## «Платформа Microsoft .NET и язык программирования C#»

## №5 Тема: Многопоточность и асинхронность

# Задание 1. Потоки:

Создайте приложение, генерирующее в побочном потоке набор простых чисел в диапазоне, указанном пользователем.

Если не указана нижняя граница, поток с стартует с 2.

Если не указана верхняя граница, генерирование происходит до завершения приложения. Числа по мере генерации должны отображаться в консоли.

* 1. Добавьте к первому заданию дополнительный поток, генерирующий набор чисел Фибоначчи. Числа должны отображаться по мере генерации.

**Рекомендации:**

Для передачи 2х параметров потоку при старте рекомендуется создать объект-обертку.

Для генерации числа рекомендуется пользоваться объектом System.Math.Random, и его методом Next(min, max)

# Задание 2. ThreadPool:

Создайте приложение, которое будет в цикле передавать задачу по генерации числа от 100 до 2000, выводить номер потока и число в консоль с задержкой равной сгенерированному числу.

# Задание 3. Task:

Создайте приложение с 4 методами, симулирующими асинхронные вызовы:

1. void Request()  
   метод, который выводит в консоль сообщение после небольшой задержки
2. void Process()  
   метод, который принимает Task, ничего с ним не делает, но с вероятностью 10% может вернуть исключение.
3. string Response()  
   метод, который принимает Task и если этот Task завершился неуспешно, то возвращает null, иначе возвращает какую-то константную строку.
4. Void Log()  
   метод, который принимает Task<string> и выводит в консоль сообщение об ошибке, если string == null.

«Упакуйте» вызовы к этим методам последовательно в Task и ContinueWith.

**Рекомендации:**

Вместо Task some = new Task(\_метод\_); some.Start();

Можете использовать Task.Run(\_метод\_);