**ФПИиКТ**

**Рабочий протокол и отчет по**

**домашней работе №3**

**Вариант №6**

Ибадуллаев Алибаба Эльбрус оглы

Группа: P3215

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург

2022г.

# 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 задается пользователем.
4. Начальное значение числа разбиения интервала интегрирования: n=4.
5. Ввод исходных данных осуществляется с клавиатуры.

# Программная реализация задачи:

1. Реализовать в программе методы по выбору пользователя, исходя из варианта:

* Метод прямоугольников (3 модификации: левые, правые, средние)
* Метод трапеций
* Метод Симпсона

1. Методы должны быть оформлены в виде отдельной(ого) функции/класса.
2. Вычисление значений функции оформить в виде отдельной(ого) функции/класса.
3. Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.
4. Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.

|  |  |  |
| --- | --- | --- |
| 6 | Симпсона  Трапеций |  |

**Вычислительная реализация задачи:**

1. Вычислить интеграл, приведенный в таблице 1 (столбец 3), точно.
2. Вычислить интеграл по формуле Ньютона – Котеса при .
3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при .

Метод средних прямоугольников.

Метод трапеций

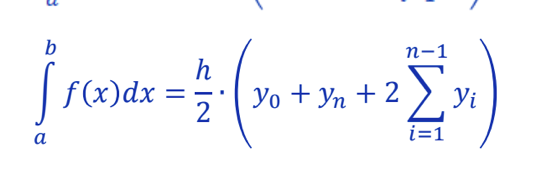
Метод Симпсона

1. Сравнить результаты с точным значением интеграла.
2. Определить относительную погрешность вычислений.
3. В отчете ***отразить последовательные вычисления***.

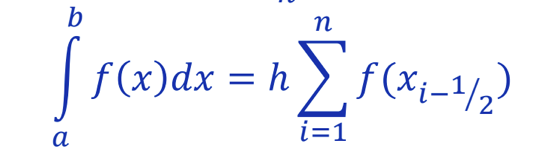
[**https://www.desmos.com/calculator/jd60r3facm**](https://www.desmos.com/calculator/jd60r3facm)

# Рабочие формулы методов

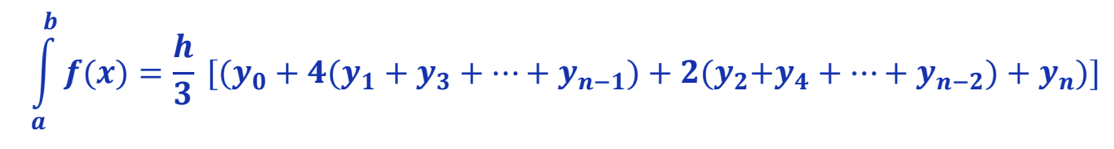
Метод трапеций:



Метод прямоугольника:



Метод Симпсона:



# Листинг программы

Метод Симпсона:

private static void simpsonMethod(double a, double b, double epsilon, int n, int iterations, double h, int afterComma, double result, double resultPrev, double resultCurr){  
 while (Math.*abs*(resultCurr - resultPrev)/(Math.*pow*(2,*K*) - 1) > epsilon){  
 result = h/3\*(*func*.apply(a)+*func*.apply(b));  
  
 for(int i = 1; i<n; i++){  
 if (i%2==0) {  
 result = result + h/3 \* 2\* *func*.apply(a+(i)\*h);  
 }  
 else {  
 result = result + h/3 \* 4 \* *func*.apply(a+i\*h);  
 }  
 }  
 resultPrev = resultCurr;  
 resultCurr = result;  
 n \*=2;  
 h = (b-a)/n;  
  
 if (iterations>=1) *printStep*(iterations, resultCurr, resultPrev, afterComma, epsilon, n);  
 iterations++;  
 }  
  
 System.*out*.println("<<RESULT>>");  
 *printStep*(iterations-1, resultCurr, resultPrev, afterComma, epsilon,n );  
}

Метод Трапеций:

private static void trapeziumMethod(double a, double b, double epsilon, int n, int iterations, double h, int afterComma, double result, double resultPrev, double resultCurr){  
 while (Math.*abs*(resultCurr - resultPrev)/(Math.*pow*(2,*K*) - 1) > epsilon){  
 result = h/2\*(*func*.apply(a)+*func*.apply(b));  
  
 for(int i = 1; i<n; i++){  
 result = result + h \**func*.apply(a+i\*h);  
 }  
 resultPrev = resultCurr;  
 resultCurr = result;  
 n \*=2;  
 h = (b-a)/n;  
  
 if (iterations>=1) *printStep*(iterations, resultCurr, resultPrev, afterComma, epsilon, n);  
 iterations++;  
 }  
  
 System.*out*.println("<<RESULT>>");  
 *printStep*(iterations-1, resultCurr, resultPrev, afterComma, epsilon,n );  
}

# Результаты выполнения программы

Метод Симпсона:

<<RESULT>>

Iteration 1

I = 103,333

|I1-I0| = 0,000 <= 0,001: true

n = 4

Метод трапеций:

<<RESULT>>

Iteration 6

I = 103,334

|I6-I5| = 0,001 <= 0,001 (true)

n = 128

Вывод:

**я научился пользоваться методом Симпсона и методом трапеций, также вспомнил как брать интегралы.**