

Лабораторный практикум по дисциплине «ОСНОВЫ ПРОГРАММИРОВАНИЯ»

направление 020302.62 «Фундаментальная информатика и информационные технологии»
семестр 2

Лабораторная работа №1 Основы языка C#: Классы. Повторение

Теоретический материал:

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

Задание 1.

Прочитать необходимый теоретический материал.

Задание 2.

Описать класс с именем «ArrayVector». Класс описывает вектор в n-мерном пространстве. Координаты конца вектора задаются массивом, количество элементов которого равно n – размерности пространства. Класс должен иметь следующую структуру:

- поле – массив элементов целого типа (координаты вектора в пространстве);
- конструктор с параметром – длиной массива;
- конструктор без параметров, создающий массив из 5 элементов;
- метод установки элемента массива по индексу SetElement() (параметры метода – индекс элемента и устанавливаемое значение);
- метод чтения элемента массива по индексу GetElement() (параметр метода – индекс элемента);

Примечание: вместо методов SetElement() и GetElement() можно реализовать индексатор – он будет необходим в последующих лабораторных работах.

- метод GetNorm() получения модуля (длины, нормы) вектора;
- метод SumPositivesFromChetIndex() подсчета суммы всех положительных элементов массива с четными номерами;
- метод SumLessFromNechetIndex() подсчета суммы тех элементов массива, которые имеют нечетные номера и одновременно меньше среднего значения всех модулей элементов массива;
- метод MultChet() подсчета произведения всех четных положительных элементов (по значению);
- метод MultNechet() подсчета произведения всех нечетных элементов (по значению), не делящихся на три;

Примечание: во всех четырех методах производящих действия с элементами массива нумерация элементов массива для конечного пользователя должна начинаться с единицы. То есть, в массиве [2, 3, 4, 5] элемент со значением «3» это второй элемент по индексу с точки зрения пользователя.

- метод SortUp() сортировки массива по возрастанию;
- метод SortDown() сортировки массива по убыванию.

Где необходимо выбрасывать разные типы исключений.

Задание 3.

Добавить класс с именем «Vectors», содержащий публичные статические методы:

- сложения двух векторов `Sum()`, который принимает в качестве параметра 2 объекта типа `ArrayVector` и возвращает новый объект `ArrayVector`;
- скалярного произведения двух векторов `Scalar()`, который принимает в качестве параметра 2 объекта типа `ArrayVector` и возвращает вещественное число;
- умножения вектора на число `MultNumber()`, который принимает в качестве параметра объект типа `ArrayVector` и вещественное число и возвращает новый объект `ArrayVector`.
- получения модуля (длины) вектора `GetNormSt()`, который принимает в качестве параметра объект типа `ArrayVector` и возвращает вещественное число.

Выбрасывать исключения в методах `Sum()` и `Scalar()` в случае невозможности проведения указанных действий над векторами (возможно, `FormatException`).

В классе `Program` в методе `Main()` реализовать всю функциональность описанного класса – написать программу, проверяющую все (!) разработанные элементы класса.

Задание 4.

Подготовить отчет о работе.