

# C# Essential

Структуры и их разновидности. Перечисления

# C# Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал  
на [TestProvider.com](http://testprovider.com)

# Структуры и их разновидности. Перечисления

# Упаковка-Распаковка

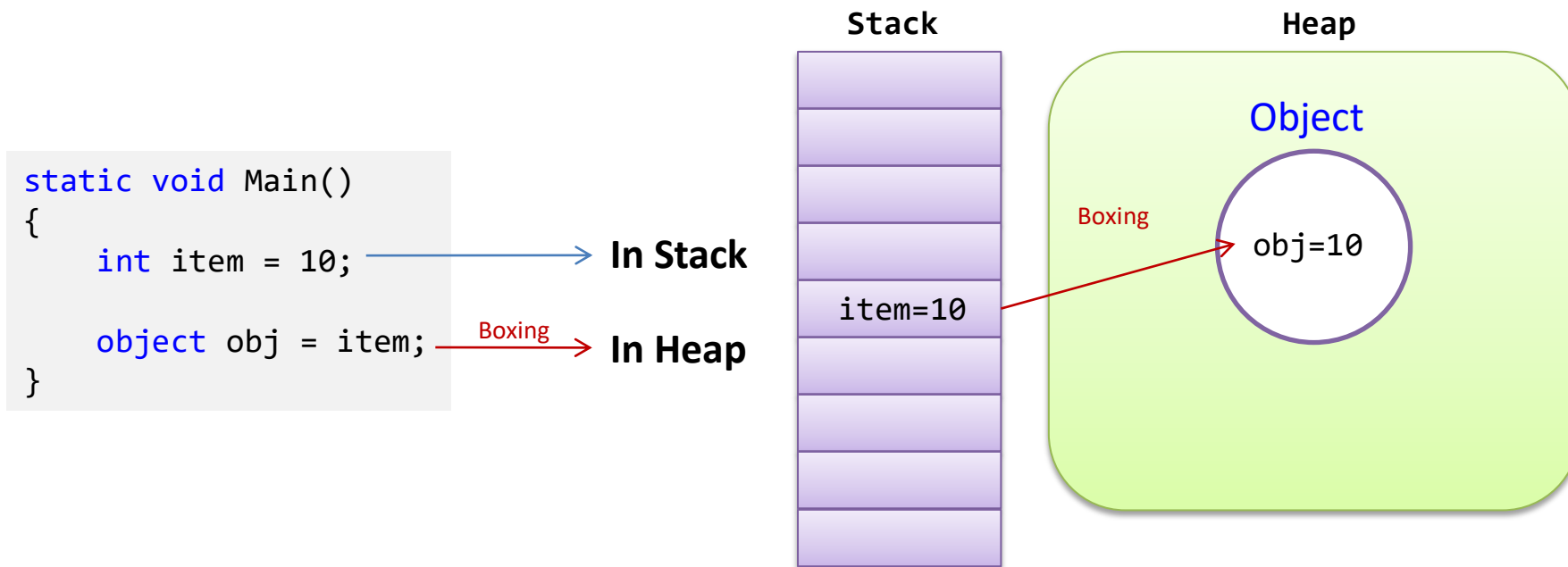
## Boxing / UnBoxing

Упаковка представляет собой процесс преобразования структурного типа в тип `object` или любой другой тип интерфейса, реализуемый этим типом. Операция распаковки извлекает структурный тип из объекта.

# Упаковка

## Boxing

Упаковка-преобразование является неявным:

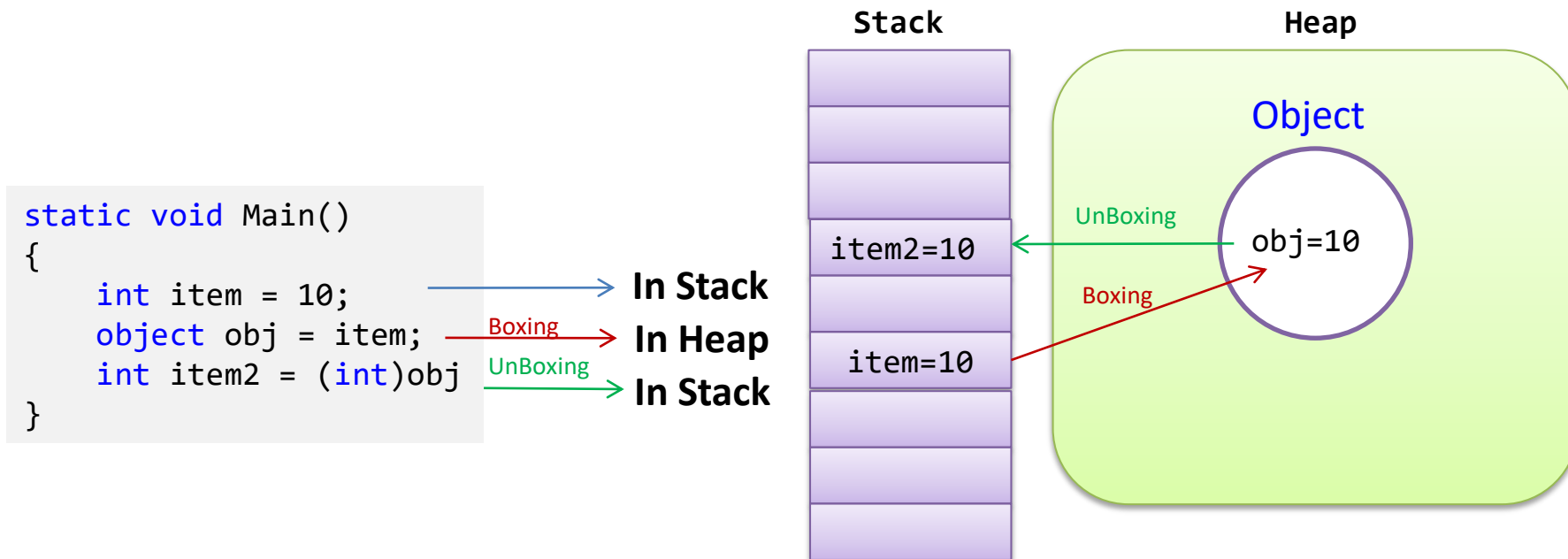


Стек (англ. stack – стопка) – структура данных с методом доступа к элементам **Last In – First Out** «последним пришел – первым вышел».

# Упаковка

## UnBoxing

Распаковка-преобразование является явным.



# Ковариантность

## Array Covariant

Ковариантность не применима к массивам элементов структурных типов.

```
static void Main()
{
    Dog[] dogs = { new Dog(), new Dog(), new Dog() };

    //IAAnimal[] animal = dogs; // Ковариантность.
    //dogs = array; // Контрвариантность.

    int[] vector = new int[3] { 1, 2, 3 };
    //object[] array = vector; // Ковариантность
}
```

# Дата и время

## DateTime

- Структура **DateTime** представляет текущее время, обычно выраженное как дата и время суток.
- Тип значения **DateTime** представляет дату и время в диапазоне от 00:00:00 1 января 0001 года (н. э.) и до 23:59:59 31 декабря 9999 года (н. э.)



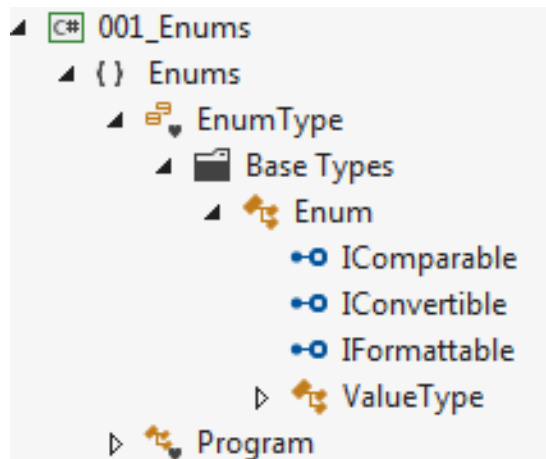
**DateTime.Now** – возвращает объект **System.DateTime**, которому присвоены текущие дата и время суток данного компьютера.



# Перечисление

## Enum

**Перечисление** — это конструкция языка, которая содержит в себе набор именованных констант, которые хранят в себе определённое значение.



Перечисления наследуются от **Enum**, который наследуется от **ValueType**, поэтому они относятся к категории структурных типов.

# Перечисление

## Тип перечисления

Перечислимый тип определяется как набор идентификаторов, с точки зрения языка играющих ту же роль, что и обычные именованные константы, но связанные с этим типом.

```
enum EnumType : byte
{
    Zero = 0,
    One = 1,
    Two = 2,
    Three = 3
}
```

Явно указан тип перечисления `byte`



Рекомендуется использовать `int` как основной тип перечисления.

# Перечисление

## Преимущества

Использование перечислений позволяет сделать исходные коды программ более читаемыми, так как позволяют заменить «**магические числа**», кодирующие определённые значения на читаемые имена.

# Перечисление

## Рекомендации

*«Непродуманное использование перечислений приводит к демографическому взрыву среди типов, что, в свою очередь, ведёт не к ясности программ, а к многословию»*

***(Никлаус Вирт)***

# C# Essential

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

