**Практикум. Создание очереди рабочих элементов при помощи ThreadPool**

Сейчас вы создадите приложение, использующее пул потоков для создания очереди методов, которые должны быть вызваны в отдельных потоках. Если при выполнении упражнения у вас возникнут затруднения, обратитесь к готовым проектам в папке Code на компакт-диске, прилагаемом к книге.

1. Создайте пустое консольное приложение с именем Thread Pool Demo.

2. Включите (а в Visual Basic импортируйте) пространство имен System. Threading.

3. Создайте метод, просто отображающий текст. Назовите его ShowMyText. Определите один параметр типа object и назовите его state.

4. В методе ShowMyText создайте переменную строку и приведите параметр state к Stringy сохранив его в текстовой переменной.

5. В методе ShowMyText выведите на экран идентификатор ManagedThreadld текущего потока, а новую строку — на консоль.

6. В методе Main консольного приложения создайте новый экземпляр делегата WaitCallback, который ссылается на метод ShowMyText.

7. Используйте ThreadPooly чтобы поместить в очередь несколько вызовов делегата WaitCallback у передавая как состояние различные строки. Полученный код выглядит примерно так:

// c#

using System.Threading;

class Program

{

static void Main(stnng[] args)

{

WaitCallback callback = new WaitCallback(ShowMyText);

ThreadPool.QueueUserWorkItem(callback, ''Hello''), ThreadPool.QueuellserWorkItem(callback, "Hi");

Th read Pool. QueuellserWo rkl tem( cal lback, "Heya"); ThreadPool.QueuellserWorkItetn(callback, "Goodbye");

Console.Read();

}

static void ShowHyText(ob]ect state)

{

string myText = (stnng)state;

Console.WriteLine( Thread: (0) - (1)",

Th read.Си r rentTh read.ManagedTh readld, myText);

}

}

8. Соберите проект, исправьте ошибки. Убедитесь, что консольное приложение успеш­но отображает все вызовы методы ShowMyText на консоли. Обратите внимание, что некоторые рабочие элементы выполняются в разных потоках.