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

Кафедра мікроелектроніки

**ЛАБОРАТОРНА РОБОТА №18**

з дисципліни «Обчислювальна математика»

Варіант 3

Роботу виконав

Ст. групи ДП-01

Деркач Євген

Роботу перевірив

Татарчук Д. Д.

Київ-2021

**ЛАБОРАТОРНА РОБОТА №18**

**Тема:** Чисельне інтегрування. Формули прямокутників, трапецій, Сімпсона

**Мета роботи:** вивчення алгоритмів Ньютона-Котеса чисельного інтегрування функції однієї змінної: квадратурних формул прямокутників, трапецій, Сімпсона та дослідження поведінки їх похибок.

**Що зробити:** обчислити інтеграл аналітично і за допомогою складеної квадратурної формули при різних кількостях підінтервалів n. Впевнитися у взаємоузгодженості отриманих результатів. Порівняти розбіжності між аналітичним і наближеними результатами при різних n 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;  
  
public class Laba18 implements Lab {  
 @Override  
 public String getName() {  
 return null;  
 }  
  
 @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 real = *extractIntegral*(a, b);  
 Writer.*add*(new Column("n", Integer.class), new Column("real", Double.class),  
 new Column("value", Double.class), new Column("e", Double.class));  
 for (int i = 1; i <= n; i += 10) {  
 final double tmp = *calc*(a, b, i);  
 Writer.*add*(i, real, tmp, real - tmp);  
 }  
 Writer.*saveResult*();  
 }  
  
 public static double calc(final double a, final double b, final double n) {  
 final double step = ((b - a) / n);  
  
 double sum = 0;  
 for (int i = 1; i < n - 1; i++)  
 sum += Laba3.*F*(a + step \* i);  
 return ((b - a) / (2 \* n)) \* (Laba3.*F*(a) + 2 \* sum + Laba3.*F*(b));  
 }  
  
 public static double extractIntegral(final double down, final double up) {  
 return (2 \* Math.*pow*(down, 3)) / 3 - 3 \* down - Math.*cosh*(down) - (2 \* Math.*pow*(up, 3)) / 3 + 3 \* up + Math.*cosh*(up);  
 }  
}