

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И
ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Кафедра «ВТ»

ЛАБОРАТОРНАЯ РАБОТА № _4_

ПО ДИСЦИПЛИНЕ «Вычислительная математика»

Выполнил(а): Чан Чунг Дык

Группа: Р3202

Вариант : 18

Санкт-Петербург

2017/2018

Цель работы: найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

Исходные данные:

1. Пользователь выбирает функцию, интеграл которой требуется вычислить (3-5 функций), из тех, которые предлагает программа.
2. Пределы интегрирования задаются пользователем.
3. Точность вычисления - 10^{-6} .
4. Начальное значение числа разбиения интервала интегрирования – $n=4$.

```
import java.util.Scanner;

public class Lab4 {

    public static final double eps = Math.pow(10,-6);

    public static final double res[] = { 13.833333,4.670774,14,0.499822};

    private final int n = 4;

    private int number;

    private int method;

    private void input(){

        Scanner sc = new Scanner(System.in);

        System.out.println("Please choose function, which you like to calculate integral");

        System.out.println("1 : integral from (1 to 2) of  $f(x) = x^2 + 3x + 7$ ";//13.8(3)

        System.out.println("2 : integral from (1 to 2) of  $f(x) = e^x$ ";//4.671

        System.out.println("3 : integral from (1 to 2) of  $f(x) = 4x+8$ ";//14

        System.out.println("4 : integral from (1 to 2) of  $f(x) = \cos(x)/2$ ";//0.9996

        number = sc.nextInt();

        System.out.println("Please choose method you want");

        System.out.println("1: Rectangle method");

        System.out.println("2: Trapezium method");

        System.out.println("3: Simpson method");
```

```

        method = sc.nextInt();

    }

    private void find1(){

        Rectangle duc = new Rectangle(number,n,1,2);

        duc.solve();

    }

    private void find2(){

        Trapezium duc = new Trapezium(number,n,1,2);

        duc.solve();

    }

    private void find3(){

        Simpson duc = new Simpson(number,n,1,2);

        duc.solve();

    }

    private void solve(){

        switch (method){

            case 1: find1(); break;

            case 2: find2(); break;

            case 3: find3(); break;

        }

    }

    public static void main(String[] args){

        Lab4 duc = new Lab4();

        duc.input();

        duc.solve();

    }

}

```

```
public class findValue {

    public static double solve(int number, double x){

        double res =0;

        switch (number){

            case 1: res = Math.pow(x,2) + 3*x + 7 ; break;

            case 2: res = Math.exp(x);break;

            case 3: res = 4*x + 8; break;

            case 4: res = Math.cos(x)/2 ; break;

        }

        return res;

    }

}
```

```
public class Rectangle {

    private int number;

    private int n;

    private double a;

    private double b;

    private double h ;

    Rectangle(int number,int n,double a,double b){

        this.number = number;

        this.n = n;

        this.a = a;

        this.b = b;

    }

    private double lValue(){

        double res = 0;

        for(int i = 0;i<n;i++){

            res += findValue.solve(number,a+i*h);

        }

    }

}
```

```

    }

    return res*this.h;
}

private double rValue(){
    double res = 0;

    for(int i = 1;i<=n;i++){
        res += findValue.solve(number,a+i*h);
    }

    return res*this.h;
}

private double mValue(){
    double res = 0;

    for(double i = 0;i<n;i++){
        res += findValue.solve(number,a+h*(i+0.5));
    }

    return res*this.h;
}

public void solve(){
    double res =0;

    while (true){
        h = (b-a)/n;

        res = mValue();

        if (Math.abs(res - Lab4.res[number-1]) <= Lab4.eps) {
            System.out.println(res + " " + n );

            return;
        }

        n*=2;
    }
}

```

```

public class Simpson {

    private int number;

    private int n;

    private double a;

    private double b;

    private double h ;

    Simpson(int number,int n,double a,double b){

        this.number = number;

        this.n = n;

        this.a = a;

        this.b = b;

    }

    public void solve(){

        while (true){

            double res = findValue.solve(number,a) + findValue.solve(number,b);

            h = (b-a)/n;

            for(int i=1;i<n;i++){

                if (i%2 == 1) res += 4*findValue.solve(number,a+i*h);

                else res+= 2*findValue.solve(number,a+i*h);

            }

            res *= h/3;

            if (Math.abs(res - Lab4.res[number-1]) <= Lab4.eps) {

                System.out.println(res + " " + n);

                return;

            }

            n*=2;

        }

    }
}

```

```

public class Trapezium {

    private int number;

    private int n;

    private double a;

    private double b;

    private double h ;

    Trapezium(int number,int n,double a,double b){

        this.number = number;

        this.n = n;

        this.a = a;

        this.b = b;

    }

    public void solve(){

        while (true){

            double res =findValue.solve(number,a) + findValue.solve(number,b);

            h = (b-a)/n;

            for(int i=1;i<n;i++){

                res+= 2*findValue.solve(number,a+i*h);

            }

            res *= h/2;

            if (Math.abs(res - Lab4.res[number-1]) <= Lab4.eps){

                System.out.println(res + " " + n);

                return;

            }

            n*=2;

        }

    }
}

```