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))

Поліщук А.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

Тема роботи:

Розроблення найпростішого Java-застосунку для обробки аудіо-даних

Мета роботи:

1. Ознайомитися з принципами розробки найпростіших Java-застосунків.
2. Навчиться використовувати елементарні типи в програмах на мові Java.
3. Навчиться використовувати керуючі оператори в програмах на мові Java.

У ході лабораторної роботи мені треба було використовувати **11.bin** файл

1. Вихідний код програми

Main.java

```
package lab3.src;
import lab3.lib.*;

public class Main {
    public static void main(String[] args)
    {
        try
        {
            int tau = 11; //Variant number
            FileReader fr = new FileReader(args[0]);
            SignalParameters sigParams;
            byte[] signalVals = fr.getBytes();
            if (signalVals == null)
            {
                System.out.println("Can not read values from file " + args[1]);
                return;
            }
            sigParams = new SignalParameters(signalVals);
            System.out.println(sigParams.valuesString(tau));
        }
        catch (IndexOutOfBoundsException e)
        {
            System.out.println("User did not pass a file name");
        }
        catch (Exception e)
        {
            System.out.println("Unexpected error\n" + e.getMessage());
        }
    }
}
```

FileReader.java

```java
package lab3.src;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class FileReader
{
    private String _path;

    public  FileReader(String path)
    {
        _path = path;
    }

    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;
    }
}
```

SignalParameters.java

```java
package lab3.lib;

//Polishuk Artem 545b

public class SignalParameters {
    private byte[] _signal;

    public SignalParameters(byte[] signal)
    {
        _signal = signal;
    }

    public byte minSignalValue()
    {
        byte minValInSignal = _signal[0];
        for(int i = 1; i < _signal.length; i++)
        {
            if(minValInSignal > _signal[i])
                minValInSignal = _signal[i];
        }
        return minValInSignal;
    }

    public byte maxSignalValue()
    {
        byte maxValInSignal = _signal[0];
        for(int i = 1; i < _signal.length; i++)
        {
            if(maxValInSignal < _signal[i])
                maxValInSignal = _signal[i];
        }
        return maxValInSignal;
    }

    public byte dynamicSignalRange()
    {
        return (byte) (maxSignalValue() - minSignalValue());
    }

    public double signalEnergy()
    {
        double res = 0;

        for (byte i:
             _signal) {
            res  = res + (Math.pow(i, 2));
        }

        return res;
    }

    public  double averageSignalPower()
    {
        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 autocorrelationSignalFunction(int tau)
    {
        double avgSignalVal = signalAvgValue();
        double res = 0;

        if(tau < 0)
        {
            res = autocorrelationSignalFunction(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 signalCorelationInterval()
    {
        double result = 0;
        double numerator = 0;

        for(int i = 0; i < _signal.length; i++)
        {
            numerator += autocorrelationSignalFunction(i);
        }

        result = numerator / autocorrelationSignalFunction(0);

        return Math.ceil(Math.abs(result));
    }

    public String valuesString(int tau)
    {
        String strResult = "Мінімальне значення відліка в сигналі: " +
minSignalValue() + "\n" +
                           "Максимальне значення відліка в сигналі: " +
maxSignalValue() + "\n" +
                           "Динамічний діапазон сигналу: " +  dynamicSignalRange() +
"\n" +
                           "Енергія сигналу: " +  signalEnergy() + "\n" +
```

```
                        "Середня потужність сигналу: " +  averageSignalPower() +
"\n" +
                        "Середнє значення відліків сигналу: " +  signalAvgValue()
+ "\n" +
                        "Дисперсія значень відліків сигналу: " +
signalDispersion() + "\n" +
                        "Функція автокореляці дискретного сигналу (тау = " + tau +
"): " +  autocorrelationSignalFunction(tau) + "\n" +
                        "Інтервал кореляції: " +  signalCorelationInterval() +
"\n";
        return strResult;
    }

}
```

2. Результат

Структура проекту в IDE представлена на рис. 1.

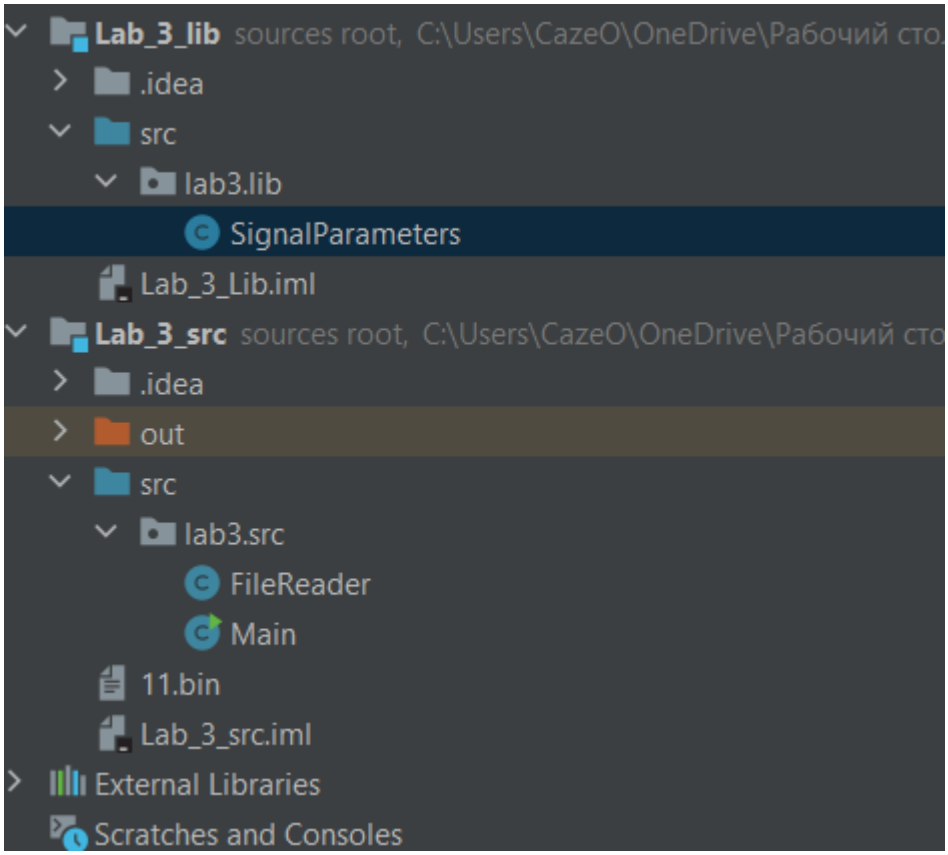

Рисунок 1 – Структура проекту

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

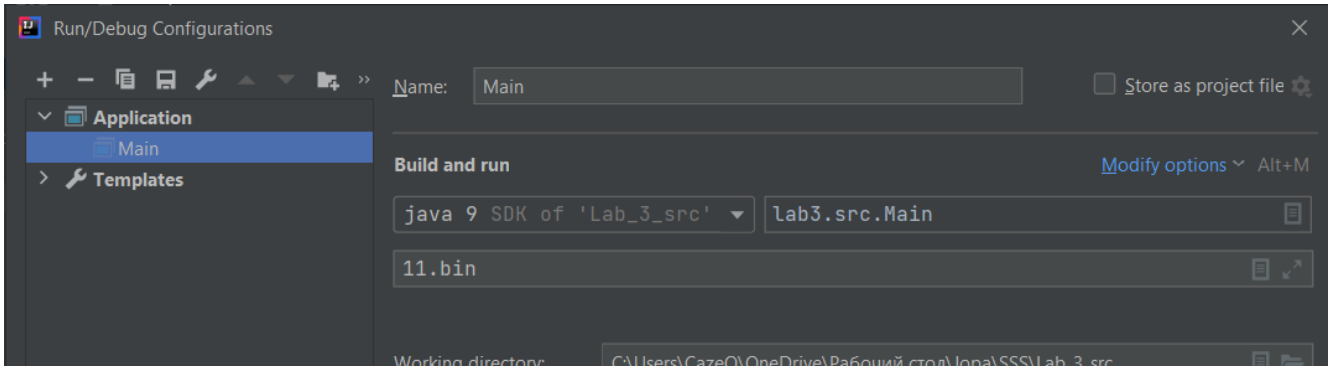

Рисунок 2 – Встановлення вхідних аргументів для додатку

Результат виконання програми представлено на рисунку 3.

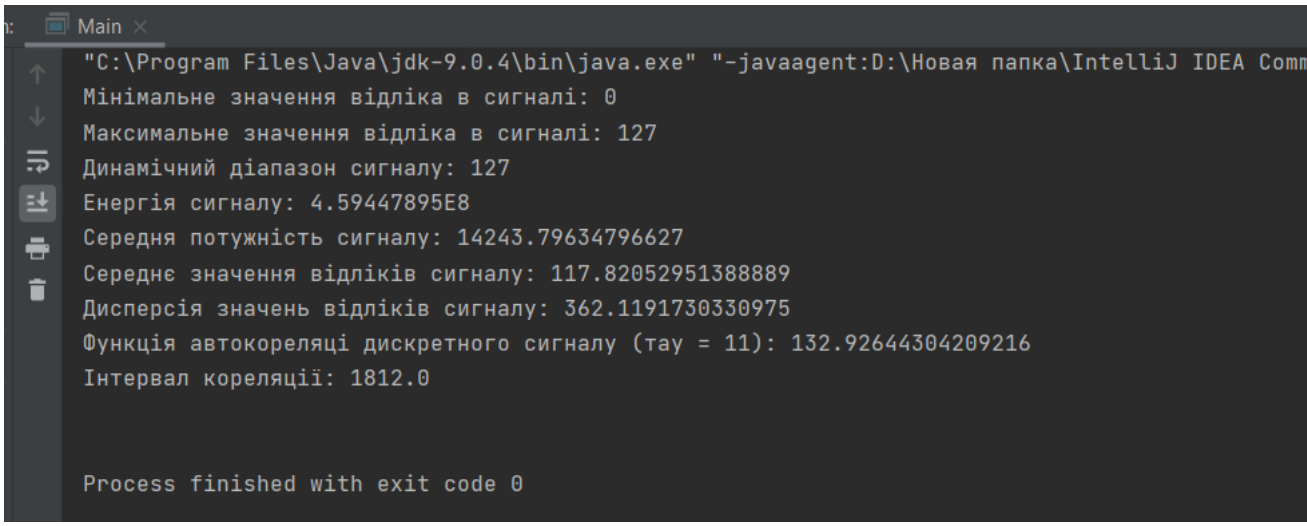

Рисунок 3 – Результат виконання програми

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

https://github.com/PolishukArtem/Cross/tree/main/LAB2

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.