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

Кафедра мікроелектроніки

**ЛАБОРАТОРНА РОБОТА №13**

з дисципліни «Обчислювальна математика»

Варіант 3

Роботу виконав

Ст. групи ДП-01

Деркач Євген

Роботу перевірив

Татарчук Д. Д.

Київ-2021

**ЛАБОРАТОРНА РОБОТА №13**

**Тема:** Оптимізація функцій однієї змінної методом золотого перетину

**Мета роботи:** вивчення алгоритму і налаштування програми для пошуку мінімуму функції одного аргументу (одновимірної оптимізації) методом золотого перетину.

**Що зробити:** знайти екстремуми функції f (x) методом золотого перетину. Впевнитись, що їх значення узгоджуються з результатами аналітичного дослідження функції f (x). Визначити порядок збіжності методу золотого перетину. Оцінити максимальну кількість десяткових знаків, які можна визначити цим методом в положенні точки екстремуму і в значенні функції в цій точці.

**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 Laba13 implements Lab {  
 @Override  
 public String getName() {  
 return "No 13. Оптимізація функцій однієї змінної методом Золотого перетину";  
 }  
  
 @Override  
 public void run() throws ForceStopException {  
 final double a = Reader.*readDouble*("Введите a: ");  
 final double b = Reader.*readDouble*("Введите b: ");  
 final double e = Reader.*readDouble*("Введите e: ");  
 final double res = *calc*(a, b, e, true);  
 Writer.*saveResult*();  
 System.*out*.println("Ответ x: " + res);  
 }  
  
 public static double calc(double a, double b, final double e, final boolean findMax) {  
 Writer.*add*(new Column("i", Integer.class), new Column("a", Double.class),  
 new Column("c", Double.class), new Column("d", Double.class), new Column("b", Double.class),  
 new Column("gamma", Double.class), new Column("fa", Double.class), new Column("fc", Double.class),  
 new Column("fd", Double.class), new Column("fb", Double.class));  
 double fa = findMax ? *F*(a) : -*F*(a);  
 double fb = findMax ? *F*(b) : -*F*(b);  
 final double r = 1 - (Math.*sqrt*(5) - 1) / 2; // Magic constant: 0.3819660112501051517954131656343618822796908201942371378645513772  
 double c = a + r \* (b - a);  
 double fc = findMax ? *F*(c) : -*F*(c);  
 double d = b - r \* (b - a);  
 double fd = findMax ? *F*(d) : -*F*(d);  
 int i = 0;  
 do {  
 Writer.*add*(i++, a, c, d, b, Math.*abs*(b - a), fa, fc, fd, fb);  
 if (fc > fd) {  
 a = c;  
 fa = fc;  
 c = d;  
 fc = fd;  
 d = b - r \* (b - a);  
 fd = findMax ? *F*(d) : -*F*(d);  
 } else {  
 b = d;  
 fb = fd;  
 d = c;  
 fd = fc;  
 c = a + r \* (b - a);  
 fc = findMax ? *F*(c) : -*F*(c);  
 }  
 } while (Math.*abs*(b - a) > e || (fa == fc && fc == fd) || (fc == fd && fd == fb));  
 return c;  
 }  
  
 public static double F(final double value) {  
 return Laba3.*F*(value);  
 }  
}