

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

Отчет по домашней работе №3

Исполнитель:
студентка группы **БПИ196**
Шилова Ксения Алексеевна

ВАРИАНТ 29

Формулировка задачи:

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.

17 ноября 2020 г.

Описание модели вычислений.

В программе была использована модель «Взаимодействующие равные», то есть отрезок интегрирования был разбит на 8 частей (число задается константой) – подобно методу дихотомии – и каждую часть считал отдельный поток. Все потоки меняли одну и ту же глобальную переменную `sum`, которая аппроксимирует интеграл.

СОСТАВЛЯЮЩИЕ ПРОГРАММЫ

- 1) Метод **`int readInterval()`**: считывание с консоли интервала интегрирования – вещественные числа `a`, `b`. Проверка их корректности.
- 2) Метод **`void displayFunctions()`**: отображений доступных для выбора подынтегральных функций. Их задано 6.
- 3) Метод **`void calculateIntegral(double (*func)(double), double lower, double higher, int numbThread)`** : процесс вычисления одним потоком части интеграла. `Func` – функция под интегралом, `lower` и `higher` – нижняя и верхняя границы интегрирования, `numbThread` – номер потока (для вывода информации).
- 4) **`double a; double b;`** - введенные вещественные числа)интервал)
- 5) **`double sum = 0;`** - переменная, аппроксимирующая интеграл
- 6) **`double delta;`** - переменная, зависящая от `a`, `b`, которая задает минимальный шаг (ширину прямоугольника)
- 7) **`const int threadNumber = 8;`** - количество потоков

Функции, которые предложены для взятия интеграла:

```

//ФУНКЦИИ
//константная функция
double f1(double x) {
    return 1;
}
//линейная
double f2(double x) {
    return x;
}
//степенная
double f3(double x) {
    return pow(x, 3);
}
//sin
double f4(double x) {
    return sin(x);
}
//показательная
double f5(double x) {
    return exp(x);
}
//корень
double f6(double x) {
    return sqrt(x);
}

```

Для того, чтобы каждый поток считал только свою часть, разбиваем весь интервал на более мелкие, и задаем их как параметры функции, которую считает отдельный поток:

```

std::thread* thr[threadNumber];
//задаем область для подсчета интеграла одним потоком
double oneInterval = (b - a) / threadNumber;
for (int i = 0; i < threadNumber; i++) {
    //считаем интегралы по частям
    thr[i] = new std::thread{ calculateIntegral, function, a + i * (oneInterval), a + (i + 1) * (oneInterval), i };
}
//удаляем
for (int i = 0; i < threadNumber; i++) {
    thr[i]->join();
    delete thr[i];
}
}

```

ТЕСТИРОВАНИЕ:

1. Некорректный интервал

```
Консоль отладки Microsoft Visual Studio
Type integration interval [a, b] (real numbers):
a = abcd
b =
Incorrect number.
C:\Users\Пользователь\source\repos\integral\Debug\integral.exe (процесс 20296) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка".
Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

```
Консоль отладки Microsoft Visual Studio
Type integration interval [a, b] (real numbers):
a = 1
b = abcd
Incorrect number.
C:\Users\Пользователь\source\repos\integral\Debug\integral.exe (процесс 36696) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

2. Интервал с положительными границами

```
C:\Users\Пользователь\source\repos\integral\Debug\integral.exe
Type integration interval [a, b] (real numbers):
a = 1
b = 5

Type number from 1 to 6 to choose the function:
f1(x) = 1
f2(x) = x
f3(x) = pow(x, 3)
f4(x) = sin(x)
f5(x) = exp(x)
f6(x) = sqrt(x)
Type "exit" to exit the program.

1
thread 1 0.500000
thread 0 1.000000
thread 2 1.500000
thread 7 2.000000
thread 4 2.500000
thread 5 3.000000
thread 6 3.500000
thread 3 4.000000
Integral from 1 to 5 f1(x) dx = 4
```

```
Integral from 1 to 5 f2(x) dx = 12
```

```
Integral from 1 to 5 f3(x) dx = 156
```

```
thread 7 0.250000
Integral from 1 to 5 f4(x) dx = 0.25664
```

```
thread 6 0.695000
Integral from 1 to 5 f5(x) dx = 145.695
```

```
thread 5 0.78689
Integral from 1 to 5 f6(x) dx = 6.78689
```

3. Интервал с отрицательными границами

```
Type integration interval [a, b] (real numbers):
a = -3
b = -1

Type number from 1 to 6 to choose the function:
f1(x) = 1
f2(x) = x
f3(x) = pow(x,3)
f4(x) = sin(x)
f5(x) = exp(x)
f6(x) = sqrt(x)
Type "exit" to exit the program.

1
thread 0 0.250000
thread 1 0.500000
thread 2 0.750000
thread 3 1.000000
thread 4 1.250000
thread 5 1.500000
thread 6 1.750000
thread 7 2.000000
Integral from -3 to -1 f1(x) dx = 2
```

```
Type number from 1 to 6 to choose the function:
f1(x) = 1
f2(x) = x
f3(x) = pow(x,3)
f4(x) = sin(x)
f5(x) = exp(x)
f6(x) = sqrt(x)
Type "exit" to exit the program.
```

```
6
Function f6 is not defined in negative numbers.
```

```
Type number from 1 to 6 to choose the function:
f1(x) = 1
f2(x) = x
f3(x) = pow(x,3)
f4(x) = sin(x)
f5(x) = exp(x)
f6(x) = sqrt(x)
Type "exit" to exit the program.
```

4. Интервал нулевой длины

```
Type integration interval [a, b] (real numbers):
a = 0
b = 0

Type number from 1 to 6 to choose the function:
f1(x) = 1
f2(x) = x
f3(x) = pow(x,3)
f4(x) = sin(x)
f5(x) = exp(x)
f6(x) = sqrt(x)
Type "exit" to exit the program.

1
Integral from 0 to 0 f1(x) dx = 0
```

5. «Большой интервал»

```
Type integration interval [a, b] (real numbers):
a = -1000
b = 1000
```

```
Type number from 1 to 6 to choose the function:  
f1(x) = 1  
f2(x) = x  
f3(x) = pow(x,3)  
f4(x) = sin(x)  
f5(x) = exp(x)  
f6(x) = sqrt(x)  
Type "exit" to exit the program.
```

```
3  
thread 0 -170898218750.000000  
thread 1 -234374625000.000000  
thread 2 -249022968750.000000  
thread 3 -249999500000.000000  
thread 4 -249022968750.000000  
thread 5 -234374625000.000000  
thread 6 -170898218750.000000  
thread 7 0.000000  
Integral from -1000 to 1000 f3(x) dx = 0
```

```
Type number from 1 to 6 to choose the function:  
f1(x) = 1
```

Список литературы

1. https://l.wzm.me/_coder/custom/parallel.programming/001.htm
2. <https://pro-prof.com/forums/topic/parallel-programming-paradigms>