

## Лабораторная работа №16: «Реализация механизма создания коллекции и применение коллекций в программах»

**Цель:** научиться разрабатывать коллекции на языке программирования C# и применять их в программах.

**Пример 1.** Пример применения параметризованной коллекции List <T>.

В предложенном примере реализована коллекция на основе параметризованной коллекции List<T>. Коллекция mas содержит список (аналог одномерного массива) экземпляров (объектов) классов типа Monster. В данной коллекции mas можно хранить элементы класса Monster, а также любого класса, производного от Monster. Другие типы данных в коллекции mas недопустимы.

Проект Lab16\_Фамилия

```
using System;
using System.Collections.Generic;
using ClassLibrary1; //описание классов Monster и Daemon смотри в Л.р. 15

namespace Lab16_Фамилия
{
    class Program
    {
        static void Main()
        {
            //объявление экземпляра коллекции List с параметром – типом данных – классом Monster
            List<Monster> mas = new List<Monster>();
            //добавление в коллекцию объектов класса Monster и производного от него Daemon
            mas.Add(new Monster(1,2,"Вася"));
            mas.Add(new Daemon(2,2,"Демон", 3));
            //вывод всех элементов в коллекции
            foreach (Monster x in mas)
                x.Passport();
        }
    }
}
```

**Задание 1.** Создайте в проекте Lab16\_Фамилия коллекцию mas1, содержащую список объектов типа int. Добавьте в коллекцию три элемента, ваши данные: Число\_рождения, Месяц\_рождения, Год\_рождения. Выведите элементы коллекции на экран.

**Задание 2.** Создайте в проекте Lab16\_Фамилия коллекцию mas2, ваши данные: Фамилия, Имя, Отчество. Выведите элементы коллекции на экран.

**Задание 3.** Создайте в проекте Lab16\_3\_Фамилия параметризованный метод Sort с параметром – типом данных. Реализуйте в нем **сортировку методом выбора** (алгоритм ее состоит в том, что сначала выбирается наименьший элемент массива и меняется местами с первым элементом. Затем просматриваются элементы, начиная со второго, и наименьший из них меняется местами со вторым элементом и так далее. Всего произведено будет n-1 замен. На последнем проходе цикла при необходимости меняются местами предпоследний и последний элементы массива). На данный метод накладывается ограничение – объекты можно сравнивать друг с другом с помощью метода CompareTo.

Отсортируйте с помощью данного метода следующие массивы:

a = { 1, 6, 4, 2, 7, 5, 3 }

b = { 1.1, 6.6, 4.4, 2.4, 7.6, 5.5, 3.3 }

c = { sdef, sd, sdfsd, sdf }

### Методические рекомендации к заданию 3

#### *Обобщенные(параметризованные) методы, ограничения на использование параметризованных типов*

##### **Ограничения**

Класс-прототип может содержать произвольное число параметров типа. Для каждого параметра можно задать ограничения, указывающие каким требованиям должен удовлетворять аргумент, соответствующий этому параметру. Например, можно указать, что это будет тип, использующий некоторый интерфейс.

Ограничения задаются после ключевого слова `where`, например:

```
public class Stack<T>  
where T:struct {...}
```

Здесь задано ограничение, что класс стек может использовать элементы только значимого типа (типа структуры). Для ссылочного типа необходимо использовать ключевое слово `class`.

Указание в качестве ограничений имени класса означает, что соответствующий параметр (аргумент) должен быть инициализирован или именем этого класса, либо его потомка.

##### **Контрольные вопросы:**

1. Что называют параметризованными коллекциями?
2. Какие имеются преимущества при использования параметризованных классов и методов?