**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«Университет ИТМО»**

**Мегафакультет:** Компьютерных технологий и управления

**Направление:** 09.03.04 «Программная инженерия»

**Лабораторная работа 2**

По дисциплине:

«Вычислительная математика»

На тему:

«Численное интегрирование»

**Выполнила:**

студентка группы **P3214**

Маньшина Елена

Витальевна

**Преподаватель:**

Малышева Татьяна

Алексеевна

Санкт-Петербург

2020

Вариант 10.

· **Цель работы:** Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами

**Блок схемы, используемых методов. Их рабочие формулы.**

Метод прямоугольников

– левые прямоугольники

– правые прямоугольники

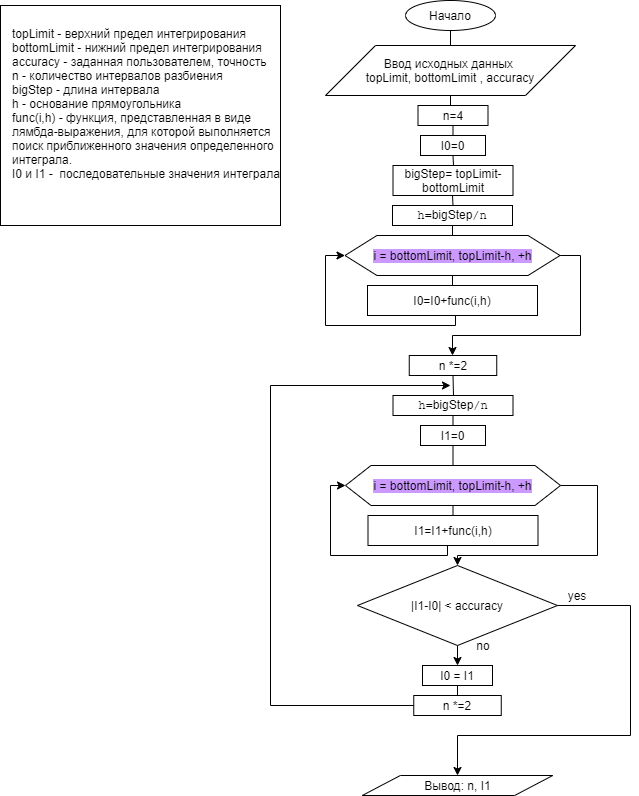
При

Метод средних

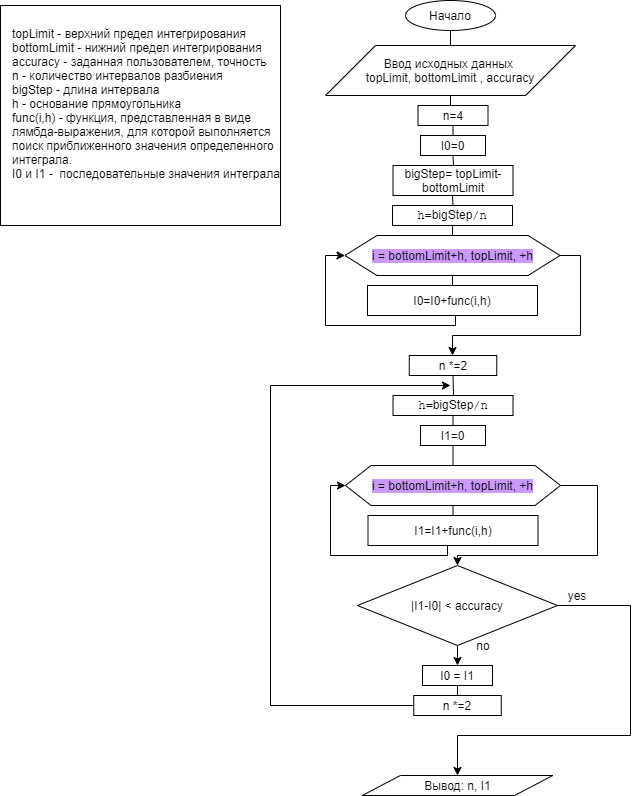
При

Метод Симпсона

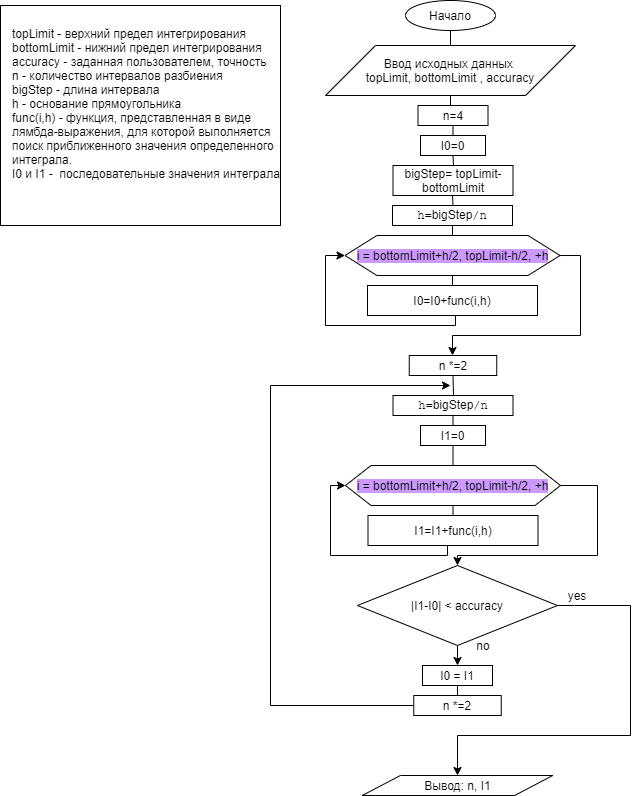
*Левые прямоугольники*

**

*Правые прямоугольники*

**

*Метод средних*

**

**Листинг программы.**

Варианты функций, передающихся в конструктор класса:

**1) 8+2x-x^2   
2) (2x)^(1/3)+x^(1/3)  
3) (x^2+x+1)^(1/3)**

**public class Function** {  
 **private double topLimit**;  
 **private double bottomLimit**;  
 **private double accuracy**;  
 **private** BiFunction<Double, Double, Double> **function**;  
 **private** String **funS**;  
  
 **public** Function(**double** topLimit, **double** bottomLimit, **double** accuracy, BiFunction<Double, Double, Double> fun, String funS) {  
 **this**.**topLimit** = topLimit;  
 **this**.**bottomLimit** = bottomLimit;  
 **this**.**accuracy** = accuracy;  
 **this**.**function** = fun;  
 **this**.**funS** = funS;  
 }  
  
 **void** leftRectangles() {  
 **int** n = 4;  
 **final double** bigStep = **topLimit** - **bottomLimit**;  
 **double** I0 = 0;  
 **double** h = bigStep / n;  
  
 **for** (**double** i = **bottomLimit**; i < **topLimit**; i += h) {  
 I0 = I0 + **function**.apply(h, i);  
 }  
 n \*= 2;  
 **while** (**true**) {  
 h = bigStep / n;  
 **double** I1 = 0;  
 **for** (**double** i = **bottomLimit**; i < **topLimit**; i += h) {  
 I1 = I1 + **function**.apply(h, i);  
 }  
  
 **if** (Math.*abs*(I1 - I0) < **accuracy**) {  
 I0 = I1;  
 **break**;  
 } **else** {  
 I0 = I1;  
 n \*= 2;  
 }  
 }  
  
 System.***out***.printf(**"Функция %s, метод левых прямоугольников\n"** +  
 **"Верхний предел %f, нижний предел %f, точность %f \n"** +  
 **"Значение интеграла: %f \n"** +  
 **"Число разбиения интервала для получения требуемой точности %d\n"**,  
 **funS**, **topLimit**, **bottomLimit**, **accuracy**, I0, n);  
  
 }  
  
 **void** rightRectangles() {  
 **int** n = 4;  
 **final double** bigStep = **topLimit** - **bottomLimit**;  
 **double** I0 = 0;  
 **double** h = bigStep / n;  
  
 **for** (**double** i = **bottomLimit**+h; i <= **topLimit**; i += h) {  
 I0 = I0 + **function**.apply(h, i);  
 }  
 n \*= 2;  
 **while** (**true**) {  
 h = bigStep / n;  
 **double** I1 = 0;  
 **for** (**double** i = **bottomLimit** + h; i <= **topLimit**; i += h) {  
 I1 = I1 + **function**.apply(h, i);  
 }  
  
 **if** (Math.*abs*(I1 - I0) < **accuracy**) {  
 I0 = I1;  
 **break**;  
 } **else** {  
 I0 = I1;  
 n \*= 2;  
 }  
 }  
  
 System.***out***.printf(**"Функция %s, метод правых прямоугольников\n"** +  
 **"Значение интеграла: %f \n"** +  
 **"Число разбиения интервала для получения требуемой точности %d\n"**,  
 **funS**, I0, n);  
 }  
  
 **void** middleRectangles() {  
 **int** n = 4;  
 **final double** bigStep = **topLimit** - **bottomLimit**;  
 **double** I0 = 0;  
 **double** h = bigStep / n;  
  
 **for** (**double** i = **bottomLimit** + h / 2; i <= **topLimit** - h / 2; i += h) {  
 I0 = I0 + **function**.apply(h, i);  
 }  
 n \*= 2;  
 **while** (**true**) {  
 h = bigStep / n;  
 **double** I1 = 0;  
 **for** (**double** i = **bottomLimit** + h / 2; i <= **topLimit** - h / 2; i += h) {  
 I1 = I1 + **function**.apply(h, i);  
 }  
  
 **if** (Math.*abs*(I1 - I0) < **accuracy**) {  
 I0 = I1;  
 **break**;  
 } **else** {  
 I0 = I1;  
 n \*= 2;  
 }  
 }  
  
 System.***out***.printf(**"Функция %s, метод средних прямоугольников\n"** +  
 **"Значение интеграла: %f \n"** +  
 **"Число разбиения интервала для получения требуемой точности %d\n"**,  
 **funS**, I0, n);  
 }  
  
 **void** calculateAllMethods() {  
 leftRectangles();  
 rightRectangles();  
 middleRectangles();  
 }  
}

//обработка ввода данных

**public class DataReceiver** {  
  
 BufferedReader **in** = **new** BufferedReader(**new** InputStreamReader(System.***in***));  
 MethodOfRectangles **methodOfRectangles** = **new** MethodOfRectangles();  
  
 **void** selectFunction() {  
  
 **boolean** success = **false**;  
  
 **while** (!success) {  
 System.***out***.println(**"Выберете одну из функций, для которой будет вычислен интеграл методом Прямоугольников\nИ введите ее номер.\n"** + **"1) 8+2x-x^2 \n"** + **"2) (2x)^(1/3)+x^(1/3) \n"** + **"3) (x^2+x+1)^(1/3)"**);  
 **try** {  
 String integrals = **in**.readLine();  
 success = **true**;  
 **double**[] parameters;  
 **switch** (integrals) {  
 **case "1"**:  
  
 parameters = getLimitOfIntegrationAndAccuracy();  
 **methodOfRectangles**.function1(parameters[0], parameters[1], parameters[2]);  
 **break**;  
 **case "2"**:  
  
 parameters = getLimitOfIntegrationAndAccuracy();  
 **methodOfRectangles**.function2(parameters[0], parameters[1], parameters[2]);  
 **break**;  
 **case "3"**:  
  
 parameters = getLimitOfIntegrationAndAccuracy();  
 **methodOfRectangles**.function3(parameters[0], parameters[1], parameters[2]);  
 **break**;  
 **default**:  
 success = **false**;  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 System.***out***.println(**"Неверный ввод!"**);  
 success = **false**;  
 }  
 }  
  
 }  
  
  
 **double**[] getLimitOfIntegrationAndAccuracy() {  
  
 **double**[] parameters = **new double**[3];  
 **boolean** exit = **false**;  
  
 **while** (!exit) {  
 **try** {  
  
 System.***out***.println(**"Введите пределы интегрирования. Верхний: "**);  
 parameters[0] = Double.*parseDouble*(**in**.readLine());  
 System.***out***.println(**"Нижний:"**);  
 parameters[1] = Double.*parseDouble*(**in**.readLine());  
 System.***out***.println(**"Введите требуемую точность:"**);  
 parameters[2] = Double.*parseDouble*(**in**.readLine());  
 exit = **true**;  
 **if** (parameters[0] < parameters[1]) {  
 System.***out***.println(**"Верхний предел меньше нижнего. Пожалуйста, повторите ввод"**);  
 exit = **false**;  
 }  
  
 } **catch** (Exception e) {  
 System.***out***.println(**"При вводе данных обнаружена ошибка. Попробуйте еще раз."**);  
 exit = **false**;  
 }  
 }  
  
 **return** parameters;  
 }  
  
}

**public class MethodOfRectangles** {  
  
 **void** function1(**double** topLimit, **double** bottomLimit, **double** accuracy){  
 Function function = **new** Function(topLimit, bottomLimit, accuracy, (Double h, Double i) -> h\*(8 + 2\*i - i\*i), **"8+2x-x^2"**);  
 function.calculateAllMethods();  
 }  
 **void** function2(**double** topLimit, **double** bottomLimit, **double** accuracy){  
 Function function = **new** Function(topLimit, bottomLimit, accuracy, (Double h, Double i) -> h\*(Math.*cbrt*(2\*i)+Math.*cbrt*(i)), **"(2x)^(1/3)+x^(1/3)"**);  
 function.calculateAllMethods();  
 }  
 **void** function3(**double** topLimit, **double** bottomLimit, **double** accuracy){  
 Function function = **new** Function(topLimit, bottomLimit, accuracy, (Double h, Double i) -> h\*(Math.*cbrt*(i\*i+i+1)), **"(x^2+x+1)^(1/3)"**);  
 function.calculateAllMethods();  
 }  
  
}

**Результаты выполнения программы.**

* **Для функции 1 ()**

Функция 8+2x-x^2, метод левых прямоугольников

Верхний предел 4,000000, нижний предел -2,000000, точность 0,010000

Значение интеграла: 35,997803

Число разбиения интервала для получения требуемой точности 128

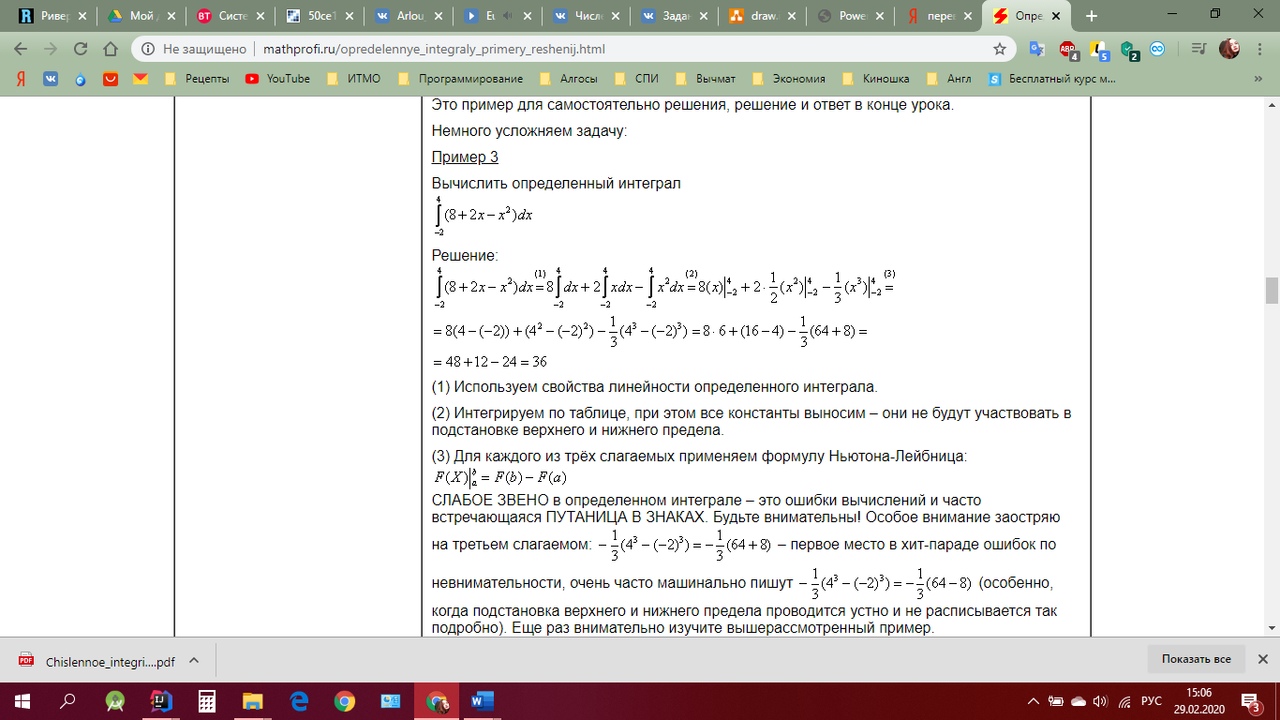
Функция 8+2x-x^2, метод правых прямоугольников

Значение интеграла: 35,997803

Число разбиения интервала для получения требуемой точности 128

Функция 8+2x-x^2, метод средних прямоугольников

Значение интеграла: 36,001099

Число разбиения интервала для получения требуемой точности 128

* **Для функции 2 (** **)**

Функция (2x)^(1/3)+x^(1/3), метод левых прямоугольников

Верхний предел 9,000000, нижний предел 1,000000, точность 0,000010

Значение интеграла: 30,035620

Число разбиения интервала для получения требуемой точности 1048576

Функция (2x)^(1/3)+x^(1/3), метод правых прямоугольников

Значение интеграла: 30,035639

Число разбиения интервала для получения требуемой точности 1048576

Функция (2x)^(1/3)+x^(1/3), метод средних прямоугольников

Значение интеграла: 30,035631

Число разбиения интервала для получения требуемой точности 1024

* **Для функции 3**  )

Функция (x^2+x+1)^(1/3), метод левых прямоугольников

Верхний предел 8,000000, нижний предел 0,000000, точность 0,000010

Значение интеграла: 21,550520

Число разбиения интервала для получения требуемой точности 2097152

Функция (x^2+x+1)^(1/3), метод правых прямоугольников

Значение интеграла: 21,550532

Число разбиения интервала для получения требуемой точности 2097152

Функция (x^2+x+1)^(1/3), метод средних прямоугольников

Значение интеграла: 21,550527

Число разбиения интервала для получения требуемой точности 128

**Вычисление заданного интеграла.**

**Точно**

**Вычисление интеграла методом Симпсона при n=10**

**Воспользуемся формулой**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 2 | 2.2 | 2.4 | 2.6 | 2.8 | 3 | 3.2 | 3.4 | 3.6 | 3.8 | 4 |
|  | 0 | 1,528 | 3,344 | 5,496 | 8,032 | 11 | 14,448 | 18,424 | 22,976 | 28,152 | 34 |

**Абсолютная погрешность вычислений:**

**Вывод:** В ходе лабораторной работы я научилась пользоваться методом прямоугольников и методом Симпсона для вычисления приближенного значения определенного интеграла.