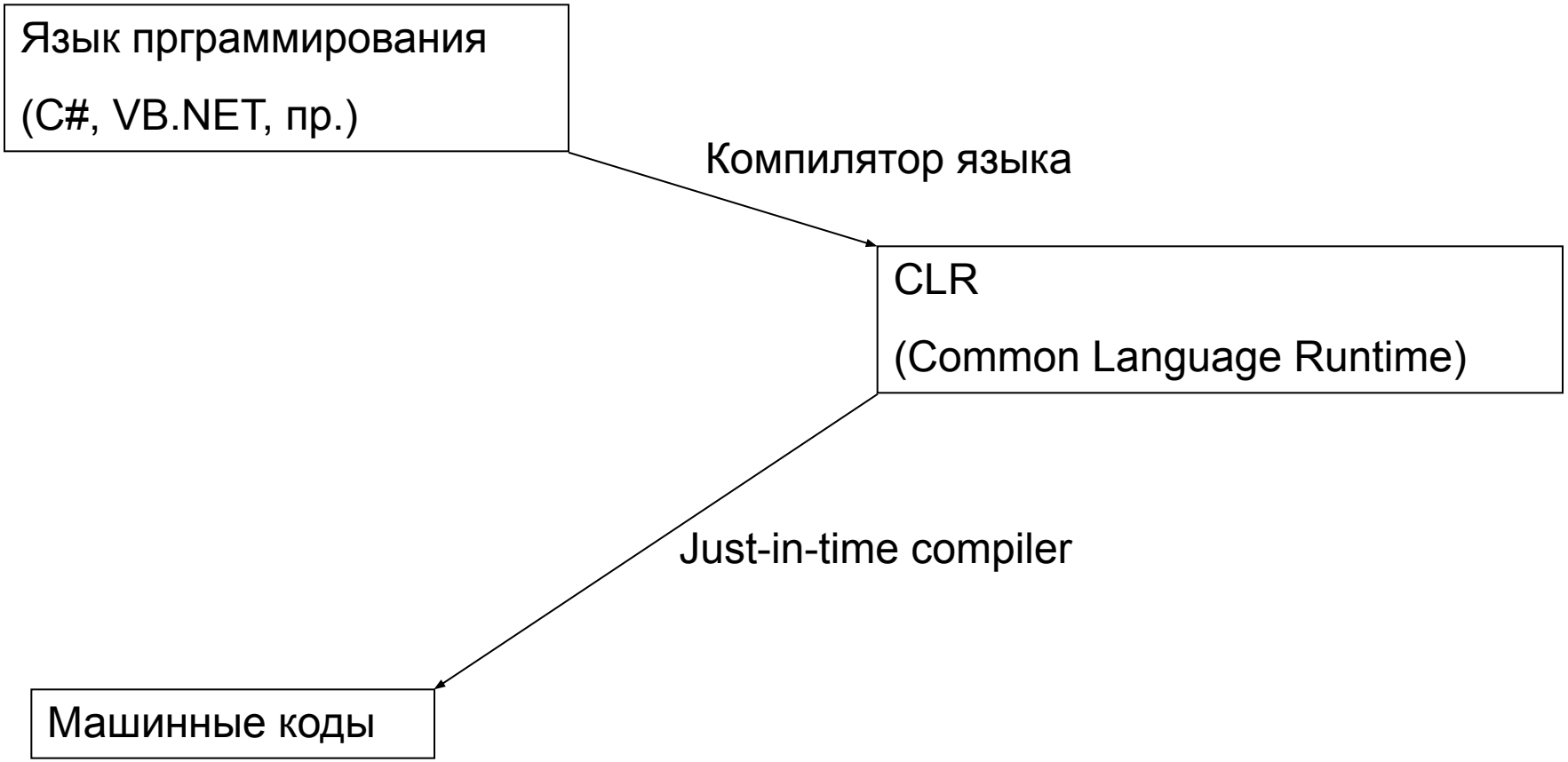
Список полезных источников

- http://msdn.microsoft.com
- http://www.rsdn.ru
- http://www.gotdotnet.ru
- http://www.gotdotnet.com
- http://www.firststeps.ru
- http://www.dotsite.spb.ru

- Интерфейс – класс, состоящий из определений функций.
- Пространство имен – именованная область определения переменных, типов, констант.

Архитектура .NET



Особенности .NET

- Платформонезависимость кода
- Управление памятью, повышенная надежность приложений
- Независимость от языка программирования
- Готовые библиотеки для различных видов разработок
- Простота связывания скомпилированных модулей
- Удаленный вызов библиотек
- Упрощение разработки
- Упрощение распространения приложений

Недостатки .NET

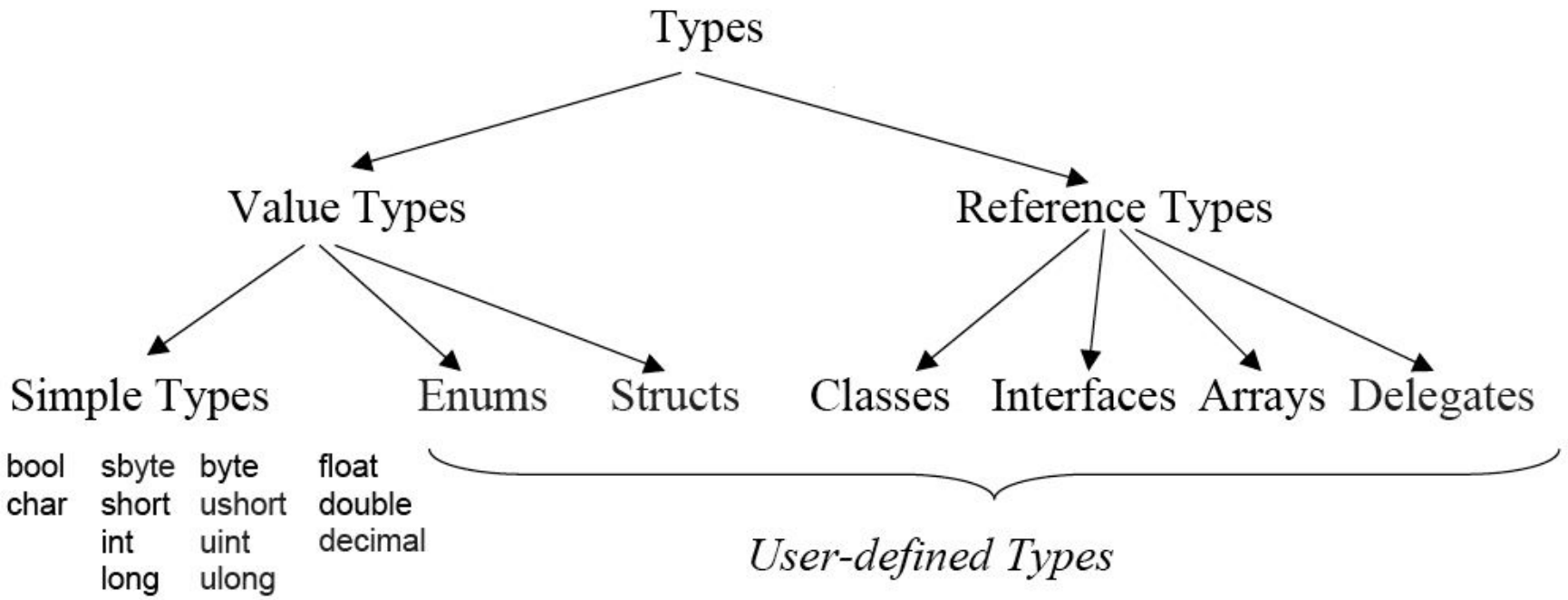
- Низкая производительность
- Требовательность к памяти
- Необходимость аккуратного проектирования для своевременного освобождения ресурсов (периферия, пр.)

.NET Framework class library

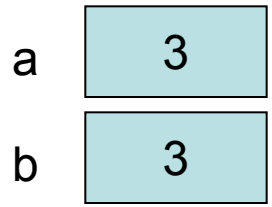
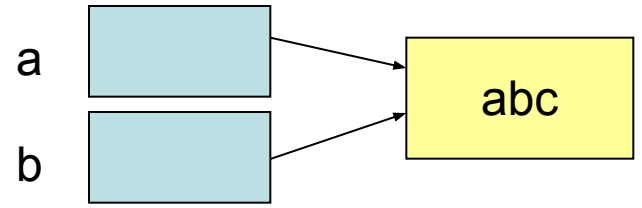
- System.Collections
- System.Data
- System.Drawing
- System.IO
- System.Windows.Forms

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_fxnetstart/html/50c4d770-0bb4-4e6a-bcf0-966bc7a3de77.htm

Типы данных


Types
Value Types
Reference Types
Simple Types
Enums
Structs
Classes
Interfaces
Arrays
Delegates
bool
char
sbyte
short
int
long
byte
ushort
uint
ulong
float
double
decimal
User-defined Types

Типы данных

	Value type	Reference type
Содержит	значение	Ссылку
Управление памятью	стек	куча
Присваивание	Копирует значение int a=3; int b =a; 	Копирует ссылку string a = “abc”; string b = a; 

Сравнение ключевых слов.

Объявление переменной.

Visual Basic

```
Dim x As Integer
Public x As Integer = 10
```

C++

```
 int x;
 int x = 10;
```

C#

```
 int x;
 int x = 10;
```

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Сравнение языков.

Комментарии.

Visual Basic

```
' comment
x = 1 ' comment
Rem comment
```

C++

```
// comment
x = 1; // comment
/* multiline
   comment */
```

C#

```
// comment
x = 1; // comment
/* multiline
   comment */
```

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Сравнение ключевых слов.

Создание экземпляра класса.

Visual Basic

```
Dim x As MyClass = New MyClass
Dim x As New MyClass
Dim x As New MyClass(100)
```

C++

```
MyClass *x = new MyClass();
MyClass *x = new MyClass(100);
```

C#

```
MyClass x = new MyClass();
MyClass x = new MyClass(100);
```

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Сравнение ключевых слов.

Объявление массива.

Задача	C++	C#	VB.NET
Объявление массива	Int a[5];	Int [ ]x = new int[5];	Dim x As integer() = new integer(5) Dim x(5) As integer
Инициализация массива	Int a[3] = {1,2,3};	Int [ ]x = new int[3] = {1,2,3};	Dim x() As integer = {1,2,3}
Переопределение массива.	-	-	ReDim [Preserve]

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Сравнение ключевых слов.

Задача	C++	C#	VB.NET
Объявление константы	const	Const readonly	Const
Создание экземпляра класса	new	new	New
Ссылка на экземпляр самого себя	this	this	Me
Значение ссылки, не ссылающейся на объект	null	null	Nothing
Проверка переменной на существование ссылки на объект	Obj == null	Obj == null	Obj is nothing

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Сравнение ключевых слов.

Модификаторы области видимости.

	C#	VB.NET
Видимый только внутри класса	Private	Private
Невидимый вне сборки	Internal	Friend
Видимый внутри класса и его потомков	Protected	Protected
Видимый внутри сборки и потомков класса	Protected internal	Protected friend
Видимый всем	Public	Public

Определение класса.

Visual Basic

[<модификатор области видимости>] class <имя класса>
    [inherits <базовый класс>]
    [implements <базовый интерфейс>[, <базовый интерфейс>…]]

    <определения класса>
End class

C#

[<модификатор области видимости>] class <имя класса> [: <базовый класс>, <базовый интерфейс>, <базовый интерфейс>…]
{
    <определения класса>
}

Пример определения класса.

Visual Basic

```
public class MyClass
    inherits MyBaseClass
    implements IMyInterface

    <определения класса>

End class
```

C#

```
public class <имя класса> MyClass : MyBaseClass, IMyInterface
{
    <определения класса>
}
```

Определение метода.

Visual Basic

<модификатор области видимости> [shared] sub <имя метода>
([{ ByRef | ByVal } <имя параметра> As <тип данных>][, прочие параметры])

    <определения метода>

End Sub

C#

<модификатор области видимости> [static] void <имя метода>
([{ ref | out }] <Тип данных> <имя параметра> [, прочие параметры])
{

    <определения метода>

}

Пример определения метода.

Visual Basic

```
Public shared sub Test(ByVal i as integer, ByRef c as char, ByRef q as integer())
 …
End sub

Dim I as integer : Dim c as char = “q” : Dim q as integer()
Test(I, c, q)
```

C#

```
Public static void Test(int i, ref char c, out int[] q)
{
        …
}

Int I; char c = ‘q’; int[] q;
Test(I, ref c, out q);
```

Определение метода, возвращающего значение.

Visual Basic

```
<модификатор области видимости> [shared] function <имя метода>
([параметры]) [As <тип данных>]

    <определения метода>

    return <значение>
End Function
```

C#

```
<модификатор области видимости> [static] <Тип данных> <имя метода>
([параметры])
{

    <определения метода>
    return <значение>;
}
```

Пример определения метода.

Visual Basic

```
Public shared sub Test(ByVal i as integer, ByRef c as char, ByRef q as integer())
 …
End sub

Dim I as integer : Dim c as char = “q” : Dim q as integer()
Test(I, c, q)
```

C#

```
Public static void Test(int i, ref char c, out int[] q)
{
        …
}

Int I; char c = ‘q’; int[] q;
Test(I, c, q);
```

Сравнение ключевых слов.

Задача	C++	C#	VB.NET
Переопределить метод	-	override	Overrides
Метод должен быть переопределен	Abstract	abstract	MustOverride
Не может быть переопределен	Sealed	sealed	NotOverridable
Метод, который может быть переопределен	Virtual	Virtual	Overridable
Спрятать метод	-	new	Shadowing

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Пример перегрузки метода.

Visual Basic

```
Public class A
        Public sub Test(ByVal i as integer, ByRef c as char)
End class

Public class B
        inherits A
        Public overloads sub Test(ByVal i as integer)
End class
```

C#

```
Public class A
{
        Public void Test(int i, ref char c)
}

Public class B : A
{
        public void Test(int i)
}
```

Пример переопределения метода.

Visual Basic

```
Public class A
        Public overridable sub Test(ByVal i as integer, ByRef c as char)
End class

Public class B
        inherits A
        Public overrides sub Test(ByVal i as integer, ByRef c as char)
End class
```

C#

```
Public class A
{
        Public virtual void Test(int i, ref char c)
}

Public class B : A
{
        public override void Test(int I, ref char c)
}
```

Пример переопределения метода.

Visual Basic

```
Public class A
        Public overridable sub Test(ByVal i as integer, ByRef c as char)
End class

Public class B
        inherits A
        Public overrides sub Test(ByVal i as integer, ByRef c as char)
End class
```

C#

```
Public class A
{
        Public virtual void Test(int i, ref char c)
}

Public class B : A
{
        public override void Test(int I, ref char c)
}
```

Определение свойства

Visual Basic

```
<модификатор области видимости> [readonly] property <Имя свойства> [As <тип данных>]
        Get
                …
                return <значение>
        End Get
        [Set(ByVal value As <тип данных>)

        End Set
End Property
```

C#

```
 <модификатор области видимости> [readonly] <тип данных> <Имя свойства>
   {
      get
      {
         ...
         rerturn <значение>;
      }
      [set
      {     …       }]
}
```

Пример определения свойства

Visual Basic

```
Dim _qwerty As Integer
    Public Property qwerty() As Integer
        Get
            Return _qwerty
        End Get
        Set(ByVal value As Integer)
            _qwerty = value
        End Set
    End Property
```

C#

```
int _qwerty;
    public int qwerty
    {
        get
        {
            return _qwerty;
        }
        set
        {
            _qwerty = value;
        }
    }
```

Сравнение управляющих конструкций.

C++	C#	VB.NET

If (<условие>)
…
[else
…]
	If (<условие>)
	…
	[else
	…]
		If <условие> then
		…
		[else
		…]
		End if

Switch (<переменая>)
{
case <значение>:
…
break;
default: …
}
	Switch (<переменая>)
	{
	case <значение>:
	…
	break;
	default: …
	}
		Select case <пер.>
		Case <значение>
		…
		
		case else
		…
		End select

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Сравнение управляющих конструкций.

C++	C#	VB.NET

While (условие)
{
…
}
	While (условие)
	{
	…
	}
		While условие
		
		…
		End while

do { … } while <условие>;	do { … } while <условие>;	do … loop while <условие>
For(<инициализация>;<условие> ;<итерация>) …;	For(<инициализация>;<условие> ;<итерация>) …;	For <счетчик> = <значение> to <значение> … Next
-	Foreach (<счетчик> in <коллекция>) …;	For each <счетчик> in <коллекция> … Next

ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_vsref/html/7c0efba1-45d9-4798-9051-9bdca3b899a6.htm

Делегаты

Делегат = тип метода

	C#	VB.NET
Объявление типа делегата.	Delegate Notifier (string sender);	Delegate sub Notifier (ByVal sender As string)
Создание переменной делегата.	Notifier note;	Dim note as Notifier

Связывание делегата с методом.	Void SayHello (string Sender)
	{
	Console.WriteLine(Sender);
	}
	
	note = new Notifier(SayHello);
		Sub SayHello (ByVal Sender As string)
		Console.WriteLine(Sender)
		End sub
		
		Note = new Notifier(AddressOf SayHello)

Вызов делегата	Note(“Hello, from Vasya”);	Note(“Hello, from Vasya”)

Делегаты

- Значение переменной делегата может быть нулевым.
- При нулевом значении вызов не может быть осуществлен.
- Обрабатываются как обычные объекты с данными.

Операторы приведения типов

VB:

CType(<объект>, <тип>) ‘при невозможности приведения произойдет исключение.

C#:

(<объект>)<тип> //при невозможности приведения произойдет исключение.

<объект> As <тип> //при невозможности приведения вернет null

MS Visual Studio

- Поддержка нескольких языков
- Подсветка синтаксиса
- Дизайнер форм, поддержка визуального наследования
- Отладка приложений
- Средства рефакторинга и навигации в коду
- Средства модульного тестирования