

## Лабораторная 14. Многопоточность II

Сдать до 14.12

### Задание 1. Race condition (2 балла)

Создайте многопоточное приложение, в котором возможна гонка данных. Приведите пример возникновения гонки (получаются разные ответы). Для демонстрации вы можете использовать искусственные задержки потоков.

### Задание 2. Распечатать по порядку (2 балла)

Пусть есть класс:

```
public class Foo {  
    public void first() { print("first"); }  
    public void second() { print("second"); }  
    public void third() { print("third"); }  
}
```

Один и тот же экземпляр Foo передается трем разным потокам. Поток А вызывает метод first(), поток В вызывает second(), а поток С вызывает third().

Напишите программу так, чтобы гарантировать, что second() выполняется после first(), а third() выполняется после second().

Мы не знаем, как потоки будут запланированы шедулером на выполнение в операционной системе. Числа в примерах ниже указывают на порядок, в котором выполняются потоки (для примера)

Пример 1:

Ввод: nums = [1,2,3]

Вывод: «firstsecondthird».

Объяснение: Асинхронно запускаются три потока. Вход [1,2,3] означает, что сначала вызывается поток А (он вызывает first ()), потом поток В (вызывает second()), а потом поток С (вызывает third ()).

Пример 2:

Ввод: nums = [1,3,2]

Вывод: «firstsecondthird».

Изначально шедулером запланированы на выполнение потоки в последовательности А,С,В, но, тем не менее, вывод мы получаем в нужном порядке.

### Задание 3. Sleepsort (3 балла)

#### А. Сортировка в потоках (1 балл)

Реализуйте алгоритм сортировки с асимптотикой  $O(N)$  (по процессорному времени).

Пусть на вход подается массив из не более 100 строк различной длины. Вам необходимо вывести эти строки, отсортированные по длине. Строки одинаковой длины могут выводиться в произвольном порядке.

Для каждой строки создайте поток и передайте ему эту строку в качестве параметра. Поток должен уснуть на время, пропорциональное длине переданной строки (необходимо вызвать метод Sleep). Затем поток выводит строку на консоль и завершается.

## Б. Вывод в связный список (2 балла)

Переделайте программу так, чтобы потоки не выводили строки на консоль, а формировали из них единый список с порядком записей, соответствующим порядку сортировки. После завершения всех «сортирующих» потоков, один из потоков (не обязательно основной) должен вывести на консоль весь список. Обратите внимание, что доступ к списку необходимо синхронизовать.

## Задание 4. Калькулятор на файлах (3 балла)

В заданной директории лежат входные текстовые файлы. Файлов может быть много (несколько десятков). Каждый файл состоит из двух строк. В первой строке - число, обозначающее действие, а во второй - числа с плавающей точкой, разделенные пробелом.

Действия могут быть следующими:

- 1 - сложение
- 2 - умножение
- 3 - сумма квадратов

Необходимо написать многопоточное приложение, которое выполнит требуемые действия над числами и сумму результатов запишет в выходной файл out.dat. Количество потоков можно задавать параметром. Каждый поток должен получить примерно одинаковое количество файлов для обработки.

## Задание 5. MyCountdownEvent (5 баллов)

Реализуйте с помощью стандартных примитивов синхронизации аналог CountdownEvent.

Ваш класс CMyCountdownEvent должен реализовать следующие методы:

```
public class CMyCountdownEvent : IDisposable
{
    public CMyCountdownEvent(int initialCount);

    public void Signal();
    public void Signal(int signalCount);

    public bool Wait(TimeSpan timeout);

    public void Dispose();
}
```

При создании в конструкторе указывается initialCount. Метод Signal() уменьшает внутренний счетчик count на 1. Метод Signal(int signalCount) уменьшает count на указанное число.

Метод Wait блокирует поток, пока count не станет равным 0. Если timeout истекает до того, как count достигнет 0, то возвращается false, иначе true. Если вызывается Signal после достижения нуля или signalCount превышает оставшиеся счетчики, то генерируется исключение.

Класс должен работать в многопоточном приложении. Выделяемые ресурсы должны освобождаться в методе Dispose.

Разрешается использовать только AutoResetEvent, ManualResetEvent, Mutex, Semaphore и lock().

Продемонстрируйте работу класса CMyCountdownEvent в различных ситуациях.