

Высшая школа экономики  
Факультет компьютерных наук

## **Пояснительная записка**

Программа, вычисляющая определённый интеграл

Выполнил: Шакуров Семен Сергеевич, БПИ194 / 2 подгруппа.  
29 вариант.

## Текст задания

29. Вычислить интеграл:

$$\int_a^b f(x) dx,$$

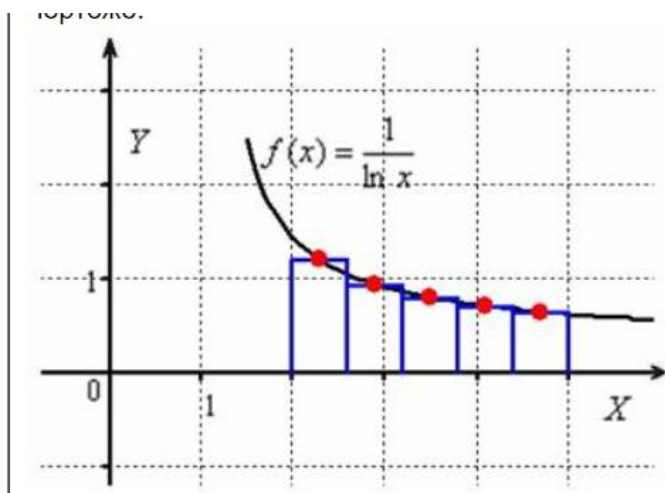
используя метод прямоугольников. Входные данные: вещественные числа  $a$  и  $b$ , функция  $f(x)$  задается с использованием описания в программе в виде отдельной функции. При суммировании использовать принцип дихотомии. Протестировать на различных функциях.

## Применяемые расчетные методы

Наиболее подходящая модель построения приложения – итеративный параллелизм.

В данном случае каждый поток будет вычислять площади определенных прямоугольников, при этом площадь одного и того же прямоугольника два разных потока вычислять не будут. Реализовано это с помощью массива, в котором за каждый элемент отвечает отдельный поток. Это исключает ситуации неуправляемого изменения одних и тех же общих данных несколькими потоками.

Вычисляется интеграл методом центральных прямоугольников. Промежуток  $[a, b]$  делится на много промежутков, в зависимости от  $h$ , и на каждом промежутке строится прямоугольник, например:



Тогда определенный интеграл будет равен сумме площадей этих прямоугольников.

## Область допустимых значений

Вначале задается левая граница интегрирования, при этом определенная функция (я определил функцию  $y = \sqrt{1 + x^3}$ ) должна существовать в этой точке. Затем задается правая граница интегрирования, при этом функция должна существовать в этой точке и значение должно быть больше левой границы.

## Тесты

C:\WINDOWS\system32\cmd.exe

```
f(x) = sqrt(1 + x^3)
Чтобы вычислить определенный интеграл функции f(x)
с шагом интегрирования 0.001 на промежутке от a до b, введите
a (>= -1): -3
Incorrect input. Try again: 0
b (>= -1 && > a): -1
Incorrect input. Try again: 0
Incorrect input. Try again: 10
Integral result: 127.858
Для продолжения нажмите любую клавишу . . .
```

C:\WINDOWS\system32\cmd.exe

```
f(x) = sqrt(1 + x^3)
Чтобы вычислить определенный интеграл функции f(x)
с шагом интегрирования 0.001 на промежутке от a до b, введите
a (>= -1): 10,5
b (>= -1 && > a): 100
Integral result: 39857.3
Для продолжения нажмите любую клавишу . . .
```

Для  $f(x) = 1 / \log(x)$

```
Чтобы вычислить определенный интеграл функции f(x)
с шагом интегрирования 0.001 на промежутке от a до b, введите
a (>= -1): 2
b (>= -1 && > a): 100
Integral result: 29.081
Для продолжения нажмите любую клавишу . . .
```

Для  $f(x) = \sin(5 * x) * \sin(7 * x)$

```
Чтобы вычислить определенный интеграл функции f(x)
с шагом интегрирования 0.001 на промежутке от a до b, введите
a (>= -1): 50
b (>= -1 && > a): 100
Integral result: -0.0862137
Для продолжения нажмите любую клавишу . . .
```

#### Список используемых источников

[http://mathprofi.ru/metod\\_prjamougolnikov.html](http://mathprofi.ru/metod_prjamougolnikov.html)

<https://nuancesprog.ru/p/5452>

## Приложение

### Текст программы

```
#include <iostream>
```

```
#include <math.h>
```

```
#include <thread>
```

```
// Вывод информации о функции
```

```
void printFunctionInfo() {
```

```
    std::cout << "f(x) = sqrt(1 + x^3)" << std::endl;
```

```
    std::cout << "Чтобы вычислить определенный интеграл функции f(x)" << std::endl;
```

```
    std::cout << "с шагом интегрирования 0.001 на промежутке от a до b, введите" << std::endl;
```

```
}
```

```
//сама функция
```

```
double f(double x) {
```

```
    return sqrt(1 + pow(x, 3));
```

```
}
```

```
// Получение левой границы интеграла.
```

```
double getA(const char *msg) {
```

```
    char answer[256];
```

```
    double x;
```

```
    printf("%s", msg);
```

```
    fgets(answer, sizeof(answer), stdin);
```

```
    while (sscanf_s(answer, "%lf", &x) != 1 || x < -1) {
```

```

        printf("Incorrect input. Try again: ");
        fgets(answer, sizeof(answer), stdin);
    }

    return x;
}

//Получение правой границы интеграла
double getB(const char* msg, double a) {
    char answer[256];
    double x;

    printf("%s", msg);
    fgets(answer, sizeof(answer), stdin);

    while (sscanf_s(answer, "%lf", &x) != 1 || x < -1 || a >= x) {
        printf("Incorrect input. Try again: ");
        fgets(answer, sizeof(answer), stdin);
    }

    return x;
}

double* arr; // Каждый поток отвечает за отдельный элемент в этом массиве

/// <summary>
/// Вычисление площадей прямоугольников отдельными потоками.
/// </summary>
/// <param name="number">Номер потока</param>

```

```

/// <param name="n">Количество разбиений на прямоугольники</param>
/// <param name="numberOfThreads">Количество потоков</param>
/// <param name="a">Левая граница интеграла</param>
/// <param name="h">Длина разделения на прямоугольники</param>
void square(int number, double n, int numberOfThreads, double a, double h) {
    for (int i = number + 1; i <= n; i += numberOfThreads) {
        double value = h * f(a + h * (i - 0.5));
        arr[number] += value;
    }
}

```

//Вычисление интеграла.

```

double integral(double a, double b, double h) {
    double n = (b - a) / h;
    int numberOfThreads = std::thread::hardware_concurrency();
    std::thread* threads = new std::thread[numberOfThreads];
    arr = new double[numberOfThreads];
    double sum = 0.0;

    for (int i = 0; i < numberOfThreads; i++) arr[i] = 0.0;

    for (int i = 0; i < numberOfThreads; i++) {
        threads[i] = std::thread(square, i, n, numberOfThreads, a, h);
        threads[i].join();
    }

    for (int i = 0; i < numberOfThreads; i++) {
        sum += arr[i];
        arr[i] = 0.0;
    }
}

```

```
    }

    return sum;
}

int main() {
    setlocale(LC_ALL, "Russian");
    printFunctionInfo();
    double a = getA("a (>= -1): ");
    double b = getB("b (>= -1 && > a): ", a);
    double h = 0.001;

    double res = integral(a, b, h);
    std::cout << "Integral result: " << res << std::endl;
    system("pause");
    return 0;
}
```