РК1 Кашурин Максим ИУ5-34Б

Особенности сортировки коллекций в языке C#

Для большинства коллекций определен метод Sort, выполняющий сортировку коллекции. Если обобщенная коллекция основана на типе значении то это не вызывает проблемы, так как правила упорядочивания элементов для них известны.

Пример сортировки коллекций целых чисел:

```
Console.WriteLine("\nСортировка списка целых чисел:");

List col = new List();

col.Add(5); col.Add(3); col.Add(2); col.Add(1); col.Add(4); col.Add(7); col.Add(6);

 Console.WriteLine("\nПеред сортировкой:");

 foreach (int i in col)// пробегаемся по значениям

 Console.Write(i.ToString() + " "); //до сортироки исходные значения коллекции

 col.Sort();

 Console.WriteLine("\nПосле сортировки:");

foreach (int i in col)

 Console.Write(i.ToString() + " ");
```

Результаты вывода в консоль: Сортировка списка целых чисел:

Перед сортировкой: 5 3 2 1 4 7 6

После сортировки: 1 2 3 4 5 6 7

Если мы попытаемся отсортировать объекты ссылочного типа в качестве элементов коллекции, то возникает небольшая проблема реализации.

Рассмотрим ранее выполненную лабораторную работу:

```
Rectangle rect = new Rectangle(5, 4);

Square square = new Square(5);

Circle circle = new Circle(5);
```

Добавление в список: List fl = new List();

```
 fl.Add(circle);
```

fl.Add(rect);

fl.Add(square);

При вызове метода сортировки: fl.Sort(); генерируется исключение InvalidOperationException с сообщением «сбой при сравнении двух элементов массива». Вызвано это тем, что в классе геометрической фигуры и ее наследниках нет информации о том, как сравнить элементы, что необходимо для сортировки элементов.

Использование для коллекции метода Sort предполагает, что элементы коллекции реализуют интерфейс IComparable. Наследование от данного интерфейса предполагает реализацию метода CompareTo, осуществляющего сравнение двух объектов.

public int CompareTo(object obj)

{ //Приведение параметра к типу "фигура"

Figure p = (Figure)obj; //Сравнение 139

if (this.Area() < p.Area()) return -1;

else if (this.Area() == p.Area()) return 0;

else return 1; //(this.Area() > p.Area()) }

Наследование класса Figure от интерфейса IComparable является обязательным. Если просто реализовать метода CompareTo, без наследования от класса, то все равно будет ошибка. Метод Sort проверяет факт наследования класса элемента списка от интерфейса IComparable.

abstract class Figure : IComparable именно этой строчкой выявлена наследование класса от интерфейса, сам метод реализовывать не нужно тк он автоматически наследуется от интерфейса.

После этих строк исключений возникнуть не должно , поэтому можно будет написать код сортирои объектов, который буде корректно работать.

List fl = new List();

fl.Add(circle);

fl.Add(rect);

fl.Add(square);

Console.WriteLine("\nПеред сортировкой:");

foreach (var x in fl) Console.WriteLine(x); //сортировка

fl.Sort();

```
Console.WriteLine("\nПосле сортировки:");
foreach (var x in fl) Console.WriteLine(x);
```

Следовательно, для сортировки коллекций, содержащих объекты классов, необходимо реализовывать в данных классах интерфейс IComparable.