

# Семинар № 7. Листы и словари

## 1. Практика «Readonly bytes»

Скачайте проект **readonly-bytes**.

Иногда есть смысл в качестве ключей в **Dictionary** или **HashSet** использовать массивы байт. Однако по умолчанию массивы сравниваются по ссылкам, а не по содержимому, а часто нужно именно по содержимому. В таких случаях можно написать класс-обёртку над массивом, который переопределит **Equals** и **HashCode** так, чтобы сравнение происходило по содержимому. В этой задаче вам нужно создать именно такую обёртку.

### Задание:

В файле **ReadonlyBytes.cs** создайте класс **ReadonlyBytes** так, чтобы все тесты из файла **ReadonlyBytesTests.cs** компилировались и проходили.

## 2. Практика «Ghosts»

Скачайте проект **ghost**.

Неаккуратная реализация **Equals** и **GetHashCode** может приводить к тому, что добавленный в **Dictionary** или **HashSet** ключ внезапно исчезает. Чтобы не попадаться на подобные ошибки в будущем, в этом задании предлагается поизучать всевозможные подобные ошибки.

### Задание:

В проекте вам даны несколько классов с уже реализованными **GetHashCode** и **Equals**. Вам нужно придумать, как их использовать, чтобы **HashSet** стал вести себя некорректно.

Изучите тест **GhostsTest.cs** и в файле **GhostsTask.cs** создайте класс **GhostsTask** так, чтобы этот тест проходил.