



Обзор платформы .NET

Бестужев Никита Евгеньевич
Преподаватель

Содержание лекции

1. Обзор .NET
2. Главные компоненты платформы .NET (CLR, CTS и CLS)
3. Различия между сборками, пространствами имен и типами

Обзор .NET

Базовые функциональные

ВОЗМОЖНОСТИ:

- Возможность обеспечения взаимодействия с существующим программным кодом
- Поддержка для многочисленных языков программирования
- Общий исполняющий механизм
- Полная и тотальная интеграция языков
- Обширная библиотека базовых классов
- Отсутствие необходимости в предоставлении низкоуровневых деталей COM
- Упрощенная модель развертывания

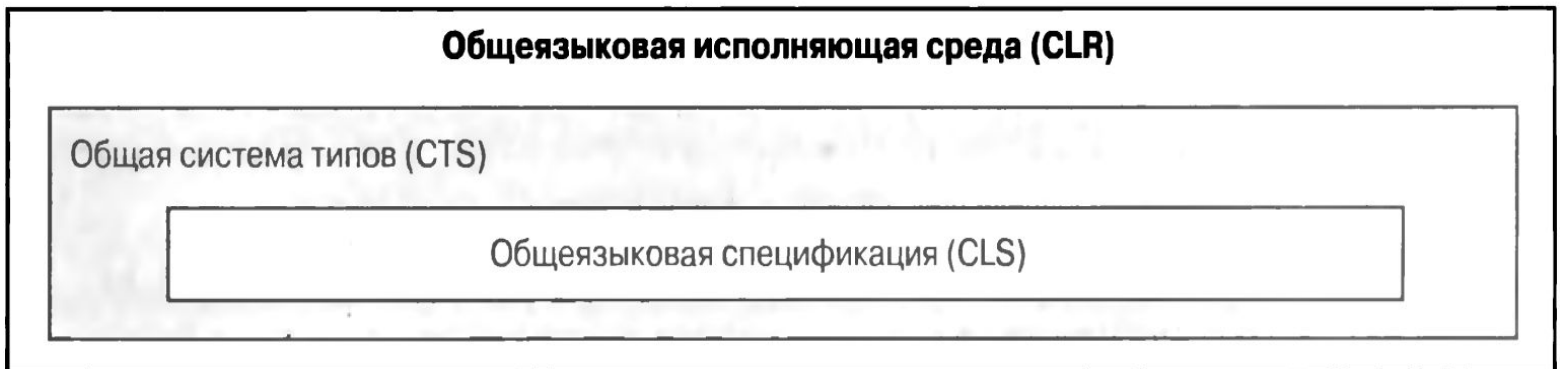
Главные компоненты платформы .NET (CLR, CTS и CLS)

Три ключевые сущности платформы:

- CLR – Common Language Runtime (общезыковая исполняющая среда)
- CTS – Common Type System (общая система типов)
- CLS – Common Language Specification (общезыковая спецификация)

Главные компоненты платформы .NET (CLR, CTS и CLS)

Роль библиотек базовых классов:



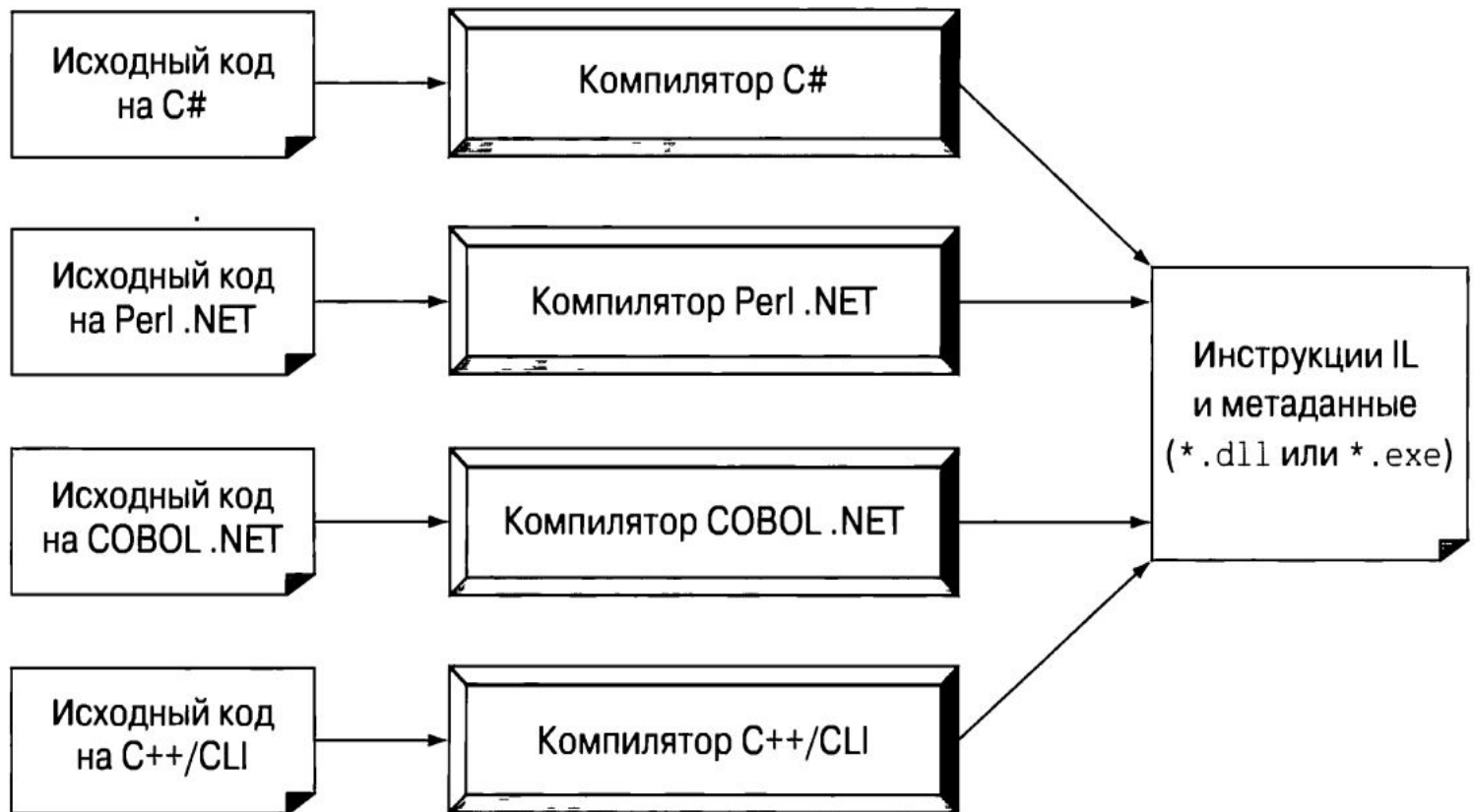
Главные компоненты платформы .NET (CLR, CTS и CLS)

Что привносит язык C#:

- Новый удобный синтаксис, похожий на Java и C
- Ряд функциональных возможностей: лямбда-выражения, анонимные типы, LINQ
- Только управляемый код

Главные компоненты платформы .NET (CLR, CTS и CLS)

Что собой представляют сборки в .NET:



Главные компоненты платформы .NET (CLR, CTS и CLS)

Что собой представляет общая система типов (CTS):

- Класс
- Интерфейс
- Структура
- Перечисление
- Делегат

Главные компоненты платформы .NET (CLR, CTS и CLS)

Типы классов:

```
class Calc  
{  
    public int Add(int x, int y)  
    {  
        return x + y;  
    }  
}
```

Главные компоненты платформы .NET (CLR, CTS и CLS)

Типы интерфейсов:

```
public interface IDraw  
{  
    void Draw();  
}
```

Главные компоненты платформы .NET (CLR, CTS и CLS)

Типы структур:

```
struct Point
```

```
{  
    public int xPos, yPos;  
    public Point (int x, int y) { xPos = x; yPos = y; }  
    public void PrintPosition ()  
    {  
        Console.WriteLine (" ({0}, {1})", xPos, yPos);  
    }  
}
```

Главные компоненты платформы .NET (CLR, CTS и CLS)

Типы перечислений:

```
public enum CharacterType  
{  
    Wizard = 100,  
    Fighter = 200,  
    Thief = 300  
}
```

Главные компоненты платформы .NET (CLR, CTS и CLS)

Типы делегатов:

```
public delegate int BinaryOp(int x, int y);
```

Главные компоненты платформы .NET (CLR, CTS и CLS)

Встроенные типы данных:

Тип данных в CTS	Ключевое слово в Visual Basic	Ключевое слово в C#	Ключевое слово в C++и CLI
System.ByteByte	Byte	byte	unsigned char
System.SByteSByte	SByte	sbyte	signed char
System.Int16	Short	short	short
System.Int32	Integer	int	int или long
System.Int64	Long	long	__int64
System.UInt16	UShort	ushort	unsigned short
System.UInt32	UInteger	uint	unsigned int или unsigned long
System.UInt64	ULong	ulong	unsigned __int64
System.SingleSingle	Single	float	float
System.DoubleDouble	Double	double	double
System.ObjectObject	Object	object	object^
System.CharChar	Char	char	wchar_t
System.StringString	String	String	String^
System.DecimalDecimal	Decimal	decimal	Decimal
System.BooleanBoolean	Boolean	bool	bool

Главные компоненты платформы .NET (CLR, CTS и CLS)

Определение типов данных:

// Определение числовых переменных в C#.

int i=0;

System.Int32 j = 0;

// Определение числовых переменных в VB.

Dim i As Integer = 0

Dim j As System.Int32 = 0

Главные компоненты платформы .NET (CLR, CTS и CLS)

Что собой представляет общеязыковая спецификация (CLS):

Набор правил, которые во всех подробностях описывают минимальный и полный комплект функциональных возможностей, которые должен обязательно поддерживать каждый отдельно взятый .NET-компилятор для того, чтобы генерировать такой программный код, который мог бы обслуживаться CLR и к которому в то же время могли бы единообразным образом получать доступ все языки, ориентированные на платформу .NET.

Главные компоненты платформы .NET (CLR, CTS и CLS)

Что собой представляет общезыковая
исполняющая среда (CLR):

Коллекция внешних служб, которые
требуются для выполнения
скомпилированной единицы
программного кода.

Различия между сборками, пространствами имен и типами

- Пространство имен (namespace) - группа связанных между собой с семантической точки зрения типов, которые содержатся в сборке
- Сборка (assembly) - большой двоичный объект, получаемый при создании файла *.dll или *.exe с помощью .NET-компилятора
- Тип - любого элемента из множества (класс, интерфейс, структура, перечисление, делегат)

Различия между сборками, пространствами имен и типами

«Hello World» на языке C#:

```
using System;  
public class MyApp  
{  
    static void Main()  
    {  
        Console.WriteLine ("Hi from C#");  
    }  
}
```

Различия между сборками, пространствами имен и типами

«Hello World» на языке VB:

Imports System

Public Module MyApp

Sub Main ()

Console.WriteLine ("Hi from VB")

End Sub

End Module

Различия между сборками, пространствами имен и типами

Получение доступа к пространствам имен программным образом:

- *using System;*
- *using System.Drawing;*
- *using System.Windows.Forms;*
- *using System.Data;*
- *using System.Data.SqlClient;*

Различия между сборками, пространствами имен и типами

Получение доступа к типам из пространств имен:

```
using System;  
class Program  
{  
    public void DisplayLogo ()  
    {  
        // Используем полностью уточненное имя.  
        System.Drawing.Bitmap companyLogo =  
            new System.Drawing.Bitmap(20, 20);  
    }  
}
```


Литература

- Эндрю Троелсен – «Язык программирования C# и платформа .NET» (старое издание есть в библиотеке)
- Павел Агуров – «C#. Сборник рецептов»
- <http://msdn.microsoft.com>



Спасибо за внимание!