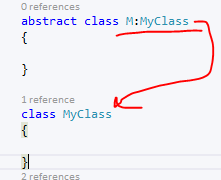


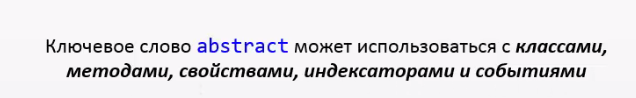
Может быть

protected abstract void AbstractMethod();

интерфейс это контракт который приводит к типу (тип пешеход, тик сын, тип друг…)

Абстрактный класс может наследовать класс





Абстрактный и конкретный классы

В абстрактном классе не может быть конструктора с параметрами

Абстрактных методов технически не существует это виртуальные методы

Абстрактный метод реализуется или переопределяется

// Абстрактный класс.

abstract class AbstractClass

{

public abstract void Method(); на самом деле это public virtual void Method() { }

}

// Конкретный класс.

class ConcreteClass : AbstractClass

{

public override void Method() обязательно должно присутствовать

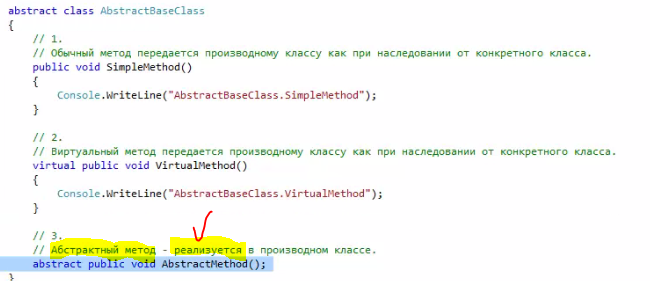
{

Console.WriteLine("Implementation");

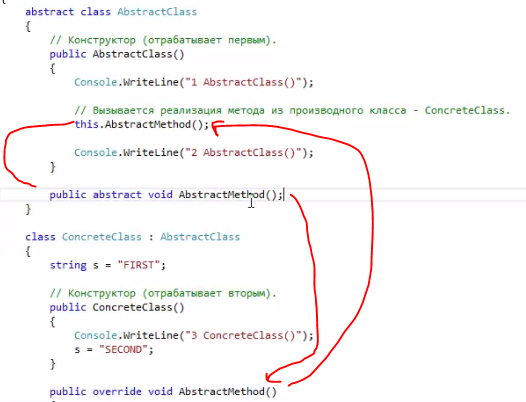
}

}

Виртуальные, обычные и абстрактные методы в производном классе

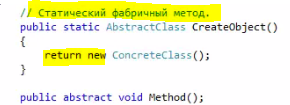


Техника шаблонного метода – когда в абстрактном классе вызывается абстрактный метод который реализован в производном классе



Абстрактный класс может в себе содержать статические члены

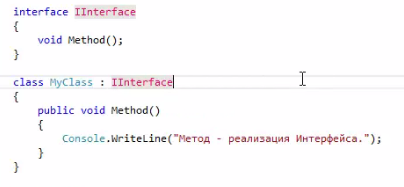
В данном примере статический метод (ФАБРИЧНЫЙ) возвращает экземпляр класса который наследуется от абстрактного класса и приводится к абстрактному классу





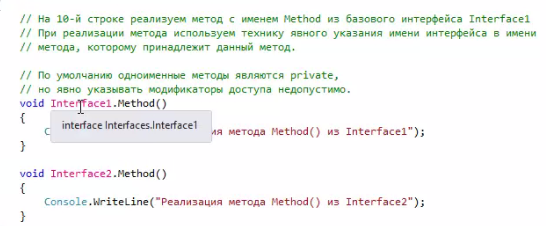
Интерфейс – набор сигнатур методов (имя метода и параметры которые принимает метод)

Реализуем интерфейс в классе MyClass и реализуем абстрактный метод интерфейса



Явная указание имени интерфейса в имени методу которому он принадлежит

Когда имеются интерфейсы с одинаковыми именами то ОБЯЗАТЕЛЬНО нужно указывать к какому интерфейсу он относится



Если в интерфейсах и в базовом классе одинаковые имена методов то

class DerivedClass : BaseClass, Interface1, Interface2

{

void Interface1.Method()

{

Console.WriteLine("Реализация метода Method1() из Interface1");

}

void Interface2.Method()

{

Console.WriteLine("Реализация метода Method2() из Interface2");

}

public new void Method()

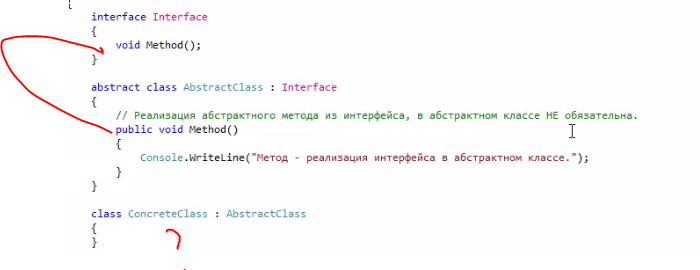
{

Console.WriteLine("Реализация метода Method1() из Interface1");

}

}

Если абстрактный класс наследует интерфейс то реализация абстрактных членов не обязательна



Но нужно сделать вот так

