

# Лабораторная работа №10

## Использование стандартных интерфейсов

### 1 Цель работы

1.1 Изучить процесс реализации стандартных интерфейсов на языке C#.

### 2 Литература

2.1 <https://metanit.com/sharp/tutorial/> – гл.7.

### 3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

### 4 Основное оборудование

4.1 Персональный компьютер.

### 5 Задание

Задания выполняются по вариантам (вариант = номер ПК). Задание должно выполняться в классе, описание которого дано в таблице 1 (достаточно, чтобы у класса были автосвойства).

Таблица 1 — Варианты для задания 2

№	Название и автосвойства класса
1	Студент (ФИО, специальность, курс)
2	Абитуриент (ФИО, класс, средний балл)
3	Авиарейс (пункт назначения, номер рейса, вместимость)
4	Книга (название, автор, цена)
5	Работник (ФИО, должность, зарплата)
6	Поезд (номер поезда, пункт назначения, количество свободных мест)
7	Товар (тип, наименование, цена)
8	Пациент (ФИО, номер полиса, год рождения)
9	Питомец (кличка, порода, возраст)
10	Заказчик (ФИО, адрес, потраченная сумма)
11	Файл (имя файла, полный путь к файлу, размер файла)
12	Статья (название, автор, год публикации)
13	Квартира (адрес, количество комнат, площадь)
14	Абонент (ФИО, номер телефона, время разговоров)

#### 5.1 Реализация стандартного интерфейса IComparable (без обобщений)

Реализовать интерфейс IComparable в классе из таблицы 1 для того, чтобы объекты сравнивались по одному из числовых полей (выбирать последнее поле в списке, например, в варианте 1 по полю курс, в варианте 9 по полю возраст).

Для выполнения сравнения в методе `public int CompareTo(object obj)` привести объект `obj` к типу разработанного класса. Далее можно использовать условный оператор для сравнения значения полей объектов или вызывать метод `CompareTo` у поля текущего объекта.

Для проверки в Program.Main создать массив объектов класса и отсортировать этот массив, вызвав Array.Sort(массив). Если все сделано корректно, массив будет отсортирован по возрастанию значений поля. На экран должен быть выведен исходный и отсортированный массив.

### 5.2 Реализация стандартного интерфейса IComparable<T> (с обобщениями)

Реализовать интерфейс IComparable в классе из таблицы 1 для того, чтобы объекты сравнивались по одному из числовых полей (выбирать последнее поле в списке, например, в варианте 1 по полю курс, в варианте 9 по полю возраст).

Для выполнения сравнения в методе public int CompareTo(тип obj) использовать условный оператор для сравнения значения полей объектов или вызывать метод CompareTo у поля текущего объекта.

Для проверки в Program.Main создать массив объектов класса и отсортировать этот массив, вызвав Array.Sort(массив). Если все сделано корректно, массив будет отсортирован по возрастанию значений поля. На экран должен быть выведен исходный и отсортированный массив.

### 5.3 Реализация стандартного интерфейса IEquatable<T> (с обобщениями)

Реализовать интерфейс IEquatable в классе из таблицы 1 для того, чтобы объекты сравнивались по всем полям (истина должна возвращаться при равенстве всех значений полей).

Для выполнения сравнения в методе public bool Equals(тип obj) использовать условный оператор для сравнения значения полей объектов.

Для проверки в Program.Main создать два объекта класса и сравнить эти объекты. На экран вывести результат сравнения.

### 5.4 Реализация стандартного интерфейса ICloneable

Реализовать интерфейс ICloneable в классе из таблицы 1 для того, чтобы выполнялось копирование объекта.

Для проверки в Program.Main создать два объекта класса и клонировать один из объектов. Проверить, как изменяются значения клонированного объекта и неклонированного объекта.

### 5.5 Реализация стандартного интерфейса IComparer<T> (с обобщениями)

Реализовать интерфейс IComparer в классе DescendingComparer для того, чтобы сортировать по убыванию числового значения.

Реализовать интерфейс IComparer в классе StringComparer для того, чтобы сортировать по возрастанию первого строкового значения.

Для проверки в Program.Main создать массив объектов класса и отсортировать этот массив, вызвав Array.Sort(массив, new ИмяComparer()). Если все сделано корректно, массив будет отсортирован по выбранному компаратору. На экран должен быть выведен исходный и отсортированный массив.

## 6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C# (Console Application).

6.2 Выполнить все задания из п.5 в одном решении LabWork10, каждое — в своем проекте (классы и интерфейсы должны быть описаны в отдельных файлах .cs и

протестированы в методе Program.Main).

При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать.

При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

6.3 Ответить на контрольные вопросы.

## **7Содержание отчета**

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

## **8Контрольные вопросы**

8.1 Для чего используется интерфейс Comparable?

8.2 Для чего используется интерфейс Equatable?

8.3 В чем отличие между обобщенным и необобщенным интерфейсами?