

Лабораторная работа №24

Использование потоков

1 Цель работы

- 1.1 Научиться разрабатывать многопоточные приложения на C#;
- 1.2 Научиться создавать и применять фоновые и основные потоки и выполнять обмен данных между ними в программах на C#.

2 Литература

- 2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВПетербург, 2019. – 512 с. – URL: <https://ibooks.ru/bookshelf/366634/reading>. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – гл.15.

2.2

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создание и приостановка потоков

Написать программу, которая создает два потока. Один поток выводит на экран числа от 0 до 9, а другой — буквы от 'A' до 'J'. Потоки должны работать параллельно. После вывода символа, должна производиться задержка на 200 миллисекунд (Thread.Sleep(задержка)).

5.2 Синхронизация потоков при помощи lock

Написать программу, где два потока пытаются одновременно изменить одну общую переменную, например, увеличить значение целочисленной переменной. Использовать блокировку (lock), чтобы избежать состояния гонки.

5.3 Использование мьютекса

Переписать программу из пункта 5.2 с использованием мьютекса вместо lock.

5.4 Обмен данными между потоками

Создать консольное приложение, в котором происходит обмен данными между потоками Main() и MyThread() через статическое поле (глобальную переменную) commonVar.

Второй поток выполняется до того момента, пока в первом переменной str не будет присвоено значение «х».

5.5 Настройка приоритета потоков

В методе Main запустить на выполнение 4 потока, каждый из которых будет вызывать метод WriteString и с его помощью выводить свой номер в окно консоли. При создании потоков в Main установить для них различные приоритеты.

Для большей наглядности в начале метода WriteString реализовать вывод на отдельной строке сообщения, что «Поток N запущен», в конце метода – «Поток N завершен» (вместо N отображать значение параметра N).

5.6 Задание

Создать

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 В чем отличие потока от процесса?

8.2 Какие преимущества дает многопоточная архитектура?

8.3 Какие существуют основные средства синхронизации потоков?

8.4 Каким образом на однопроцессорных компьютерах исполняются многопоточные приложения?

8.5 Для чего в C# используется класс Thread?