КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

Кафедра мікроелектроніки

**ЛАБОРАТОРНА РОБОТА №17**

з дисципліни «Обчислювальна математика»

Варіант 3

Роботу виконав

Ст. групи ДП-01

Деркач Євген

Роботу перевірив

Татарчук Д. Д.

Київ-2021

**ЛАБОРАТОРНА РОБОТА №17**

**Тема:** Апроксимація функціональних залежностей методом найменших квадратів

**Мета роботи:** застосування методу найменших квадратів для побудови поліноміального середньоквадратичного наближення функції.

**Що зробити:** побудувати наближення функції f (x) лінійним поліномом за методом найменших квадратів, використовуючи набір із n псевдовипадкових «експериментальних» точок (xі, yі) (i = 1, 2, ... n), що розташовані на кривій f (x) та імітують серію вимірювань. Дослідити поведінку коефіцієнтів

МНК-полінома та средньоквадратичну похибку наближення σ в залежності від числа точок n. Провести аналогічні дослідження, додавши до «експериментальних» точок випадкові похибки. Додатково – побудувати МНК-наближення поліномами більш високих степенів та дослідити залежність σ від кількості точок n та степені МНК-полінома m.

**Код програми:**

package com.Jeka8833.LabsKPITwo.lab.labs;  
  
import com.Jeka8833.LabsKPITwo.Column;  
import com.Jeka8833.LabsKPITwo.ForceStopException;  
import com.Jeka8833.LabsKPITwo.Reader;  
import com.Jeka8833.LabsKPITwo.Writer;  
import com.Jeka8833.LabsKPITwo.lab.Lab;  
  
import java.util.Arrays;  
import java.util.Random;  
  
public class Laba17 implements Lab {  
  
 private static final Random *RANDOM* = new Random();  
  
 @Override  
 public String getName() {  
 return "Апроксимація функціональних залежностей методом найменших квадратів";  
 }  
  
 @Override  
 public void run() throws ForceStopException {  
 final int n = Reader.*readInt*("Введите n: ");  
 final double a = Reader.*readDouble*("Введите a: ");  
 final double b = Reader.*readDouble*("Введите b: ");  
  
 final double[] x = new double[n];  
 final double[] y = new double[n];  
  
 Writer.*add*(new Column("xi", Double.class), new Column("yi", Double.class));  
 for (int i = 0; i < n; i++) {  
 x[i] = a + (b - a) \* *RANDOM*.nextDouble();  
 y[i] = Laba3.*F*(x[i]);  
 Writer.*add*(x[i], y[i]);  
 }  
  
 final double xSum = Arrays.*stream*(x).sum();  
 final double ySum = Arrays.*stream*(y).sum();  
 final double xQuadSum = Arrays.*stream*(x).map(operand -> operand \* operand).sum();  
 double xySum = 0;  
 for (int i = 0; i < n; i++)  
 xySum += x[i] \* y[i];  
  
 final double[] res = Laba8.*calc*(2, new double[][]{  
 {n, xSum},  
 {xSum, xQuadSum}}, new double[]{ySum, xySum});  
 if (res == null) {  
 System.*out*.println("Ошибка вычисления матрицы");  
 return;  
 }  
  
 double S = 0;  
 for (int i = 0; i < n; i++) {  
 final double tmp = (res[0] + res[1] \* x[i]) - y[i];  
 S += tmp \* tmp;  
 }  
  
 Writer.*add*(new Column("c1", Double.class), new Column("c2", Double.class), new Column("sigma", Double.class));  
 Writer.*add*(res[0], res[1], Math.*sqrt*(S / n));  
 Writer.*saveResult*();  
 }  
}