**Введение в многопоточность. Класс Thread**

1) Изучить класс Thread, его свойства(CurrentContext, CurrentThread, IsAlive, IsBackground, Name, Priority, ThreadState) и методы (GetDomain, GetDomainID, Sleep, Abort, Interrupt, Join, Start).

**Получение информации о потоке**

2) Используя вышеописанные свойства и методы получить след. информацию о текущем потоке:

- Имя потока

- Запущен ли поток

- Приоритет потока

- Статус потока

- Домен приложения

**Статус потока**

3) Создать поток. Посмотреть статусы до запуска потока, после запуска потока, после остановки потока.

**Приоритеты потоков**

4) По умолчанию какой приоритет имеет поток?

5) Создать поток, изменить его приоритетность на Highest. Запустить поток и вывести информацию о приоритете.

**Создание потоков. Делегат ThreadStart**

6) Посмотреть делегат ThreadStart.

7) Создать метод Count без параметров, который в цикле от 1 до 9 выводит в консоль «Второй поток», квадрат этих чисел и останавливает поток на 400 мс. Далее в методе Main с помощью делегата ThreadStart создать новый поток, передав ему метод Count. Запустить поток. После этого написать в методе Main цикл от 1 до 9 выводит в консоль «Главный поток», квадрат этих чисел и останавливает поток на 300 мс.

**Потоки с параметрами и ParameterizedThreadStart**

8) Познакомится с делегатом ParameterizedThreadStart.

9) Какими ограничениями сталкиваемся при использовании ParameterizedThreadStart?

10) Переделать метод Count так, чтобы он принимал параметры. Далее воспроизвести все, что было написано в пункте 7 с использованием ParameterizedThreadStart

Синхронизация потоков

11) class Program

{

static int x=0;

static void Main(string[] args)

{

for (int i = 0; i < 5; i++)

{

Thread myThread = new Thread(Count);

myThread.Name = "Поток " + i.ToString();

myThread.Start();

}

Console.ReadLine();

}

public static void Count()

{

x = 1;

for (int i = 1; i < 9; i++)

{

Console.WriteLine("{0}: {1}", Thread.CurrentThread.Name, x);

x++;

Thread.Sleep(100);

}

}

}

Запустить код и посмотреть что произойдет

12) Изучить lock. В выше указанном примере сделать синхронизацию потоков.

13) Имеется пустой участок земли (двумерный массив) и план сада, который необходимо реализовать. Эту задачу выполняют два садовника, которые не хотят встречаться друг с другом. Первый садовник начинает работу с верхнего левого угла сада и перемещается слева направо, сделав ряд, он спускается вниз. Второй садовник начинает работу с нижнего правого угла сада и перемещается снизу вверх, сделав ряд, он перемещается влево. Если садовник видит, что участок сада уже выполнен другим садовником, он идет дальше. Садовники должны работать параллельно. Создать многопоточное приложение, моделирующее работу садовников.