# Проверяемое задание № 2

**Тема 2. Управление ресурсами в .NET. Сборка мусора**

Лекция 2.1. Управление ресурсами в .NET. Сборка мусора

**Цель работы**

Сформировать понимание принципа работы сборщика мусора и практические навыки использования класса GC. Научиться определять финализатор и понимать необходимость управления ресурсами в управляемом коде, а также использовать паттерн dispose.

**Задание**

1. «Создать новый класс, который будет работать с текстовым файлом как «с массивом символов».
2. a) Создать конструктор, который принимает путь файла и длину «последовательности символов», создает новый файл с «совместным доступом» к чтению и записи, а также в режиме чтения/записи и подготавливает файл: устанавливает длину и заполняет пробелами.

* Для создания потока файла использовать класс FileStream.

1. b) Сделать индексатор, который пишет/читает символ по указанному числовому индексу.

* Для записи в файл использовать класс StreamWriter.
* Для чтения из файла использовать класс StreamReader.

1. c) Файл должен быть в ANSI-кодировке, которая используется по умолчанию в операционной системе.

* Нужно найти подходящее свойство в классе System.Text.Encoding.

d) Реализовать метод Close() для освобождения ресурсов.

* В качестве примера реализации можно рассмотреть реализацию FileStream.Close().

e) Класс должен реализовывать интерфейс IDisposable.

f) Создать свойство Length, возвращающее длину «массива».

g) Доработать класс:

1. - «Создание нового файла»: в данный момент есть конструктор для создания файла, который нужно сделать «закрытым», чтобы нельзя было создать объект класса извне. Необходим открытый статический метод Create(…), который будет создавать через конструктор объект и возвращать созданный объект.
2. - «Чтение существующего файла»: нужно создать новый закрытый конструктор, который будет открывать существующий файл и работать с ним. Следует сделать открытый статический метод Read(…), который будет создавать через соответствующий конструктор объект и возвращать созданный объект.

Приложение должно:

1. a) сохранить в файл строку вида «[01] Привет мир!» с использованием созданного класса; освободить ресурсы с применением try/finally;
2. b) с помощью созданного класса открыть созданный файл (пункт а) и изменить в нём один символ, чтобы в файле получилось «[02] Привет мир!»; освободить ресурсы с применением using». [2]

**Рекомендации по выполнению**

1. Необходимо вовремя сбрасывать буфер объектов работы с файлов на диск.
2. Следует правильно устанавливать длину «массива» и заполнять «файл» при создании «объекта».
3. Должен быть общий метод для открытия и подготовки потока (создающий объекты классов FileStream, StreamWriter, StreamReader).
4. 4. Следует добавить проверку того, чтобы при чтении из файла был действительно прочитан новый символ.

- В противном случае нужно выбрасывать исключение подходящего типа.

1. - Воспроизвести под отладкой такую ситуацию и проверить, чтобы исключение действительно сгенерировалось.
2. 5. Приложение должно корректно обрабатывать ошибки работы с файлами.
3. - Обработку ошибок нужно делать в «пользовательском» классе, а не в созданном «файловом массиве».

**Важные материалы для выполнения задания (!)**

1. Framework Design Guidelines: Dispose Pattern – <http://msdn.microsoft.com/en-us/library/b1yfkh5e.aspx>
2. .NET Framework Development Guide: Implementing a Dispose Method – <http://msdn.microsoft.com/en-us/library/fs2xkftw.aspx>
3. IDisposable.Dispose Method – <http://msdn.microsoft.com/en-us/library/system.idisposable.dispose.aspx>
4. Object.Finalize Method – <http://msdn.microsoft.com/en-us/library/system.object.finalize.aspx>
5. NET Framework Development Guide: Using Objects That Encapsulate Resources – <http://msdn.microsoft.com/en-us/library/3bwa4xa9.aspx>
6. Using Statement (C# Reference) – <http://msdn.microsoft.com/en-us/library/yh598w02(v=vs.80).aspx>
7. Composing Streams – <http://msdn.microsoft.com/en-us/library/e4y2dch9.aspx>

**Требования к предоставлению результатов (!)**

**К отчету следует приложить:**

* листинг программ, написанных на языке программирования C#;
* скриншоты и краткое описание работы классов и методов приложений.