Задача 1.

Создать класс на языке C#, который:

- называется TaskQueue и реализует логику пула потоков;

- создает указанное количество потоков пула в конструкторе;

- содержит очередь задач в виде делегатов без параметров:

delegate void TaskDelegate();

- обеспечивает постановку в очередь и последующее выполнение делегатов с помощью метода

void EnqueueTask(TaskDelegate task);

Задача 2.

Реализовать консольную программу на языке C#, которая:

- принимает в параметре командной строки путь к исходному и целевому каталогам на диске;

- выполняет параллельное копирование всех файлов из исходного каталога в целевой каталог;

- выполняет операции копирования параллельно с помощью пула потоков;

- дожидается окончания всех операций копирования и выводит в консоль информацию о количестве скопированных файлов.

Задача 3.

Создать класс на языке C#, который:

- называется Mutex и реализует двоичный семафор с помощью атомарной операции Interlocked.CompareExchange.

- обеспечивает блокировку и разблокировку двоичного семафора с помощью public-методов Lock и Unlock.

Задача 4.

Реализовать консольную программу на языке C#, которая:

- принимает в параметре командной строки путь к сборке .NET (EXE- или DLL-файлу);

- загружает указанную сборку в память;

- выводит на экран полные имена всех public-типов данных этой сборки, упорядоченные по пространству имен (namespace) и по имени.

Задача 5.

Создать на языке C# статический метод класса Parallel.WaitAll, который:

- принимает в параметрах массив делегатов;

- выполняет все указанные делегаты параллельно с помощью пула потоков;

- дожидается окончания выполнения всех делегатов.

Реализовать простейший пример использования метода Parallel.WaitAll.

Задача 6.

Создать на языке C# обобщенный (generic-) класс DynamicList<T>, который:

- реализует динамический массив с помощью обычного массива T[];

- имеет свойство Count, показывающее количество элементов;

- имеет свойство Items для доступа к элементам по индексу;

- имеет методы Add, Remove, RemoveAt, Clear для соответственно добавления, удаления, удаления по индексу и удаления всех элементов;

- реализует интерфейс IEnumerable<T>.

Реализовать простейший пример использования класса DynamicList<T> на языке C#.