Курс “Языки программирования”

Лекция 02.

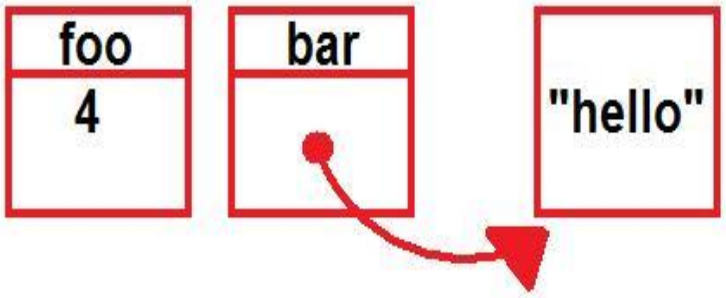
C# как объектно-ориентированный язык.
Типы данных.
Ссылочные и значимые типы.
Приведение типов.

Типы данных

Значимые типы (Value types) хранят значение (переменная значимого типа непосредственно хранит данные).

Ссылочные типы (Reference types) хранят ссылку на значение (ссылка указывает на определенную область памяти, где хранятся фактические данные).

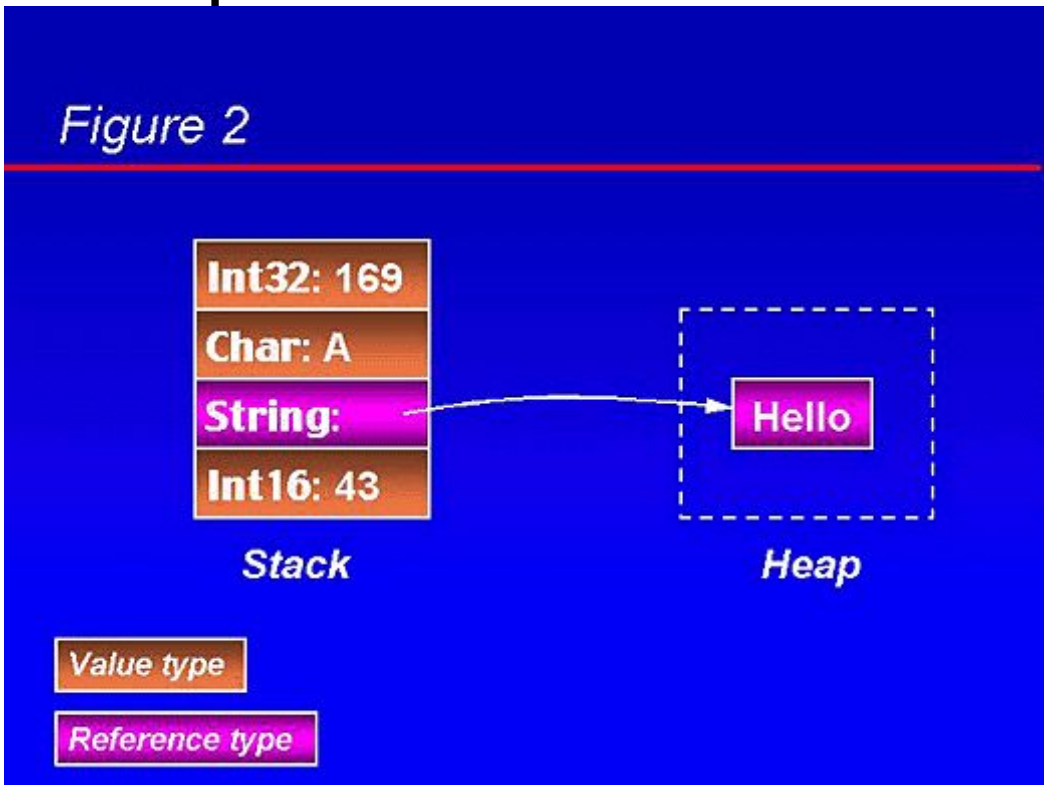
```
int foo = 4;
string bar = "hello";
```



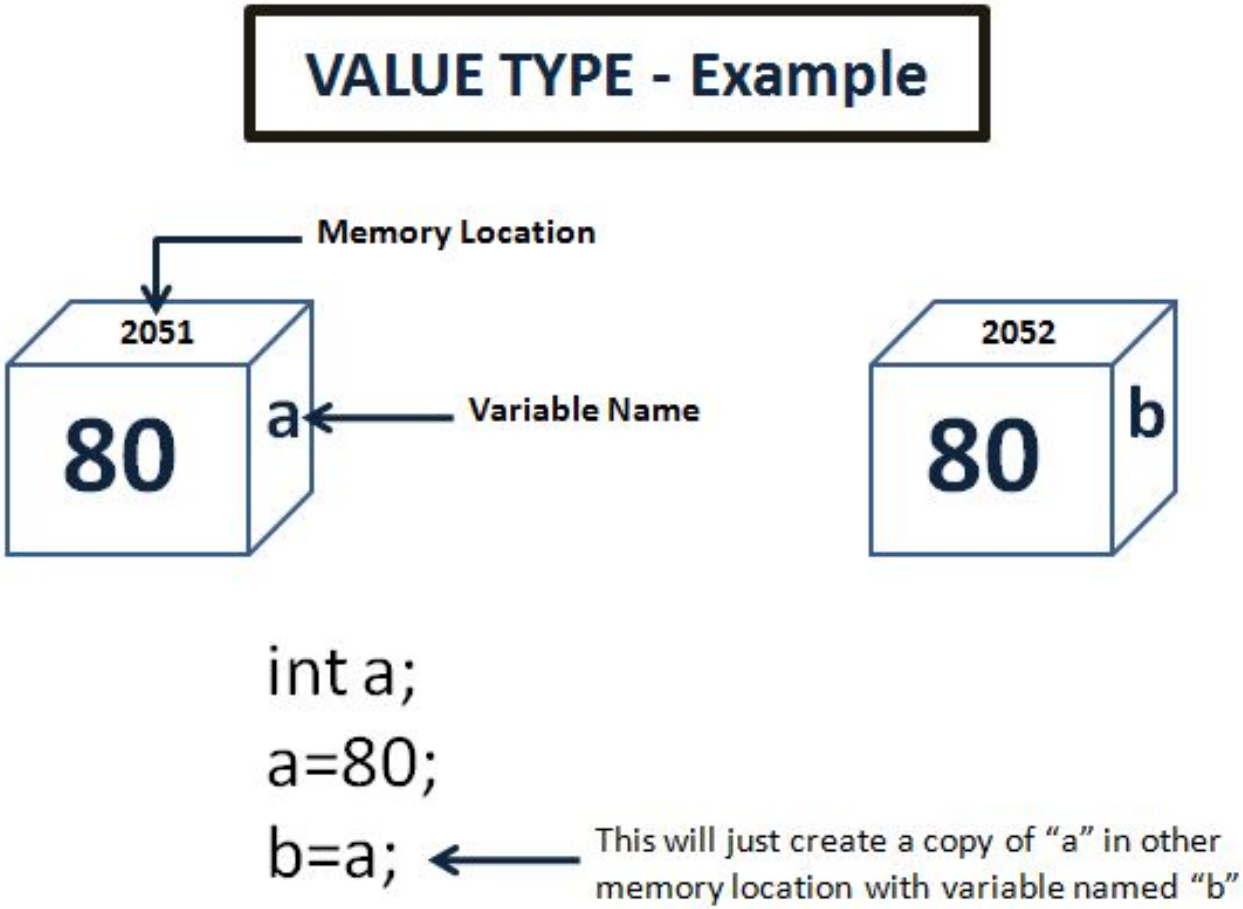
Типы данных

1. Объекты ссылочных типов всегда хранятся в области памяти, называемой управляемой кучей (Heap)

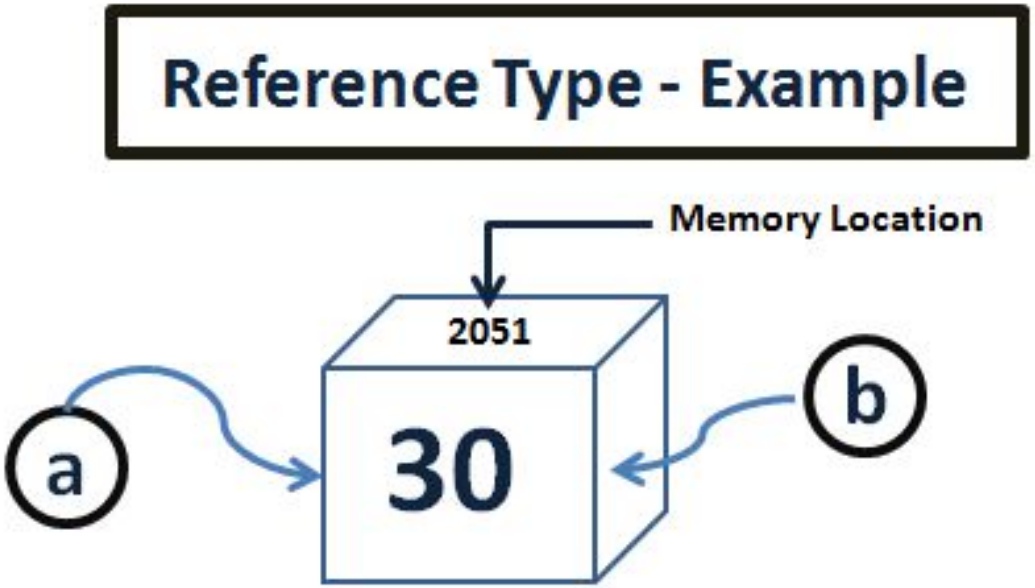
2. Значимые типы хранятся в стеке потока (Stack). Стек потока – это область памяти, которая используется для передачи параметров в методы и хранения определенных в пределах методов локальных переменных.



Значимый тип



Ссылочный тип



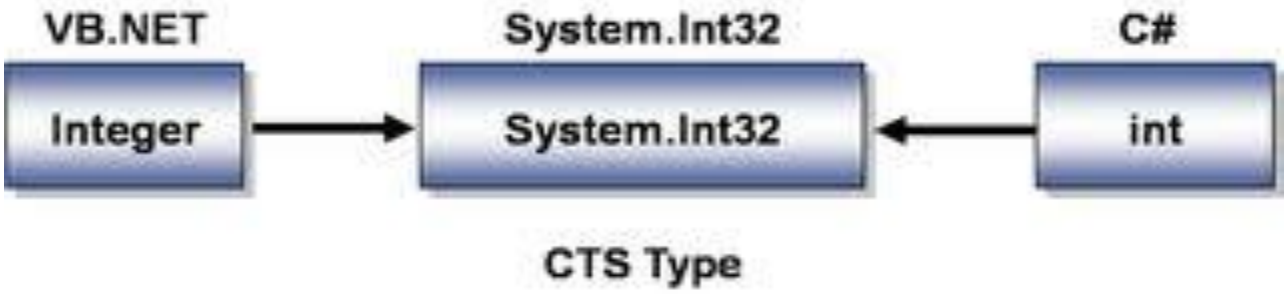

```
Employee a=new Employee();
a.age = 26;
Employee b=a;
b.age = 30
```

This will just create a new variable which will point to same location as “a” points.

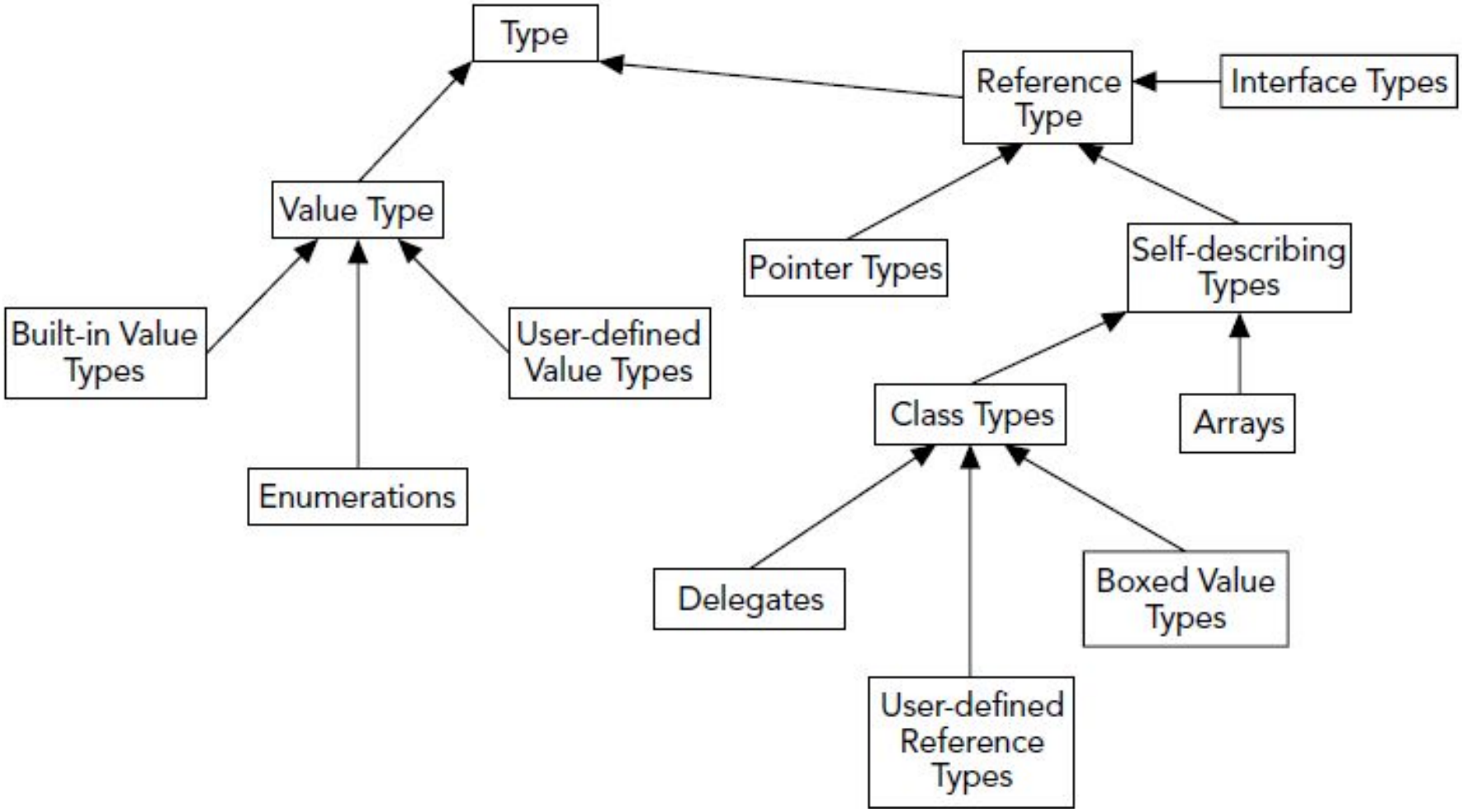
Общая система типов

CTS (Common Type System) - общая система типов в CLR. Стандарт, который описывает определение типов и их поведение.

- **CTS поддерживает только единичное наследование (в отличие от C++)**
- **Все типы наследуются от System.Object (Object – корень все остальных типов)**



Иерархия типов C#



Что такое переменная?

Переменная представляет именованное место в памяти для хранения порции данных

Переменная характеризуется:

- Имя (Name)
- Адрес (Address)
- Тип данных (Data type)
- Значение (Value)
- Область видимости (Scope)
- Время жизни (Lifetime)

Типы данных

byte

- System.Byte
- Целое беззнаковое число
- 1 байт
- 0 до 255

short

- System.Int16
- Целые числа (маленький диапазон)
- 2 байта
- -32 768 до 32 767

int

- System.Int32
- Целые числа
- 4 байта
- -2 147 483 648 до 2 147 483 647

long

- System.Int64
- Целые числа (большой диапазон)
- 8 байт
- -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807

float

- System.Single
- Числа с плавающей точкой
- 4 байта
- + / - 3,4 × 10 ^ 38

Типы данных

double

- System.Double
- Числа двойной точности (более точные) с плавающей точкой
- 8 байт
- + / -1,7 × 10 ^ 308

decimal

- System.Desimal
- Денежный значения
- 16 байт
- 28 значащих цифр

char

- System.Char
- Один символ
- 2 байта
- N/A

bool

- System.Bool
- Логический
- 1 байт
- true или false

Типы данных

string	object
• System.String • Последовательность символов • 2 байта на симол • N/A	• System.Object • Служит базовым классом для всех типов в мире .NET • Позволяет сохранять любой тип в объектной переменной

Объявление и присваивание переменных

Идентификатор может содержать только буквы, цифры и символы подчеркивания

Идентификатор должен начинаться с буквы или символа подчеркивания

Идентификатор не должен быть одним из ключевых слов, которые C# резервирует для собственного использования

При объявлении переменной для ее хранения должно быть зарезервировано место в памяти, размер которого определяется типом, поэтому при объявлении переменной необходимо указать тип хранимых данных

Объявление и присваивание переменных

```
DataType variableName;
// or
DataType variableName1, variableName2;
int variableName;
```

```
variableName = value;
```

После объявления переменной можно присвоить значение для его дальнейшего использования в приложении с помощью оператора присваивания

Тип выражения при присваивании должен соответствовать типу переменной

```
int numberOfEmployees;
numberOfEmployees = "Hello";
```

CTE

При объявлении переменной, пока ей не присвоено значение, она содержит случайное значение

Объявление и присваивание переменных

При объявлении переменных вместо указания явного типа данных можно использовать ключевое слово var

```
var price = 20;
```

Неявную типизацию можно использовать для любых типов, включая массивы, обобщенные типы и пользовательские специальные типы

Неявная типизация применима только для локальных переменных в контексте какого-то метода или свойства

CTE

```
class ThisWillNeverCompile
{
    private var mylnt = 10;
    public var MyMethod(var x, var y) { }
}
```

Объявление и присваивание переменных

Неявно типизированную локальную переменную можно возвращать вызывающему методу, при условии, что возвращаемый тип этого метода совпадает с типом, лежащим в основе определенных с помощью var данных

```
static int GetAnIntValue()
{
    var retVal = 9;
    return retVal;
}
```

Локальным переменным, объявленным с помощью ключевого слова var, не допускается присваивать в качестве начального значения null

```
var myObj = null;
```

Объявление и присваивание переменных

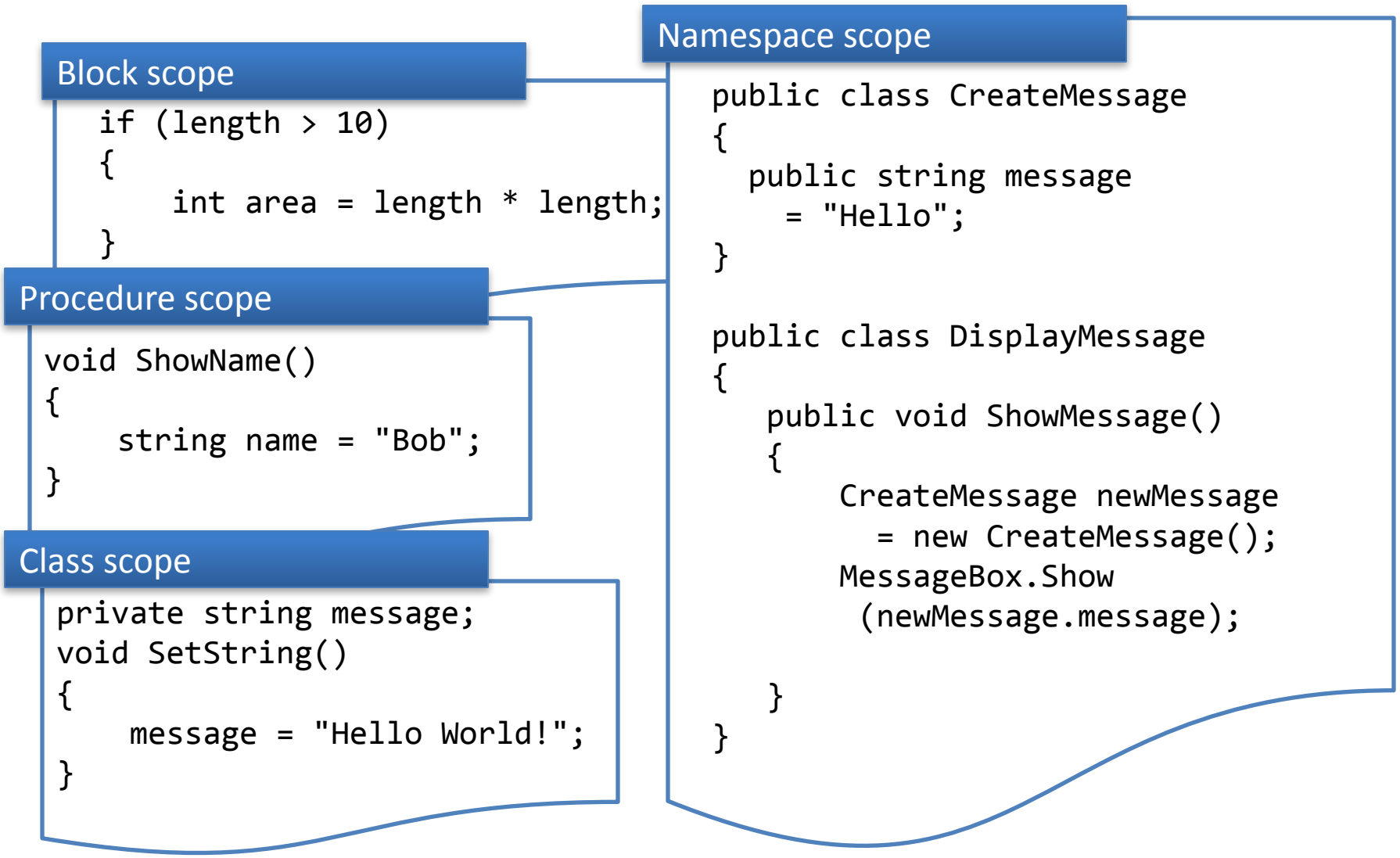
Значение неявно типизированной локальной переменной может быть присвоено другим переменным, причем как неявно, так и явно типизированным

```
var myObj = (int?)null;
myObj = 78;

var myCar = new Car();
myCar = null;
```

```
var myInt2 = 0;
var anotherInt = myInt;
string myString2 = "Wake up!";
var myData = myString;
```

Область видимости переменной



Преобразование типов данных

Неявное преобразование (implicit conversion)

Выполняется автоматически CLR согласно операциям, которые завершаются гарантированно успешно без потери информации

```
int a = 4;
long b;
b = a;
```

Неявное преобразование типа `int` в `long`

Преобразование типов данных

Явное преобразование (explicit conversion) или приведение (casting)

Требует, чтобы был написан код для выполнения преобразования, которое, в противном случае, может привести к потере информации или ошибке

```
DataType variableName1 = (castDataType)variableName2;
. . .
string possibleInt = "1234";
int count = Convert.ToInt32(possibleInt);
. . .
int number = 1234;
string numberString = number.ToString();
. . .
int number = 0;
string numberString = "1234";
if (int.TryParse(numberString, out number))
{// Conversion succeeded, number now equals 1234}
```

Константы и переменные только для чтения

Константы

Используются только для хранения неизменяемых данных

Объявляются с помощью ключевого слова const

Значение можно инициализировать только во время разработки

```
const DataType variableName = Value;
const double PI = 3.14159;
int radius = 5;
double area = PI * radius * radius;
double circumference = 2 * PI * radius;
```

Константы и переменные только для чтения

Переменные только для чтения (read-only)

Используются только для хранения неизменяемых данных

Объявляются с помощью ключевого слова readonly

Значение можно инициализировать во время выполнения

```
readonly DataType variableName = Value;
readonly string currentDateTime = DateTime.Now.ToString();
```

Неявно типизированные переменные (var keyword)

Ключевое слово var сообщает компилятору о необходимости определения типа переменной из выражения, находящегося с правой стороны оператора инициализации.

```
var numTypes = 13;

var values = new[] { 1, 2, 3 };
var person = new { FirstName = "Rod", LastName = "Stephens" };

var value1 = 100;  //int
var value2 = 1000000000; //int
var value3 = 10000000000; //long
var value4 = 10000000000000000000; //syntax error (value is too big)
var value5 = 1.23; //double
var value6 = new { Description= "Pencils", Quantity = 12, PriceEach =
0.25m }; //an object with the three fields: Description, Quantity,
      //and PriceEach.

                Console.WriteLine(person.GetType().Name);
```

Nullable-типы

Значение Null указывает на то, что значение неизвестно, или, другими словами, переменная не содержит никаких данных (значения нет).

```
int? count;
//or
Nullable<int> count;
//set count equal to null
count=null;
//set count equal to some value
count=1234;
//check if a variable has a value
if (count.HasValue) Console.WriteLine("count = " + count);
else Console.WriteLine("count = null");
//or
if (count != null) Console.WriteLine("count = " + count);
if (count == null) Console.WriteLine("count = null");
```

Спасибо за внимание