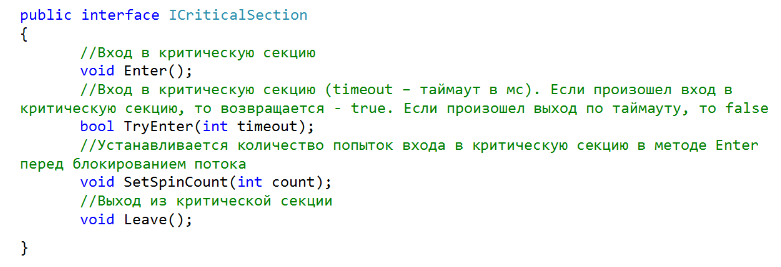
# Отчет по Лабораторной работе №3

Постановка задачи

Знакомство с возможностями стандартной библиотеки языка c# для работы с потоками. Изучение классов Mutex, Interlocked, AutoResetEvent. Реализация критической секции с использованием данных классов.

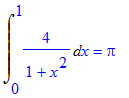
Задание: Написать класс (на языке c#), реализующий следующий интерфейс:



Данный класс должен реализовать критическую секцию с использованием System.Threading.Interlocked.

# Описание алгоритма

Мы будем считать число Пи путем численного интегрирования



 Возьмем самый простой из методов численного инетгрирования — метод прямоугольников. Объявим некоторое количество (stepCount) прямоугольников, на которые мы разобьем нашу площадь под интегралом, посчитаем основание:

step = 1/stepCount;

Вычислим всю площадь, посчитав значение функции в каждом прямоугольнике и домножив на основание.

for (unsigned long i=0; i< stepCount; i++)

{

x = ( i + 0.5 ) \* step;

sum = sum + 4.0 / (1.+ x\*x);

}

# Таблица с данными о времени выполнения и полученном значении числа π

ThreadsCount = 8



Вывод

При увеличении количества спинов увеличивается время работы программы. Увеличение timeout не значительно влияет на время выполнения. Выполнение TryEnter занимает больше времени, чем Enter, так как затрачивает его на ожидание.

# Листинг

<https://github.com/lezhnina-anna/PP/tree/master/lab3>