**Практика 1.** **Создание нескольких потоков**

В этом упражнении вы создадите простое консольное приложение и запустите два пото­ка одновременно.

1. Создайте консольное приложение с именем SimpleThreadingDemo.

2. Объявите статический метод с именем Counting.

3. В новом классе импортируйте пространство имен System. Threading.

4. В новом методе создайте цикл for, в котором выполняется отсчет от 1 до 10.

5. В цикле for выведите на экран текущий счет и идентификатор ManagedThreadld теку­щего потока.

6. После вывода данных на консоль приостановите текущий поток на 10 миллисекунд.

7. Вернитесь в метод Main и создайте делегат StartThread со ссылкой на метод Counting.

8. Создайте два потока со ссылками на метод Counting.

9. Запустите оба потока.

10. Объедините потоки, чтобы приложение не завершалось до завершения обоих пото­ков. Полученный код выглядит примерно так:

// C#

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

class Program

{

static void Main(string[J args)

{

ThreadStart starter = new ThreadStart(Counting), Thread first = new Thread(starter);

Thread second = new Thread(starter);

first. StartQ; second.Start();

first.Join(); second.Join();

Console.Read();

}

stdtic void CountingO

{

for (int i = 1; i <= 10; i++)

{

Console.WriteLineC'Count: {0} - Thread' {1}", l, Thread CurrentThread ManagedThreadld);

Thread.Sleep(10);

}

}

11. Соберите проект, исправьте ошибки. Убедитесь, что консольное приложение пока­зывает вывод потоков, работающих параллельно. Это видно по чередованию чи­сел, выводимых потоками, которое зависит быстродействия оборудования компь­ютера. В однопроцессорной системе счет будет идти по порядку, а в многопроцес­сорной (или на компьютере с многоядерным процессором) цифры будут «переме­шаны» сильнее.