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))

Косован Д.Р.

(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 цели

1.1Ознакомится с принципами разработки простейших Java-приложений;

1.2Научиться использовать елементарные типы в программах на языке Java;

1.3Научиться использовать управляющие операторы в программах на языке Java.

1. Постановка задания работы

Реализовать Java-программу, которая вычисляет фзические характеристики цифрового сигнала, приведенные в разделе кратких теоретических ведомостей. Отсчет цифрового сигнала (целые числа, которые меняются в дыиапазоне 0-255) сохраняются в файле, путь и имя которого должны передаваться программе через командный ряд. Функции вычисления пересчитанных физических характеристик должны быть реализованы в виде библиотеки, которая подключается.Результаты расчета характеристик выводятся в консоль(функции автокореляции сторить не нужно).

Мой вариант 8, поэтому необходимо использовать файл 8.bin.

1. Исходный код программы

Клас Main.java

package lab3.src;  
  
import lab3.lib.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 try {  
 int tau = 8; // 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: Kosovan Danyyil

\*/

public class SignalParams {

private byte[] \_signal;

public SignalParams(byte[] signal) {

\_signal = signal;

}

public byte signalMinVal()

{

byte minValInSignal = \_signal[0];

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

if(minValInSignal > \_signal[i])

minValInSignal = \_signal[i];

}

return minValInSignal;

}

public byte signalMaxVal()

{

byte maxValInSignal = \_signal[0];

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

if(maxValInSignal < \_signal[i])

maxValInSignal = \_signal[i];

}

return maxValInSignal;

}

public byte signalDynamicRange() {

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

}

public double signalEnergy() {

double res = 0;

for (byte i:

\_signal) {

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

}

return res;

}

public double signalAvgPower() {

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

}

public double signalAvgValue() {

double res = 0;

for (byte i:

\_signal) {

res += i;

}

return res/\_signal.length;

}

public double signalDispersion() {

double signalAvgVal = signalAvgValue();

double res = 0;

for (byte i:

\_signal) {

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

}

return res/\_signal.length;

}

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;

}

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));

}

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.

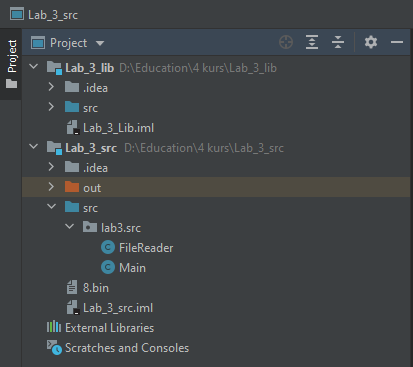


Рисунок 1 – Структура проекту

Настройка для выполнения программы(установка аргументу для программы) представлено на рис. 2.

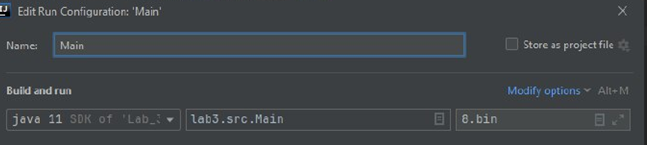


Рисунок 2 – Установка исходных данны харгументов для приложения

Результат выполнения программы представлен на рисунке 3.

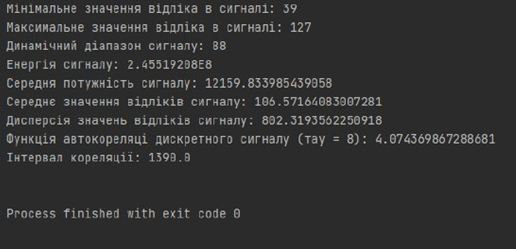


Рисунок 3 – Результат выполнения программы

1. Ссылка на репозиторий с проектом в системе управления.

<https://github.com/FattyWizard/Cross-Platform-techologies>

**Выводы:**

В результате выполнения лабораторной работы я ознакомился с основными принципами написания простых Java-приложений, научился использовать элементарные типы и управляющие операторы в Java. Также ознакомился с файловым вводом/выводом на Java. Также попрактиковался в работе с IDE для разработки на Java – IntelliJ IDEA.