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

Кафедра мікроелектроніки

**ЛАБОРАТОРНА РОБОТА №21**

з дисципліни «Обчислювальна математика»

Варіант 3

Роботу виконав

Ст. групи ДП-01

Деркач Євген

Роботу перевірив

Татарчук Д. Д.

Київ-2021

**ЛАБОРАТОРНА РОБОТА №21**

**Тема:** Системи звичайних диференціальних рівнянь

**Мета роботи:** вивчення алгоритмів розв’язку системи звичайних диференціальних рівнянь на прикладі моделювання руху супутника Землі або іншого небесного тіла.

**Що зробити:** розрахувати висоту r0 перицентру чи апоцентру орбіти, яку вам пропонується промоделювати, та швидкість v0 супутника в цій точці. Використовуючи ці величини як початкові умови, розрахувати рух супутника протягом одного оберту по орбіті і відобразіити результати розрахунків графічно. Додатково – дослідити, як похибка розрахунків залежить від кроку по часу. Перевірити, наскільки точно повна енергія супутника та його момент імпульсу зберігають незмінне значення при розрахунках.

**Код програми:**

package com.Jeka8833.LabsKPITwo.lab.labs;  
  
import com.Jeka8833.LabsKPITwo.Column;  
import com.Jeka8833.LabsKPITwo.ForceStopException;  
import com.Jeka8833.LabsKPITwo.Writer;  
import com.Jeka8833.LabsKPITwo.lab.Lab;  
  
import java.util.function.BiFunction;  
  
public class Laba21 implements Lab {  
  
 private static final BiFunction<Double, Double, Double> *dx* = (t, y) -> Math.*sin*(t) - y;  
 private static final BiFunction<Double, Double, Double> *dy* = (t, x) -> Math.*cos*(t) - x;  
  
 private static final int *t0* = 0;  
 private static final int *tMax* = 1;  
 private static final int *x0* = 0;  
 private static final int *y0* = 1;  
  
 @Override  
 public String getName() {  
 return null;  
 }  
  
 @Override  
 public void run() throws ForceStopException {  
 final int n = 1000;  
 final double[] t = new double[n];  
 final double[] dyx = new double[n];  
 final double[] dxy = new double[n];  
 final double h = (*tMax* - *t0*) / (double) n;  
 for (int i = *t0* \* n; i < *tMax* \* n; i++) {  
 t[i] = i / (double) n;  
 }  
  
 dyx[0] = *y0*;  
 dxy[0] = *x0*;  
  
 *calc*(*dx*, t, dyx, -h);  
 *calc*(*dy*, t, dxy, -h);  
  
 Writer.*add*(new Column("t", Double.class), new Column("x", Double.class), new Column("y", Double.class),  
 new Column("x-real", Double.class), new Column("y-real", Double.class), new Column("x-error", Double.class), new Column("y-error", Double.class));  
 for (int i = 0; i < n; i++) {  
 final double t1 = t[i];  
 final double x = -0.5 \* Math.*exp*(-t1) \* (Math.*exp*(2 \* t1) - 1);  
 final double y = 0.5 \* (Math.*exp*(-t1) + Math.*exp*(t1) + 2 \* Math.*sin*(t1));  
 Writer.*add*(t1, dxy[i], dyx[i],  
 x, y,  
 dxy[i] - x, dyx[i] - y);  
 }  
 Writer.*saveResult*();  
 throw new ForceStopException();  
 }  
  
 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);  
 y[n + 1] = y[n] + (dy1 + 2.0 \* (dy2 + dy3) + dy4) / 6.0;  
 }  
 }  
}