

**Лабораторный практикум по дисциплине**  
**«ЯЗЫКИ ПРОГРАММИРОВАНИЯ С СТРУКТУРЫ ДАННЫХ»**  
направление 020302.62 «Фундаментальные информатика и информационные технологии»  
семестр 2

**Лабораторная работа №2**  
**Основы языка C#: Односвязный список**

Теоретический материал:

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

Задание 1.

Прочитать теоретический материал.

Задание 2.

Привести класс «вектор» с именем “ArrayVector” к следующей структуре:

- поле – массив элементов целого типа (координаты конца вектора в n-мерном пространстве);
- конструктор с параметром – длиной массива;
- конструктор без параметра, задающий длину массива 5;
- индексатор для организации доступа к элементам массива, выбрасывающий исключение при вводе некорректного индекса;
- метод GetNorm() вычисления модуля/нормы вектора;
- свойство для чтения числа координат вектора.

Где необходимо выбрасывать, исключения в случае невозможности проведения указанных действий над векторами.

Задание 3.

Описать класс «односвязный список» с именем «LinkedListVector», содержащий координаты конца вектора в n-мерном пространстве в виде динамического односвязного списка. Каждый элемент массива представляет собой отдельный объект «узел» класса «Node», класс Node является внутренним для класса LinkedListVector.

Структура класса «Node»:

- поле – элемент целого типа (по умолчанию = 0);
- поле – ссылка на элемент класса Node (по умолчанию = null);

Структура класса «LinkedListVector»:

- поле – ссылка на начало списка (на экземпляр класса Node);
- конструктор с параметром – длиной списка;
- конструктор без параметра, задающий длину списка 5;
- индексатор для организации доступа к элементам списка, выбрасывающий исключение при некорректном индексе;
- метод GetNorm() вычисления модуля вектора;
- свойство для чтения числа координат вектора;

а также методы:

- удаления и добавления элемента в конец и в начало;
- удаления и добавления элемента в заданную позицию.

Где необходимо выбрасывать, исключения в случае невозможности проведения указанных действий над векторами.

#### Задание 4.

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

- Sum() сложения двух векторов, который принимает в качестве параметра 2 объекта ArrayVector и возвращает новый объект ArrayVector;
- Scalar() скалярного произведения двух векторов, который принимает в качестве параметра 2 объекта ArrayVector и возвращает целое число;
- GetNormSt() получения модуля/нормы вектора, который принимает в качестве параметра объект ArrayVector и возвращает вещественное число.

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

#### Задание 5.

Проверить функциональность классов в методе Main() класса Program. Разработать адекватный пользовательский интерфейс.

Отлавливать все возможные исключения – некорректный ввод пункта меню, некорректный ввод данных, несовпадение длин векторов в статических методах класса Vectors и т.д. То есть сделать программу правильно реагирующей на предсказуемые ошибки.

#### Задание 6.

Подготовить отчет о работе.