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

Кафедра мікроелектроніки

**ЛАБОРАТОРНА РОБОТА №20**

з дисципліни «Обчислювальна математика»

Варіант 3

Роботу виконав

Ст. групи ДП-01

Деркач Євген

Роботу перевірив

Татарчук Д. Д.

Київ-2021

**ЛАБОРАТОРНА РОБОТА №20**

**Тема:** Звичайні диференціальні рівняння. Задача Коші

**Мета роботи:** вивчення методу Рунге-Кутти розв’язку задачі Коші для звичайного диференціального рівняння.

**Що зробити:** отримати розв’язок звичайного диференціального рівняння на певному інтервалі аналітично і за допомогою однієї з квадратурних формул Рунге-Кутти. Впевнитися у взаємоузгодженості отриманих результатів. Побудувати графіки y(t), y'(t) та фазову траекторію y'(y). Порівняти розбіжності між аналітичним і наближеними результатами при різних кроках інтегрування і визначити порядок точності квадратурної формули.

**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.function.BiFunction;  
  
public class Laba20 implements Lab {  
  
 private static final BiFunction<Double, Double, Double> *f* = (t, y) -> 1 / ((1 + Math.*exp*(-t)) \* y);  
 private static final int *t0* = -3;  
 private static final double *y0* = 0.6;  
 private static final int *tmax* = 7;  
  
  
 @Override  
 public String getName() {  
 return "No 20. Звичайні диференціальні рівняння. Задача Коші";  
 }  
  
 @Override  
 public void run() throws ForceStopException {  
 final int n = Reader.*readInt*("Введите n: ");  
 final double[] t = new double[n];  
 final double[] y = new double[n];  
 final double h = (Math.*max*(*tmax*, 0) - Math.*max*(*t0*, 0)) / (double) n;  
 y[0] = *y0*;  
  
 *calc*(*f*, t, y, h);  
 Writer.*add*(new Column("t", Double.class), new Column("y", Double.class), new Column("yreal", Double.class));  
 for (int i = 0; i < n; i++) {  
 Writer.*add*(t[i], y[i], Math.*sqrt*(2 \* Math.*log*(1 + Math.*exp*(t[i]))));  
 }  
 Writer.*saveResult*();  
 }  
  
  
 static void calc(final BiFunction<Double, Double, Double> func, final double[] t, final double[] y, final double h) {  
 for (int n = 0; n < t.length - 1; n++) {  
 final double dy1 = h \* func.apply(t[n], y[n]);  
 final double dy2 = h \* func.apply(t[n] + h / 2.0, y[n] + dy1 / 2.0);  
 final double dy3 = h \* func.apply(t[n] + h / 2.0, y[n] + dy2 / 2.0);  
 final double dy4 = h \* func.apply(t[n] + h, y[n] + dy3);  
 t[n + 1] = t[n] + h;  
 y[n + 1] = y[n] + (dy1 + 2.0 \* (dy2 + dy3) + dy4) / 6.0;  
 }  
 }  
}