

Лабораторный практикум по дисциплине «ЯЗЫКИ ПРОГРАММИРОВАНИЯ С СТРУКТУРЫ ДАННЫХ»

направление 020302.62 «Фундаментальные информатика и информационные технологии»
семестр 2

Лабораторная работа №5 Основы языка C#: Потоки данных

Теоретический материал:

1. Павловская Т.А. C#. Программирование на языке высокого уровня.

Задание 0.

Прочитать теоретический материал.

Задание 1.

В класс `Vectors` добавить следующие методы:

- записи вектора в байтовый поток:
`void OutputVector(Vector v, Stream out);`
- чтения вектора из байтового потока:
`Vector InputVector(Stream in).`

Записанный вектор должен представлять собой последовательность чисел, первым из которых является размерность вектора, а остальные числа являются значениями координат вектора.

Проверить работоспособность методов в классе `Program`, в качестве байтового потока используя файловый поток (создать файл данных в текущей папке).

Задание 2.

Добавить в класс `Vectors` следующие методы:

- записи вектора в символьный поток:
`void WriteVector(Vector v, TextWriter out);`
- чтения вектора из символьного потока:
`Vector ReadVector(Reader in).`

В данном случае рекомендуется считать, что один вектор записывается в одну строку (числа разделены пробелами, можно использовать переопределенный метод `ToString()`). Для чтения вектора из символьного потока рекомендуется использовать метод `Split()` класса `String`.

Проверить работоспособность методов в классе `Program`, в качестве текстового потока используя файловый поток (создать текстовый файл в текущей папке).

Задание 3.

Модифицировать классы `ArrayVector` и `LinkedListVector` таким образом, чтобы они были сериализуемыми.

Продемонстрировать возможности сериализации в классе `Program`, записав в файл объект, затем считав и сравнив его с исходным, для чего вывести содержимое обоих

объектов на экран, можно также использовать метод `Equals()`, унаследованный от класса `Object`.

Задание 4.

Протестировать работу приложения в классе `Program`, разработать адекватный интерфейс пользователя. Необходимо отлавливать и обрабатывать все возможные исключения. Воспользоваться пользовательским интерфейсом из лабораторной работы 4.

Задание 5.

Подготовить отчет о работе.