else{

file.println("You are not connected to internet.");

}

file.println();

file.flush();

}

/\*\*

\* Method gets information about the paper tray of the printer

\*/

public void getPaperTrayInfo(){

file.println("As of now, in your paper tray rests " + paperTray.checkPaperAmount() + " sheets of paper out of " + paperTray.checkMaxPaperAmount() + " possible.");

file.println("The format of the paper is " + paperTray.checkPaperType() + ".");

file.println();

file.flush();

}

/\*\*

\* Method fills up printer to the brim

\*/

public void hastyFillUp(){

file.println("You decided to try and place into your printer maximal amounts of ink and paper, no matter the consequences.");

file.println("...");

file.println(ink.fillRedInk(100));

file.println("...");

file.println(ink.fillGreenInk(100));

file.println("...");

file.println(ink.fillBlueInk(100));

file.println("...");

file.println("...");

file.println(paperTray.changePaperAmount(paperTray.checkMaxPaperAmount()));

file.println();

file.flush();

}

/\*\*

\* Method fills up printer with set amount of paper and inks

\*@param rAmount amount of red ink

\*@param gAmount amount of green ink

\*@param bAmount amount of blue ink

\*@param pTypeIndex index of the paper type

\*@param pAmount amount of the paper

\*/

public void toughtfulFillUp(float rAmount, float gAmount, float bAmount, int pTypeIndex, int pAmount){

file.println("You decided to try and place into your printer ink and paper, paying attention to what are you doing.");

file.println("...");

file.println(ink.fillRedInk(rAmount));

file.println("...");

file.println(ink.fillGreenInk(gAmount));

file.println("...");

file.println(ink.fillBlueInk(bAmount));

file.println("...");

file.println(paperTray.changePaperType(pTypeIndex));

file.println("...");

file.println(paperTray.changePaperAmount(pAmount));

file.println();

file.flush();

}

/\*\*

\* Method connects printer tto the internet, either via WiFi or Ethernet

\*@param ssid SSID, if WiFi

\*@param pass Password, if WiFi

\*@param connectionTypeInex index of the connection type

\*/

public void connectPrinterToDaInternet(String ssid, String pass, int connectionTypeInex){

file.println("You decided to try and connect your printer to the internet.");

file.println("...");

if(connectionTypeInex == 1){

file.println(internetAdapter.connectToWifi(ssid, pass));

}

else if(connectionTypeInex == 2){

file.println(internetAdapter.connectToEthernet());

}

file.println();

file.flush();

}

/\*\*

\* Method tries to print something via internet

\* @param sheets amount of printed sheets

\*/

public void printViaInternet(int sheets){

file.println("You decided to try to print some papers on your printer via the internet.");

file.println("...");

if(internetAdapter.checkIfConnected()){

if(paperTray.checkPaperAmount() < sheets || ink.checkRedInkLevel() < sheets || ink.checkGreenInkLevel() < sheets || ink.checkBlueInkLevel() < sheets){

file.println("You failed! Your printer dont have something!");

}

else{

file.println("Your prints came out nice.");

file.println(paperTray.changePaperAmount(-sheets));

}

}

else{

file.println("You failed! Your printer is not connected to da internet!");

}

file.println();

file.flush();

}

/\*\*

\* Method tries to print something from the Flash drive

\* @param sheets amount of printed sheets

\*/

public void printViaFlsh(int sheets){

file.println("You decided to try to print some papers on your printer via Flash drive.");

file.println("...");

if(paperTray.checkPaperAmount() < sheets || ink.checkRedInkLevel() < sheets || ink.checkGreenInkLevel() < sheets || ink.checkBlueInkLevel() < sheets){

file.println("You failed! Your printer dont have something!");

}

else{

file.println("Your prints came out nice.");

file.println(paperTray.changePaperAmount(-sheets));

}

file.println();

file.flush();

}

public void dispose() {

file.close();

}

}

Ink.java

/\*\*

\* Class <code>Ink</code> implements inks of printer

\*/

public class Ink {

private float redPersentage;

private float greenPersentage;

private float bluePersentage;

/\*\*

\* Constructor

\*/

public Ink(){

this.redPersentage = 100;

this.greenPersentage = 100;

this.bluePersentage = 100;

}

/\*\*

\* Constructor

\* @param <code>redP</code> The persentage of red ink

\* @param <code>greenP</code> The persentage of red ink

\* @param <code>blueP</code> The persentage of red ink

\*/

public Ink(float redP, float greenP, float blueP){

this.redPersentage = redP;

this.greenPersentage = greenP;

this.bluePersentage = blueP;

}

/\*\*

\* Method returns the persentage of red ink

\* @return The persentage of red ink

\*/

public float checkRedInkLevel(){

return redPersentage;

}

/\*\*

\* Method returns the persentage of green ink

\* @return The persentage of red ink

\*/

public float checkGreenInkLevel(){

return greenPersentage;

}

/\*\*

\* Method returns the persentage of blue ink

\* @return The persentage of red ink

\*/

public float checkBlueInkLevel(){

return bluePersentage;

}

/\*\*

\* Method adds up red ink

\* @param amount amount of ink pouring

\* @return consequences of your actions

\*/

public String fillRedInk(float amount){

if(amount <= 0){

return "You poured in exactly nothing.\nRed ink level stayed on " + checkRedInkLevel() + " persents.";

}

redPersentage = redPersentage + amount;

if(redPersentage > 100){

float diff = redPersentage - 100;

redPersentage = 100;

return "You poured in too much and " + diff + " persents of ink are now spilled!\nRed ink reservoir is full.";

}

else{

return "You poured in " + amount + " persents of ink.\nRed ink reservoir contains " + checkRedInkLevel() + " persents of ink.";

}

}

/\*\*

\* Method adds up green ink

\* @param amount amount of ink pouring

\* @return consequences of your actions

\*/

public String fillGreenInk(float amount){

if(amount <= 0){

return "You poured in exactly nothing.\nGreen ink level stayed on " + checkGreenInkLevel() + " persents.";

}

greenPersentage = greenPersentage + amount;

if(greenPersentage > 100){

float diff = greenPersentage - 100;

greenPersentage = 100;

return "You poured in too much and " + diff + " persents of ink are now spilled!\nGreen ink reservoir is full.";

}

else{

return "You poured in " + amount + " persents of ink.\nGreen ink reservoir contains " + checkGreenInkLevel() + " persents of ink.";

}

}

/\*\*

\* Method adds up blue ink

\* @param amount amount of ink pouring

\* @return consequences of your actions

\*/

public String fillBlueInk(float amount){

if(amount <= 0){

return "You poured in exactly nothing.\nBlue ink level stayed on " + checkBlueInkLevel() + " persents.";

}

bluePersentage = bluePersentage + amount;

if(bluePersentage > 100){

float diff = bluePersentage - 100;

bluePersentage = 100;

return "You poured in too much and " + diff + " persents of ink are now spilled!\nBlue ink reservoir is full.";

}

else{

return "You poured in " + amount + " persents of ink.\nBlue ink reservoir contains " + checkBlueInkLevel() + " persents of ink.";

}

}

}

InternetAdapter.java

import java.util.Random;

/\*\*

\* Class <code>InternetAdapter</code> implements paper tray of printer

\*/

public class InternetAdapter {

enum ConnectionTypes{

NOT\_CONNECTED, WIFI, ETHERNET

}

Random rand = new Random();

private ConnectionTypes connectionType;

private boolean isConnected;

private int connectionStrength;

private String wifiSSID;

private String wifiPassword;

/\*\*

\* Constructor

\*/

public InternetAdapter() {

this.connectionType = ConnectionTypes.NOT\_CONNECTED;

this.isConnected = false;

this.connectionStrength = 0;

this.wifiSSID = "";

this.wifiPassword = "";

}

/\*\*

\* Constructor

\*@param <code>CType</code> index of the connection type

\*@param <code>IsC</code> bool, if connected is true

\*@param <code>CStrength</code> strength of the connection

\*@param <code>ssid</code> ssid, if WiFi

\*@param <code>pass</code> password, if WiFi

\*/

public InternetAdapter(int CType, boolean IsC, int CStrength, String ssid, String pass){

this.connectionType = ConnectionTypes.values()[CType];

this.isConnected = IsC;

this.connectionStrength = CStrength;

this.wifiSSID = ssid;

this.wifiPassword = pass;

}

/\*\*

\* Method returns the connection type

\* @return The connection type

\*/

public ConnectionTypes checkConnectionType(){

return connectionType;

}

/\*\*

\* Method returns if printer connected to the internet

\* @return If printer connected to the internet

\*/

public boolean checkIfConnected(){

return isConnected;

}

/\*\*

\* Method returns strength of the connection

\* @return The strength of connection to the internet

\*/

public int checkConnectionStrength(){

return connectionStrength;

}

/\*\*

\* Method returns ssid of the WiFi

\* @return The ssid of the WiFi

\*/

public String checkWifiSSID(){

return wifiSSID;

}

/\*\*

\* Method returns password of the WiFi

\* @return The password of the WiFi

\*/

public String checkWifiPassword(){

return wifiPassword;

}

/\*\*

\* Method connects internet adapter to the WiFi network

\* @param ssid SSID

\* @param pass Password

\* @return consequences of your actions

\*/

public String connectToWifi(String ssid, String pass){

isConnected = true;

connectionType = ConnectionTypes.WIFI;

connectionStrength = rand.nextInt(101);

wifiSSID = ssid;

wifiPassword = pass;

return "Your printer succesfully connected to the WiFi '"+wifiSSID+"' with password '"+wifiPassword+"' and strength of the signal of "+connectionStrength+" percent.";

}

/\*\*

\* Method connects internet adapter to the Ethernet cable

\* @return consequences of your actions

\*/

public String connectToEthernet(){

isConnected = true;

connectionType = ConnectionTypes.ETHERNET;

connectionStrength = 100;

wifiSSID = "";

wifiPassword = "";

return "Your printer succesfully connected to the Ethernet cable with strength of the signal of "+connectionStrength+" percent. Nice choice!";

}

/\*\*

\* Method disconnects internet adapter from the internet

\* @return consequences of your actions

\*/

public String disconnect(){

connectionType = ConnectionTypes.NOT\_CONNECTED;

isConnected = false;

connectionStrength = 0;

wifiSSID = "";

wifiPassword = "";

return "You disconnected your printer from any sources of internet. Now it cannot be hacked!";

}

}

PaperTray.java

/\*\*

\* Class <code>PaperTray</code> implements paper tray of printer

\*/

public class PaperTray {

enum PaperTypes{

A1, A2, A3, A4, A5

};

private PaperTypes paperType;

private int paperAmount;

private int maxPaperAmount;

/\*\*

\* Constructor

\*/

public PaperTray(){

this.paperType = PaperTypes.A4;

this.paperAmount = 1;

this.maxPaperAmount = 100;

}

/\*\*

\* Constructor

\*@param <code>ptIndex</code> index of the paper type

\*@param <code>pAmount</code> amount of paper in the tray

\*@param <code>maxPAmount</code> maximal amount of paper in the tray

\*/

public PaperTray(int ptIndex, int pAmount, int maxPAmount){

this.paperType = PaperTypes.values()[ptIndex - 1];

this.paperAmount = pAmount;

this.maxPaperAmount = maxPAmount;

}

/\*\*

\* Method returns the amount of paper in the tray

\* @return The amount of paper in the tray

\*/

public int checkPaperAmount(){

return paperAmount;

}

/\*\*

\* Method returns the type of paper in the tray

\* @return The type of paper in the tray

\*/

public PaperTypes checkPaperType(){

return paperType;

}

/\*\*

\* Method returns the maximal amount of paper in the tray

\* @return The maximal amount of paper in the tray

\*/

public int checkMaxPaperAmount(){

return maxPaperAmount;

}

/\*\*

\* Method changes paper type in the tray

\* @param ptIndex index of the paper type

\* @return consequences of your actions

\*/

public String changePaperType(int ptIndex){

if(ptIndex < 0 || ptIndex > PaperTypes.values().length){

return "You tried to shove a paper with an invalid format into your printer!\nYou failed.";

}

else{

paperType = PaperTypes.values()[ptIndex - 1];

paperAmount = 1;

return "You plased one sheet of " + checkPaperType() + " paper into your printers paper tray.";

}

}

/\*\*

\* Method changes amount of paper in the tray

\* @param amount amount of the paper

\* @return consequences of your actions

\*/

public String changePaperAmount(int amount){

if(amount > 0){

if(amount + paperAmount > maxPaperAmount){

paperAmount = maxPaperAmount;

return "You tried to shove too much paper in your printers paper tray!\nAll excess paper is now cowering your rooms floor.\nYour printers tray is full, containing " +checkPaperAmount()+" sheets of paper";

}

else{

paperAmount = paperAmount + amount;

return "Your printers tray is containing " +checkPaperAmount()+" sheets of paper now.";

}

}

else if(amount < 0){

if(amount + paperAmount < 0){

paperAmount = 0;

return "You tried to take nonexisting papers from your printers tray, you schizo! After all REAL paper was taken, your printers tray is empty, containing " +checkPaperAmount()+" sheets of paper";

}

else{

paperAmount = paperAmount + amount;

return "Your printers tray is containing " +checkPaperAmount()+" sheets of paper now.";

}

}

else{

return "You stared on your paper tray.\n It stared back.";

}

}

}

**Результати роботи в текстовому файлі:**

**Log.txt**

You succesfully turned your printeron. Great job, champ!

You decided to try and connect your printer to the internet.

...

Your printer succesfully connected to the Ethernet cable with strength of the signal of 100 percent. Nice choice!

You decided to try and place into your printer maximal amounts of ink and paper, no matter the consequences.

...

You poured in too much and 100.0 persents of ink are now spilled!

Red ink reservoir is full.

...

You poured in too much and 100.0 persents of ink are now spilled!

Green ink reservoir is full.

...

You poured in too much and 100.0 persents of ink are now spilled!

Blue ink reservoir is full.

...

...

You tried to shove too much paper in your printers paper tray!

All excess paper is now cowering your rooms floor.

Your printers tray is full, containing 100 sheets of paper

Ink levels of your printer are:

Red: 100.0%

Green: 100.0%

Blue: 100.0%

You are connected to the internet via ETHERNET.

Connection strength: 100

As of now, in your paper tray rests 100 sheets of paper out of 100 possible.

The format of the paper is A4.

You decided to try to print some papers on your printer via the internet.

...

Your prints came out nice.

Your printers tray is containing 95 sheets of paper now.

You decided to try to print some papers on your printer via Flash drive.

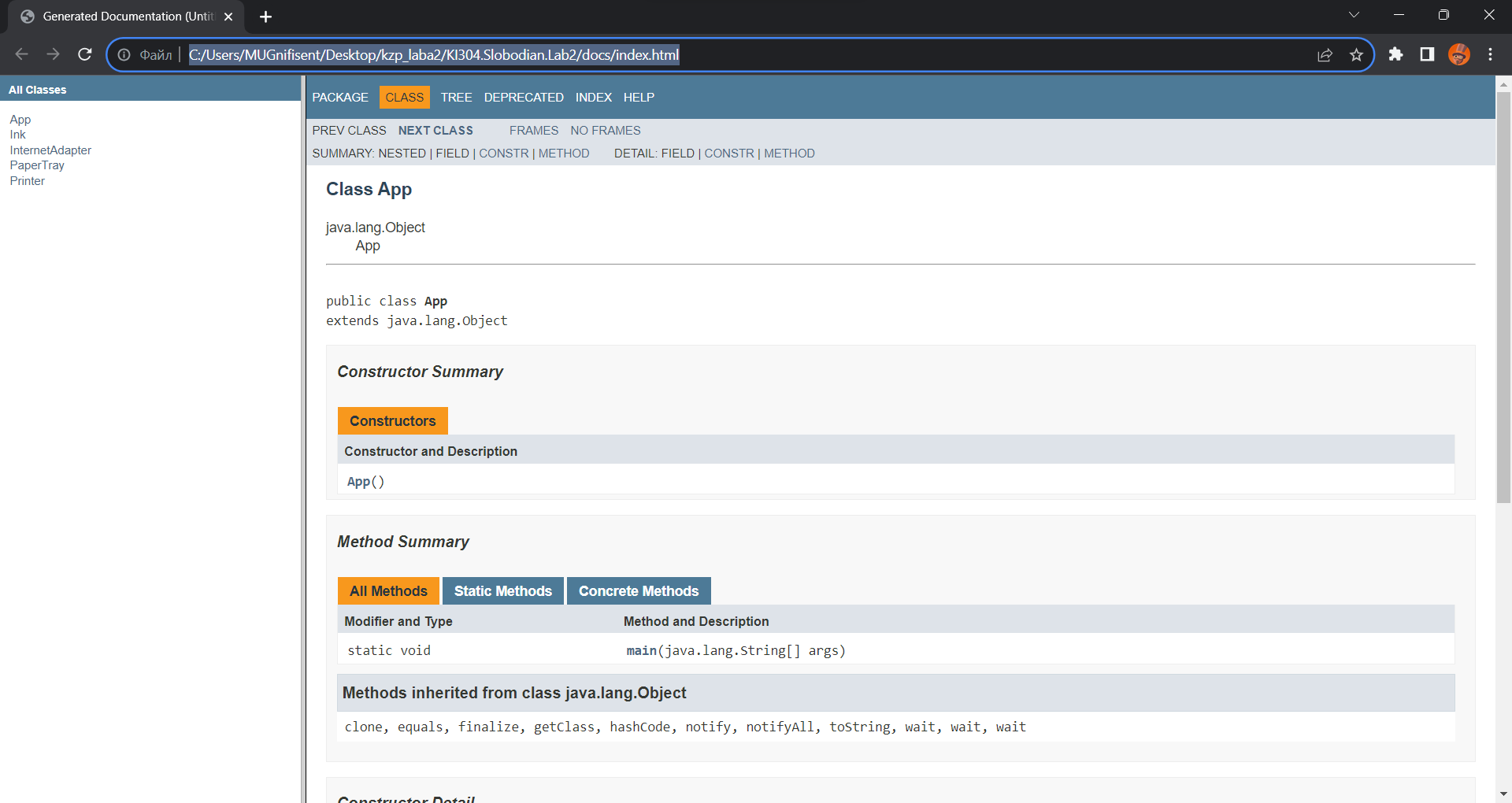
...

You failed! Your printer dont have something!

You disconnected your printer from any sources of internet. Now it cannot be hacked!

You succesfully turned your printer off.

**Документація:**

****

**Висновок:** на цій лабораторній роботі я ознайомився з процесом розробки класів та пакетів мовою Java. Також розробив клас, який описує роботу телевізора, та продемонстрував його принцип роботи.