C# Essential

Наследование и полиморфизм

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

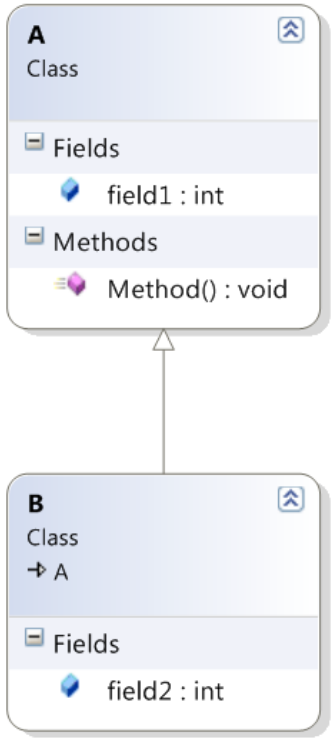
Тема

Наследование и полиморфизм

Наследование

Парадигма ООП

Наследование – механизм объектно-ориентированного программирования (наряду с инкапсуляцией, полиморфизмом и абстракцией), позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.




```
class A
{
    public int field1;
    public void Method()
    {
        /* … */
    }
}


class B : A
{
    public int field2;
}
```

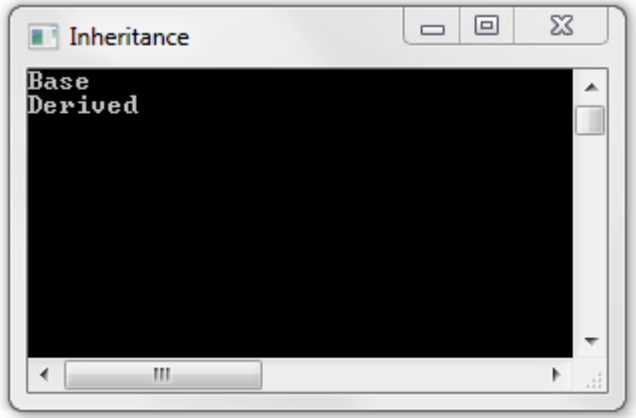
```
static void Main()
{
    B b = new B();
    b.field1=5;
    b.field2=8;
    b.Method();
}
```

Вызов конструктора базового класса

Использование ключевого слово `base` для вызова конструктора базового класса.

```
public class BaseClass
{
    public BaseClass()
    {
        Console.WriteLine("Base");
    }
}

public class DerivedClass : BaseClass
{
    public DerivedClass()
        : base()
    {
        Console.WriteLine("Derived");
    }
}
```



Приведение типов

Приведение к базовому типу

Приведение к базовому типу используется для сокрытия реализации членов производного класса.

```
BaseClass instance = new DerivedClass();
```

Переменная `instance` типа `BaseClass` хранит ссылку на экземпляр класса `DerivedClass`.

Приведение типов

UpCast и DownCast

UpCast – приведение экземпляра производного класса к базовому типу.

```
BaseClass up = new DerivedClass();
```

DownCast – приведение экземпляра базового типа к производному типу.

```
DerivedClass down = (DerivedClass) up;
```

DownCast невозможен без предварительного UpCast.

Полиморфизм

Парадигма ООП

Полиморфизм – возможность объектов с одинаковой спецификацией иметь различную реализацию.

Формы полиморфизма:

1. Ad-hoc полиморфизм;

2. Классический (принудительный) полиморфизм:
   - использование виртуальных членов (переопределение `virtual` / `override`);
   - приведение типов.

В случае одновременного использования двух форм классического полиморфизма, первая форма нейтрализует вторую (доминирует над второй).

sealed

Модификатор

При применении к классу, модификатор `sealed` запрещает другим классам наследоваться от этого класса.

Модификатор `sealed` можно использовать с методами или свойствами. Это позволяет запретить переопределять виртуальные методы или свойства в производных классах.

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association