

## Практическая работа № 6

**Цель работы:** изучить принципы работы с интерфейсами

Интерфейс представляет ссылочный тип, который может определять некоторый функционал - набор методов и свойств без реализации. Затем этот функционал реализуют классы и структуры, которые применяют данные интерфейсы.

### Определение интерфейса

Для определения интерфейса используется ключевое слово **interface**. Как правило, названия интерфейсов в C# начинаются с заглавной буквы **I**, например, **IComparable**, **IEnumerable** (так называемая венгерская нотация), однако это не обязательное требование, а больше стиль программирования.

Что может определять интерфейс? В целом интерфейсы могут определять следующие сущности:

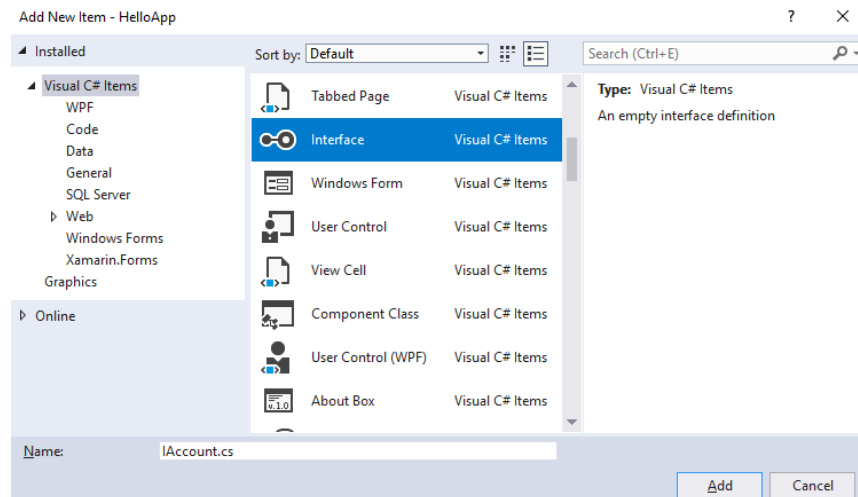
- Методы
- Свойства
- Индексаторы
- События
- Статические поля и константы (начиная с версии C# 8.0)

Как и классы, интерфейсы по умолчанию имеют уровень доступа **internal**, то есть такой интерфейс доступен только в рамках текущего проекта. Но с помощью модификатора **public** мы можем сделать интерфейс общедоступным:

```
public interface IMovable
{
    void Move();
}
```

Стоит отметить, что в Visual Studio есть специальный компонент для добавления нового интерфейса в отдельном файле. Для добавления интерфейса в проект можно нажать правой кнопкой мыши на проект и в появившемся контекстном меню выбрать **Добавить-> Новый элемент** и в

диалоговом окне добавления нового компонента выбрать пункт **Interface**:



## Задание

### Основное

За основу для задания взять интерфейсы **IIndexable** и **ISeries**. Требуется реализовать интерфейс «IndexableSeries» таким образом, чтобы получить арифметическую прогрессию. Продемонстрировать работу класса с помощью метода PrintSeries.

```
public interface IIndexable
{
    double this[int index] { get; }
}

public interface ISeries
{
    double GetCurrent();
    bool MoveNext();
    void Reset();
}
```

### Требования

1. Создать интерфейс IndexableSeries, который наследует интерфейсы ISeries и IIndexable.
2. Создать классы ArithmeticProgression и GeometricProgression, которые реализуют интерфейс IndexableSeries.
  1. ArithmeticProgression – класс реализующий [Арифметическую прогрессию](#)
  2. GeometricProgression – класс реализующий [Геометрическую прогрессию](#)

1. В классе Program создать публичный статический метод PrintSeries. Метод должен принимать два параметра:
  - типа ISeries – ссылка на интерфейс;
  - типа int – сколько выводить элементов последовательности.
4. Метод PrintSeries должен печатать в консоль заданное количество элементов – каждый элемент в новой строке.
5. В классе Program создать публичный статический метод PrintIndexable. Метод должен принимать два параметра:
  - типа Indexable – ссылка на интерфейс;
  - типа int – сколько выводить элементов последовательности.
6. Метод PrintIndexable должен печатать в консоль заданное количество элементов – каждый элемент в новой строке.
7. В методе Main создать экземпляры классов арифметической прогрессии и списка, и вывести некоторое количество элементов каждой последовательности с помощью методов PrintSeries и PrintIndexable.
8. Добавить совой проект на GitHub

### **Полезные источники:**

[Интерфейсы](#)

[Уроки C#. Интерфейсы.](#)