МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки

**Лабораторна робота**

з Кросплатформенні технології

(назва дисципліни)

на тему: «Розроблення найпростішого Java-застосунку для обробки аудіо-даних»

Виконав: студент 4 курсу групи № 545б

напряму підготовки (спеціальності)

123 – комп’ютерна інженерія

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва напряму підготовки (спеціальності))

Тимошенко О.О.

(прізвище й ініціали студента)

Прийняв: асистент каф.503

Годованюк П.А.

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Харків – 2020

1. Постановка мети

1.1 Ознайомитися з принципами розробки найпростіших Java-застосунків ;

1.2 Навчиться використовувати елементарні типи в програмах на мові Java;

1.3 Навчиться використовувати керуючі оператори в програмах на мові Java.

1. Постановка завдання роботи

Реалізувати Java-програму, яка обчислює фізичні характеристики цифрового сигналу, наведені в розділі коротких теоретичних відомостей. Відлік цифрового сигналу (цілі числа, які змінюються в діапазоні 0-255) зберігаються в файлі, шлях і ім'я якого повинні передаватися програмі через командний рядок. Функції обчислення перерахованих фізичних характеристик повинні бути реалізовані у вигляді бібліотеки, що підключається. Результати розрахунку характеристик виводяться в консоль (функцію автокореляції будувати не потрібно).

Мій варіант 13, тому мені потрібно використовувати файл 13.bin.

1. Вихідний код програми

Клас Main.java

package lab3.src;

import lab3.lib.\*;

/\*\*

\* Класс предавляющий точку входа в приложение,

\* содержит метод main

\*/

public class Main {

public static void main(String[] args) {

try {

int tau = 13; // this is my variant

// create field

FileReader fr = new FileReader(args[0]);

SignalParams sigParams;

// read bytes from file

byte[] signalVals = fr.getBytes();

if (signalVals == null) {

System.out.println("Can`t read signal values from file " + args[1]);

return;

}

// calculate params

sigParams = new SignalParams(signalVals);

System.out.println(sigParams.toStringValues(tau));

}

catch (IndexOutOfBoundsException e) {

System.out.println("You didnt pass an argument(file name)");

}

catch (Exception e) {

System.out.println("Unexpected error\n" + e.getMessage());

}

}

}

Клас FileReader.java

package lab3.src;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

/\*\*

\* Класс для чтения массива байт из файла

\*/

public class FileReader {

private String \_path;

/\*\*

\* Конструктор для создания объект для чтения байтов

\* из заданного в параметре файла

\*

\* @param path путь к файлу

\*/

public FileReader(String path) {

\_path = path;

}

/\*\*

\* Читает все содержимое файла и возращает это содержимое

\* как массив байтов

\*

\* @return содержимое файла в виде массива байтов

\*/

public byte[] getBytes() {

FileInputStream fis = null;

try {

fis = new FileInputStream(\_path);

return fis.readAllBytes();

}

catch (FileNotFoundException e) {

System.out.println(e.getMessage());

}

catch (IOException e) {

System.out.println(e.getMessage());

}

finally {

if(fis != null) {

try {

fis.close();

}

catch (IOException e) {

System.out.println(e.getMessage());

}

}

}

return null;

}

}

Клас SignalParams.java

package lab3.lib;

import java.util.ArrayList;

/\*\*\*

\* @Author: Oleh Tymoshenko

\*/

/\*\*

\* Класс для вычисления значений параметров сигнала

\*/

public class SignalParams {

private byte[] \_signal;

/\*\*

\* Создание объекта для вычисления значение сигнала

\* на основе массива байтов

\*

\* @param signal массив байтов, который представляет значения

\* отсчетов сигнала в диапазоне [0;255]

\*/

public SignalParams(byte[] signal) {

\_signal = signal;

}

/\*\*

\* Минимальное значение среди отсчетов в сигнале

\*

\* @return минимальный отсчет в сигнале

\*/

public byte signalMinVal()

{

byte minValInSignal = \_signal[0];

for(int i = 1; i < \_signal.length; i++){

if(minValInSignal > \_signal[i])

minValInSignal = \_signal[i];

}

return minValInSignal;

}

/\*\*

\* Максимальное значение среди отсчетов в сигнале

\*

\* @return максимальный отсчет в сигнале

\*/

public byte signalMaxVal()

{

byte maxValInSignal = \_signal[0];

for(int i = 1; i < \_signal.length; i++){

if(maxValInSignal < \_signal[i])

maxValInSignal = \_signal[i];

}

return maxValInSignal;

}

/\*\*

\* Динамический диапазон сигнала

\*

\* @return значение динамического диапазонана сигнала

\*/

public byte signalDynamicRange() {

return (byte) (signalMaxVal() - signalMinVal());

}

/\*\*

\* Энергия сигнала

\*

\* @return значение энергии сигнала

\*/

public double signalEnergy() {

double res = 0;

for (byte i:

\_signal) {

res = res + (Math.pow(i, 2));

}

return res;

}

/\*\*

\* Средняя мощность сигнала

\*

\* @return средняя мощность сигнала

\*/

public double signalAvgPower() {

return (signalEnergy()/\_signal.length);

}

/\*\*

\* Среднее значеие отсчетов сигнала

\*

\* @return среднее значение отсетов сигнала

\*/

public double signalAvgValue() {

double res = 0;

for (byte i:

\_signal) {

res += i;

}

return res/\_signal.length;

}

/\*\*

\* Дисперсия значений отсчетов сигнала

\*

\* @return дисперсия значений отсчетов

\*/

public double signalDispersion() {

double signalAvgVal = signalAvgValue();

double res = 0;

for (byte i:

\_signal) {

res += Math.pow((i - signalAvgVal), 2);

}

return res/\_signal.length;

}

/\*\*

\* Функция автокорреляции

\*

\* @param tau - тау для функции автокорреляции

\* @return значение функции автокорреляции для заданного тау

\*/

public double signalAutocorrelationFunc(int tau) {

double avgSignalVal = signalAvgValue();

double res = 0;

if(tau < 0) {

res = signalAutocorrelationFunc(0-tau);

}

else{

double sum = 0;

for(int i = 0; i < \_signal.length - tau; i++) {

sum += ((\_signal[i+tau] - avgSignalVal)\*

(\_signal[i] - avgSignalVal));

}

res = sum / (\_signal.length - tau);

}

return res;

}

/\*\*

\* Интервал кореляции

\*

\* @return интервал кореляции

\*/

public double signalCIntervalCorrelation() {

double res = 0;

double numerator = 0;

for(int i = 0; i < \_signal.length; i++) {

numerator += signalAutocorrelationFunc(i);

}

res = numerator / signalAutocorrelationFunc(0);

return Math.ceil(Math.abs(res));

}

/\*\*

\* Возвращает все параметры в удобочитаемом виде

\*

\* @return строка, представляющая все параметры

\*/

public String toStringValues(int tau) {

String resStr = "Мінімальне значення відліка в сигналі: " + signalMinVal() + "\n" +

"Максимальне значення відліка в сигналі: " + signalMaxVal() + "\n" +

"Динамічний діапазон сигналу: " + signalDynamicRange() + "\n" +

"Енергія сигналу: " + signalEnergy() + "\n" +

"Середня потужність сигналу: " + signalAvgPower() + "\n" +

"Середнє значення відліків сигналу: " + signalAvgValue() + "\n" +

"Дисперсія значень відліків сигналу: " + signalDispersion() + "\n" +

"Функція автокореляці дискретного сигналу (тау = " + tau + "): " + signalAutocorrelationFunc(tau) + "\n" +

"Інтервал кореляції: " + signalCIntervalCorrelation() + "\n";

return resStr;

}

}

1. Результат

Структура проекту в IDE представлена на рис. 1.

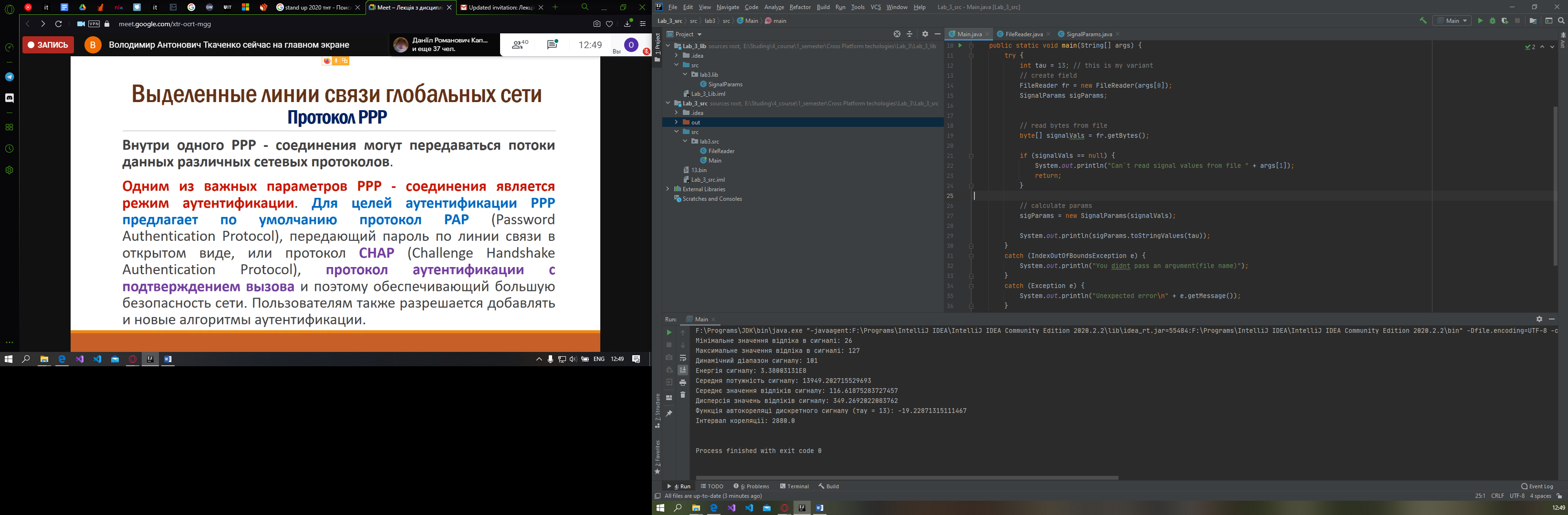


Рисунок 1 – Структура проекту

Налаштування для виконання програми (встановлення аргументу для програми) представлено на рис. 2.

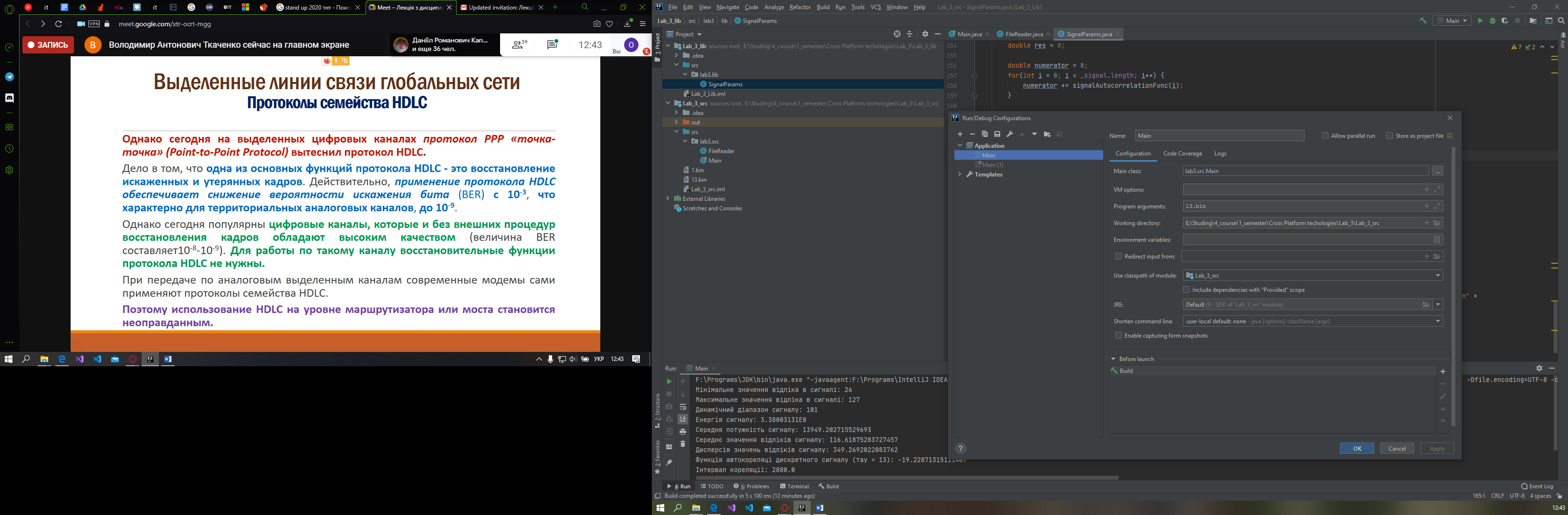


Рисунок 2 – Встановлення вхідних аргументів для додатку

Результат виконання програми представлено на рисунку 3.

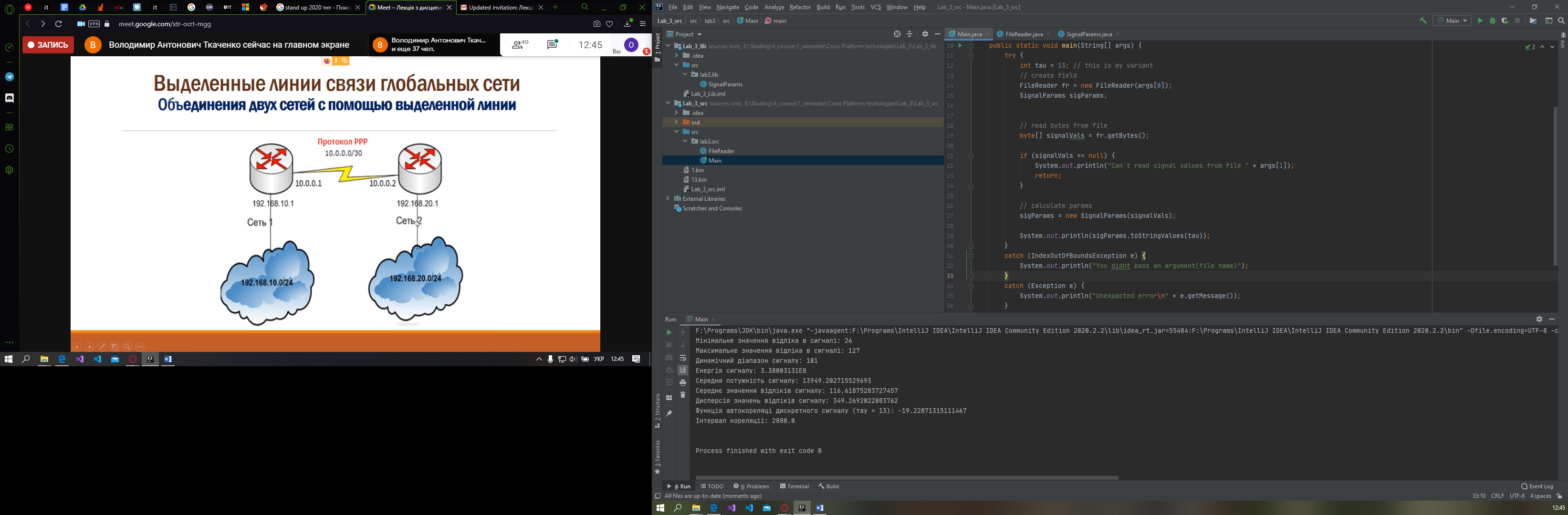


Рисунок 3 – Результат виконання програми

1. Посилання на репозитарій із проєктом у системі керування версіями.

<https://github.com/OlehTymoshenko/Cross-platform-technologies>

**Висновки:**

В результаті виконання лабораторної роботи я ознайомився з основними принципами написання простих Java-застосунків, навчився використовувати елементарні типи та керуючі оператори в Java. Також ознайомився з файловим вводом/виводом на Java. Також попрактикувався в роботі з IDE для розробки на Java – IntelliJ IDEA.