

## **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 данными 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.

#### **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**

Создать консольное приложение, где в основном потоке Main создается новый поток, исполняющий метод Print, который непрерывно печатает символ 1 в бесконечном цикле. Одновременно главный поток непрерывно печатает символ 0 в бесконечном цикле. В конце каждого шага цикла должна выполняться задержка:

```
Thread.Sleep(1000);
```

Вместо 1000 в разных циклах указать разные значения — поток будет приостанавливаться на указанное количество миллисекунд.

##### **5.2 Настройка приоритетов потоков**

Создать консольное приложение, в который добавить метод WriteString, принимающий объект N и в цикле выводящий его 1000 раз.

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).

##### **5.3 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 первой переменной `str` не будет присвоено значение «х».

#### 5.4 Создание пула потоков

Создать консольное приложение, в которое добавить метод, выполняющий поиск и вывод на экран делителей переданного в метод числа в следующем формате: `у делится нацело на х`. Пример:

6 делится нацело на 1.

6 делится нацело на 2.

6 делится нацело на 3.

6 делится нацело на 6.

Добавить в приложение пул потоков, в который добавить 10 потоков (должны вызывать метод поиска делителей, для каждого потока передавать свое число, например числа от  $N$  до  $N+10$ ).

### 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 выполняются многопоточные приложения?

8.5 Для чего в C# используется класс `Thread`?