Введение в платформу Microsoft .NET

ШАГ
компьютерная
АКАДЕМИЯ

Введение в платформу Microsoft .NET.

История и этапы развития технологий программирования.

Причины возникновения платформы Microsoft .NET.

Сравнительный анализ преимуществ и недостатков платформы Microsoft .NET.

Основные причины возникновения платформы Microsoft .NET

необходимость межплатформенной переносимости (возможность выполняться независимо от операционной системы и архитектуры вычислительной техники)

межъязыковая интеграция

необходимость создания среды исполнения программных решений (предоставление безопасного исполнения приложений, решение проблемы производительности ОС, посредством контроля за распределением ресурсов)

необходимость в упрощении процесса развёртывания и уменьшение вероятности конфликта версий

Сравнительный анализ преимуществ платформы Microsoft .NET.

Полное межязыковое взаимодействие
(возможность писать программы для .NET на различных языках программирования C++/CLI, Visual Basic.NET, JavaScript и д.р.)

Интегрированная среда времени выполнения приложений
(управляет выделением памяти, обеспечивает доступ к различного рода ресурсам)

Обширная библиотека базовых классов
(библиотека классов позволяет избегать сложностей, связанных с выполнением низкоуровневых обращений к API-интерфейсам, и предлагает согласованную объектную модель, используемую всеми поддерживающими .NET языками)

Архитектурная независимость приложений
(использование промежуточного кода, компиляция в исполнимый код при запуске приложения с учетом особенностей архитектуры ПЭВМ)

Упрощенная модель развертывания приложений
(нет необходимости регистрировать приложение в системном реестре, платформа .NET позволяет сосуществовать нескольким версиям одной и той же библиотеки)

Сравнительный анализ недостатков платформы Microsoft .NET.

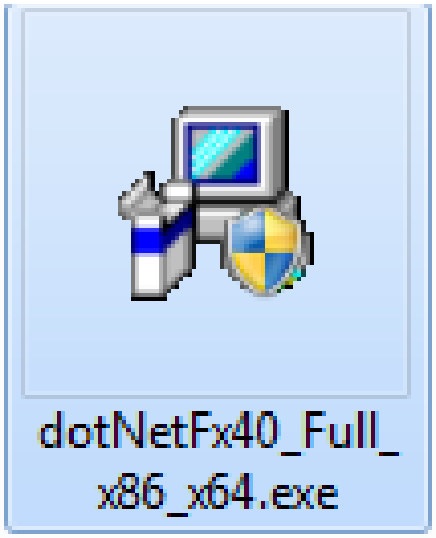
задержки при первом запуске приложения так, как компиляция происходит после запуска приложения

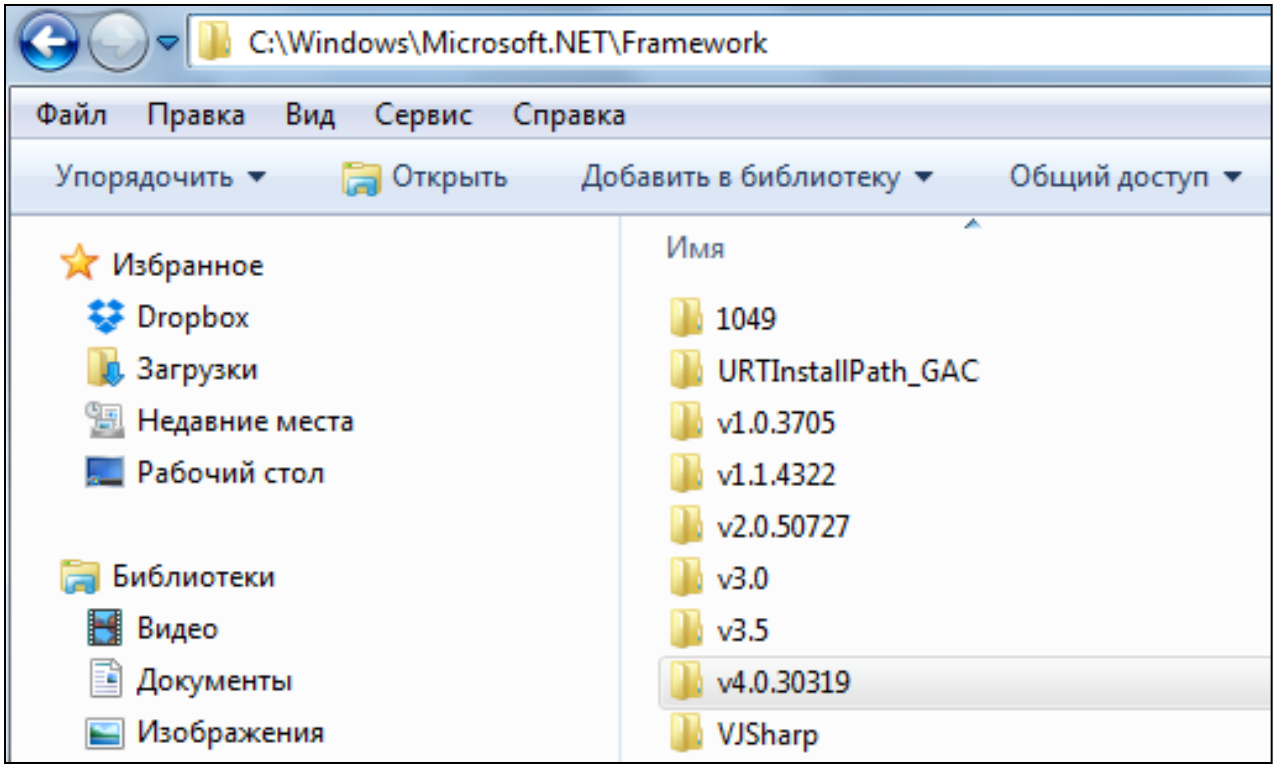
относительно медленное исполнение приложения вследствие того, что интегрированная среда исполнения забирает часть системных ресурсов на свои нужды.

управляемый код, не подвергшийся обфуксации, легко может быть декомпилирован в сравнении с естественным кодом, что может привести к потере коммерческой тайны.

межплатформенная непереносимость .NET-приложений при наличии в операционной системе .NET Framework, .NET-приложение должно корректно работать в любой операционной системе. Однако .NET Framework существует только для операционных систем Microsoft Windows.

Что такое .NET Framework?





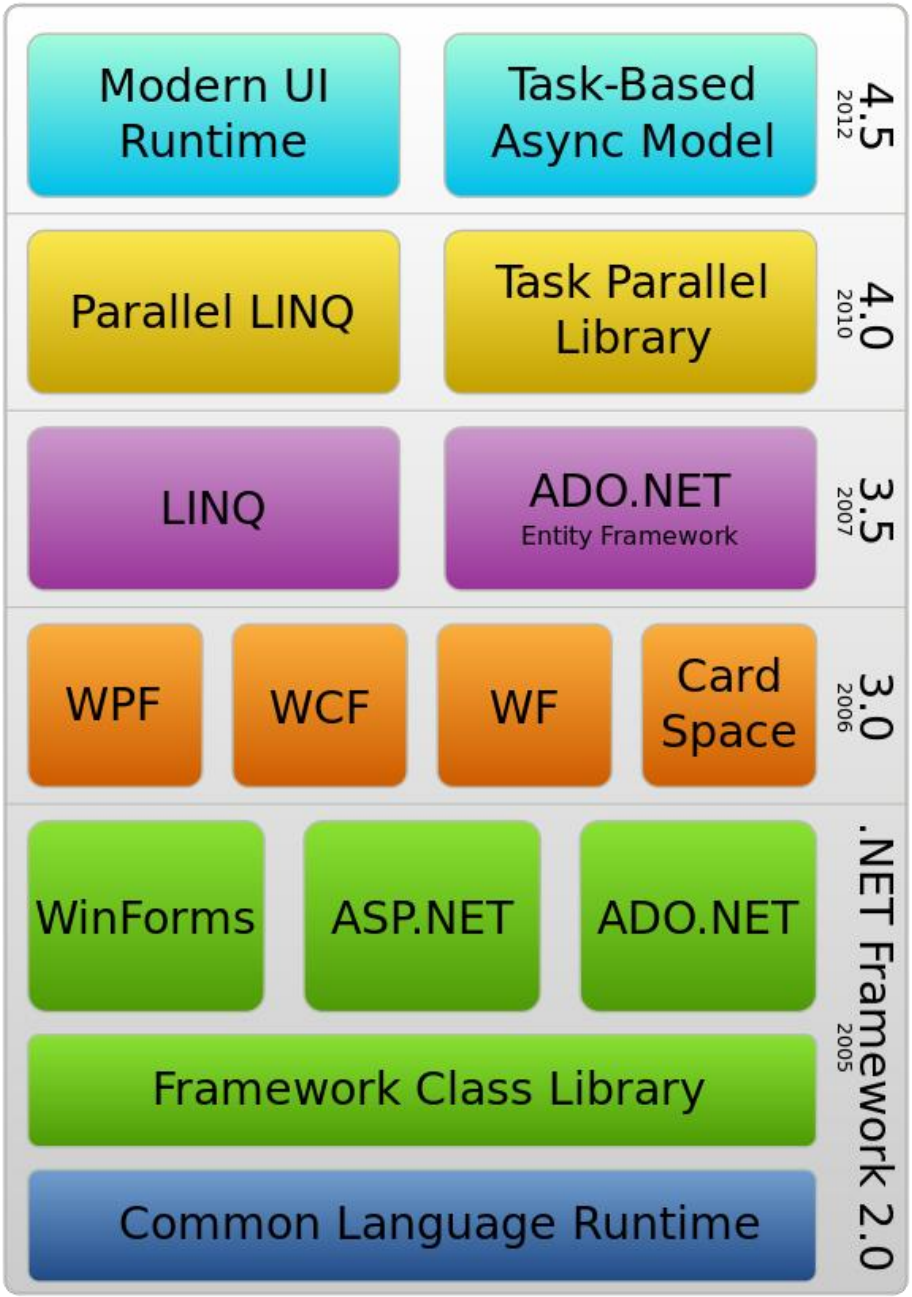
C:\Windows\Microsoft.NET\Framework

Список версий .NET Framework

Версия	Номер версии	Дата выхода	Visual Studio	По умолчанию в Windows
1.0	1.0.3705.0	1.05.2002	Visual Studio .NET	
1.1	1.1.4322.573	1 апреля 2003 года	Visual Studio .NET 2003	Windows Server 2003
2.0	2.0.50727.42	11 июля 2005 года	Visual Studio 2005	
3.0	3.0.4506.30	6 ноября 2006 года	Visual Studio 2005 + расширения	Windows Vista, Windows Server 2008
3.5	3.5.21022.8	9 ноября 2007 года	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	12 апреля 2010 года	Visual Studio 2010	
4.5	4.5.50709.17929	5 августа 2012 года	Visual Studio 2012	Windows 8, Windows Server 2012
4.5.1	4.5.50938.18408	17 октября 2013 года	Visual Studio 2013	Windows 8.1, Windows Server 2012 R2

Список версий .NET Framework

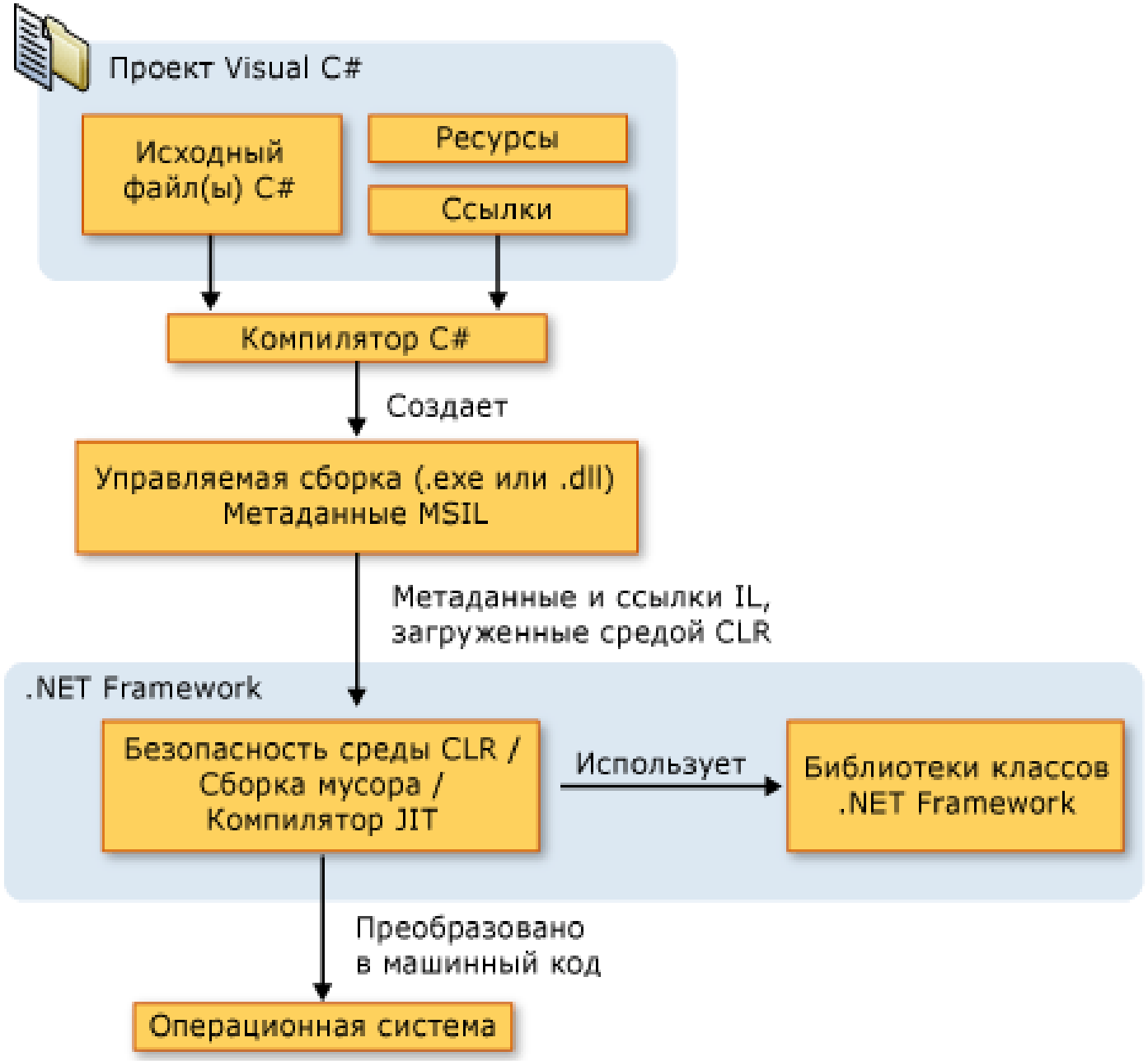
Версия	Номер версии	Дата выхода	Visual Studio	По умолчанию в Windows
4.5.2	4.5.51209.34209	5 мая 2014 года		Windows 10
4.6	4.6.1038.0	20 июля 2015 года	Visual Studio 2015	Windows 10
4.6.1	4.6.23123.0	17 ноября 2015 года	Visual Studio 2015 Update 1	Windows 10 Version 1511
4.6.2	2.0 — 4.6.2; Core 1.0	2017-03-07	Visual Studio 2017	10 v1607



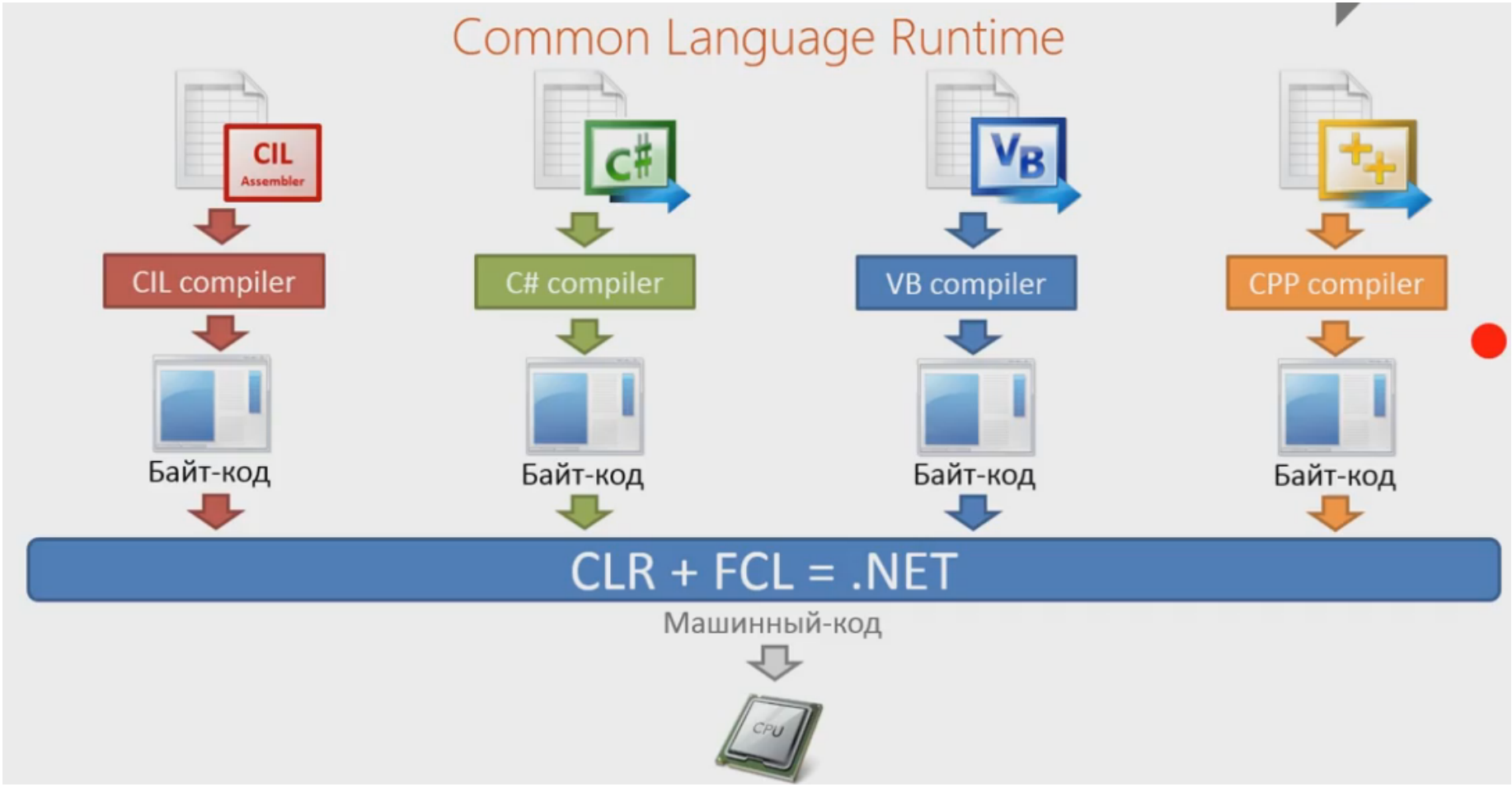
The .NET Framework Stack

Базовые понятия платформы Microsoft .NET

- Схема компиляции и исполнения приложения платформы Microsoft .NET.
- Языки платформы Microsoft .NET.
- Язык MSIL (Microsoft Intermediate Language).
- Понятия метаданных, манифеста, сборки.



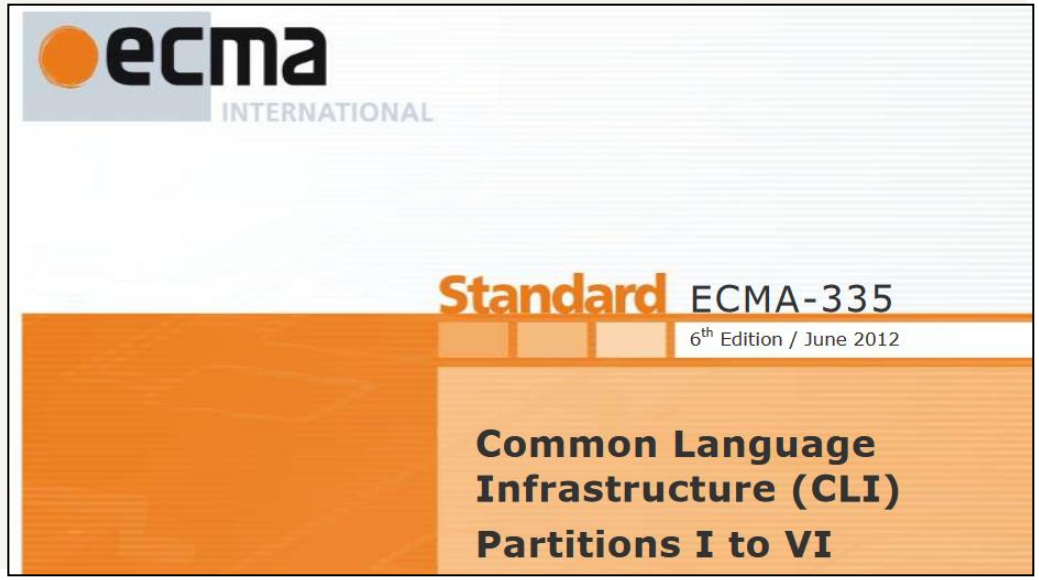
Упрощенная схема трансляции в .NET



Common Intermediate Language (сокращённо CIL) — «высокоуровневый ассемблер» виртуальной машины .NET. Промежуточный язык, разработанный фирмой «Microsoft» для платформы .NET Framework.

Ранее язык назывался «Microsoft Intermediate Language (MSIL)», однако был переименован создания стандарта «ECMA-335».

http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf



http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-335.pdf

Преимущества CIL

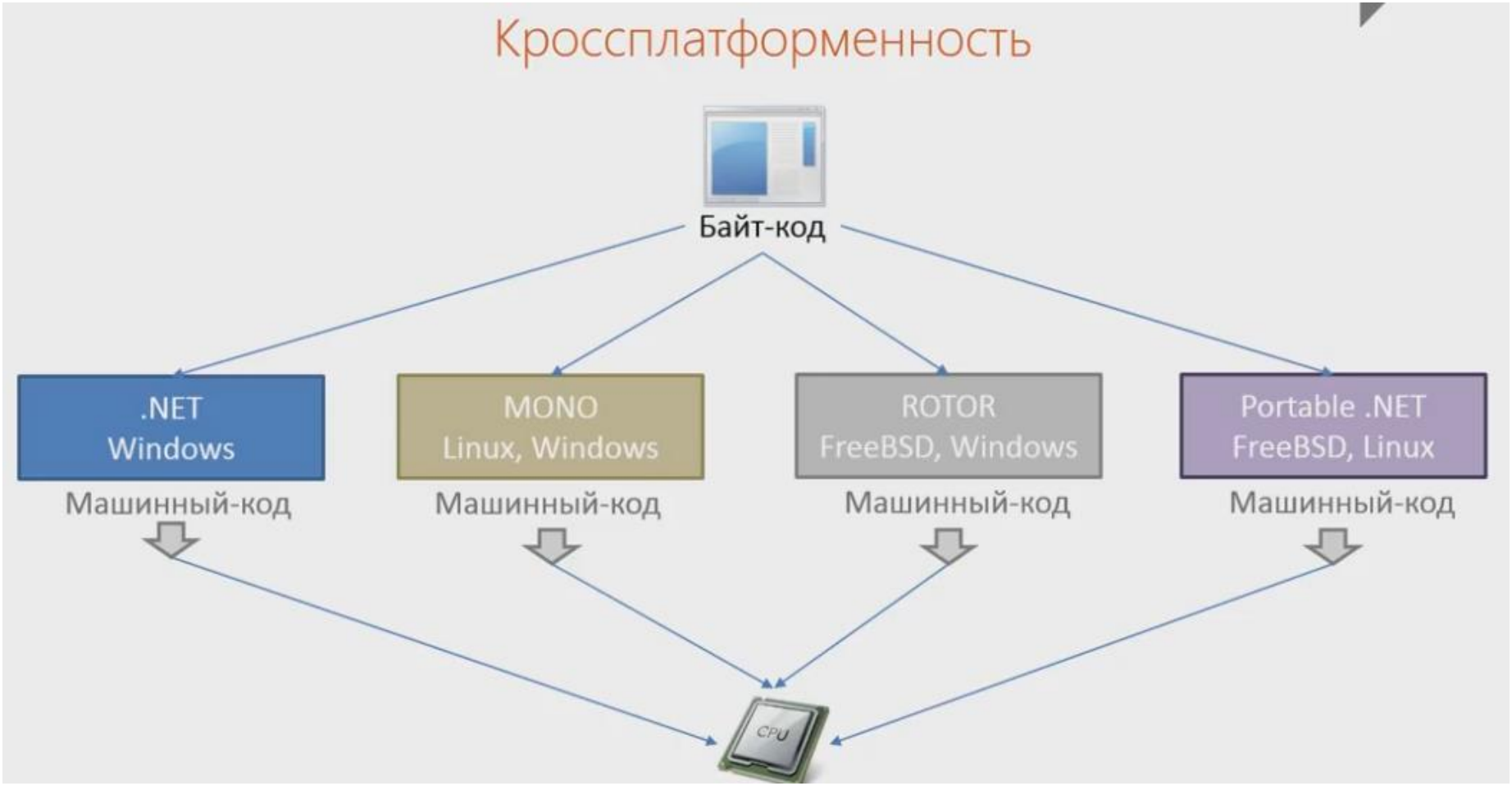
-независимость от программной платформы.

С помощью языков .NET можно создавать программы, которые будут работать под самыми разными операционными системами

- полное межъязыковое взаимодействие.

Любой код на любом языке программирования .NET (C#, VB. C++) компилируется в стандартный набор инструкций MSIL, проблем во взаимодействии между блоками кода MSIL не будет.

Кроссплатформенность



Исходный код программы на языке C#

```
using System;
namespace Culculator
{  class Program
    {
       public class Culc
        {  public Culc() { }
            public int Add(int a, int b)
            { return a + b;
            }
        }
        static void Main(string[] args)
        {
            Culc culculator = new Culc();
            int Summa=culculator.Add(5, 10);
            Console.WriteLine("Сумма равна {0}",Summa);
            Console.ReadLine();
        }
 }}
```

Исходный код программы на языке VB

```
Imports System
Module Module1

    Sub Main()
        Dim c As New Calc
        Dim ans As Integer = c.Add(5, 10)
        Console.WriteLine("Сумма равна {0}.", ans)
        Console.ReadLine()
    End Sub
End Module
Class Calc
    Public Function Add(ByVal x As Integer, ByVal y As Integer) As Integer
        Return x + y
    End Function
End Class
```

Culc::Add : int32(int32,int32)

Find Find Next

```
.method public hidebysig instance int32  Add(int32 a,
                                             int32 b) cil managed
{
  // Code size       9 (0x9)
  .maxstack  2
  .locals init ([0] int32 CS$1$0000)
  IL_0000:  nop
  IL_0001:  ldarg.1
  IL_0002:  ldarg.2
  IL_0003:  add
  IL_0004:  stloc.0
  IL_0005:  br.s       IL_0007
  IL_0007:  ldloc.0
  IL_0008:  ret
} // end of method Culc::Add
```

Код CIL программы на языке C#

CalculatorVB.Calc::Add : int32(int32,int32)

Find Find Next

```
.method public instance int32  Add(int32 x,
                                   int32 y) cil managed
{
  // Code size       9 (0x9)
  .maxstack  2
  .locals init ([0] int32 Add)
  IL_0000:  nop
  IL_0001:  ldarg.1
  IL_0002:  ldarg.2
  IL_0003:  add.ovf
  IL_0004:  stloc.0
  IL_0005:  br.s       IL_0007
  IL_0007:  ldloc.0
  IL_0008:  ret
} // end of method Calc::Add
```

Код CIL программы на языке VB

Двоичные файлы (.exe или .dll) для платформы .NET называются сборками (assembly).

Логическая группировка одного или нескольких **управляемых модулей**, содержащих MSIL, и файлов ресурсов.

Самая маленькая единица с точки зрения повторного использования, безопасности и управления версиями

Сборка состоит из следующих частей:

1. **Код на промежуточном языке CIL**. Данный код является независимым от операционной системы и типа процессора, на котором будет выполняться приложение.

2. **Метаданные** – специальное описание всех пользовательских типов данных (методов, свойств, событий, базовых классов, интерфейсов), размещенных в сборке.

3. **Манифест** (часть метаданных) – описание сборки: версия, ограничения безопасности, список внешних сборок и файлов, необходимых для работы данной сборки.

Назначение метаданных

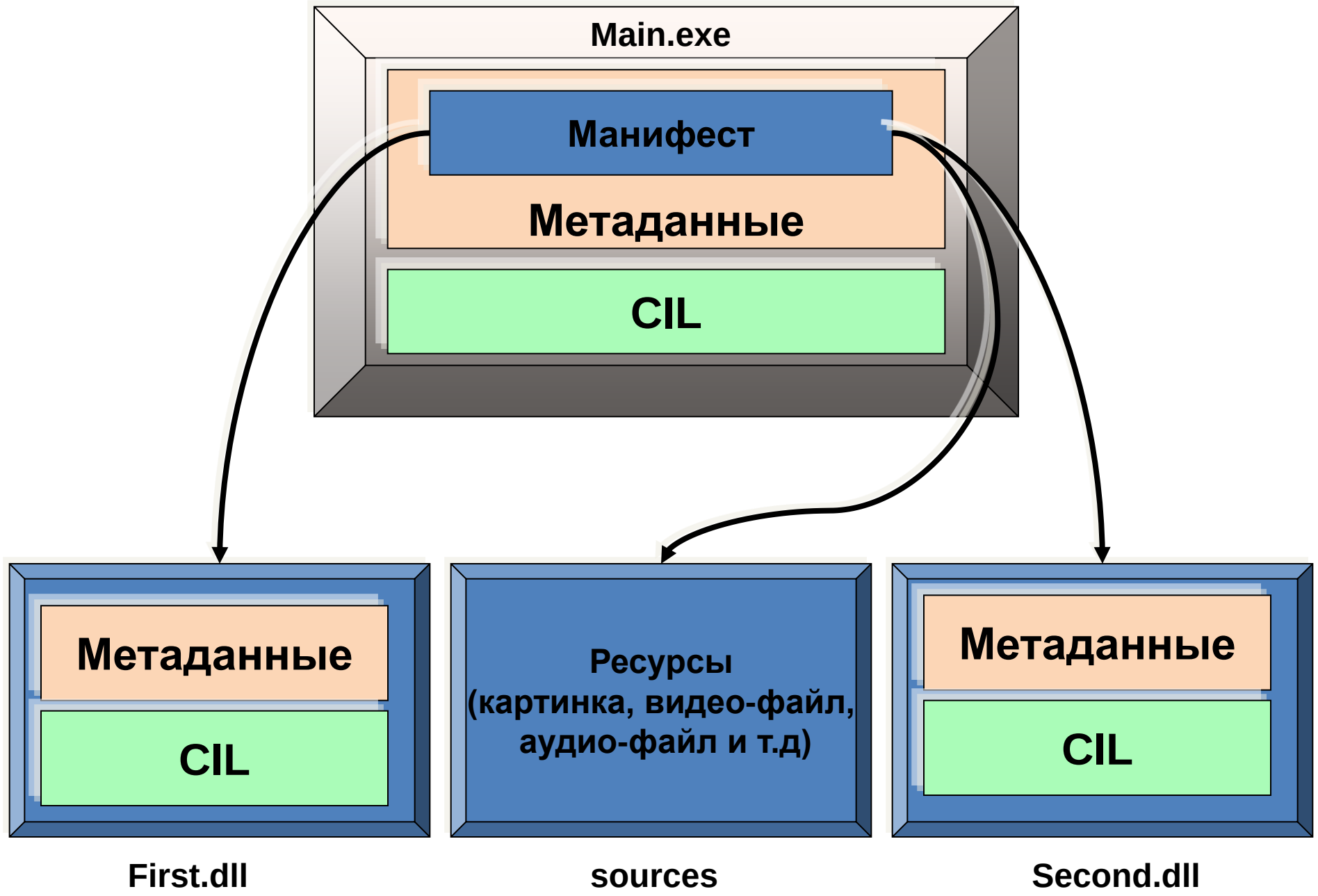
1. Метаданные устраняют необходимость в заголовочных и библиотечных файлах при компиляции, так как все сведения о типах/членах, на которые есть ссылки, содержатся в файле с MSIL кодом, в котором они реализованы.

2. Visual Studio .NET использует метаданные для облегчения написания кода. Ее функция **IntelliSense** анализирует метаданные и сообщает, какие методы, свойства, события и поля предлагает тип и какие параметры требуются методам.

3. В процессе верификации кода CLR использует метаданные, чтобы убедиться, что код совершает только «безопасные» операции.

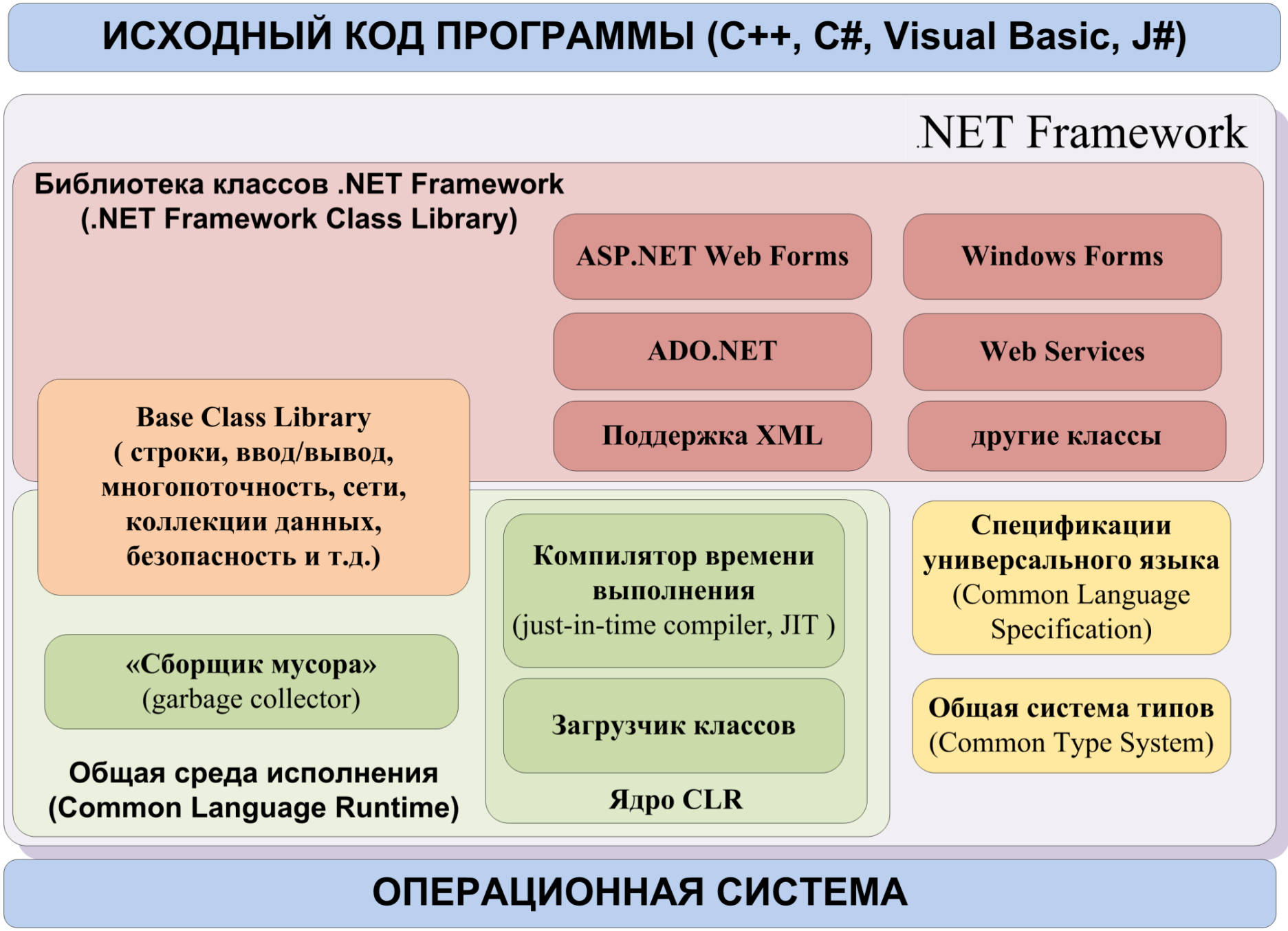
4. Метаданные позволяют **сериализовать** поля объекта в блок памяти, переслать данные на удаленную машину и затем **десериализовать**, восстановив объект и его состояние на этой машине.

5. Метаданные позволяют сборщику мусора отслеживать жизненный цикл объектов. Используя метаданные, сборщик мусора определяет тип объектов и узнает, какие поля в них ссылаются на другие объекты.


Main.exe
Манифест
Метаданные
CIL
Метаданные
CIL
Ресурсы
(картинка, видео-файл,
аудио-файл и т.д)
Метаданные
CIL
First.dll
sources
Second.dll

Базовые понятия платформы Microsoft .NET

- Архитектура платформы Microsoft .NET.
- Общеязыковая среда исполнения CLR (common language runtime).
- Стандартная система типов CTS (common type system).
- Стандартная языковая спецификация CLS (common language specification).
- Библиотека классов FCL (BCL).


ИСХОДНЫЙ КОД ПРОГРАММЫ (C++, C#, Visual Basic, J#)
.NET Framework
Библиотека классов .NET Framework
(.NET Framework Class Library)
ASP.NET Web Forms
Windows Forms
ADO.NET
Web Services
Поддержка XML
другие классы
Base Class Library
(строки, ввод/вывод,
многопоточность, сети,
коллекции данных,
безопасность и т.д.)
Компилятор времени
выполнения
(just-in-time compiler, JIT)
Спецификации
универсального языка
(Common Language
Specification)
«Сборщик мусора»
(garbage collector)
Загрузчик классов
Общая система типов
(Common Type System)
Общая среда исполнения
(Common Language Runtime)
Ядро CLR
ОПЕРАЦИОННАЯ СИСТЕМА

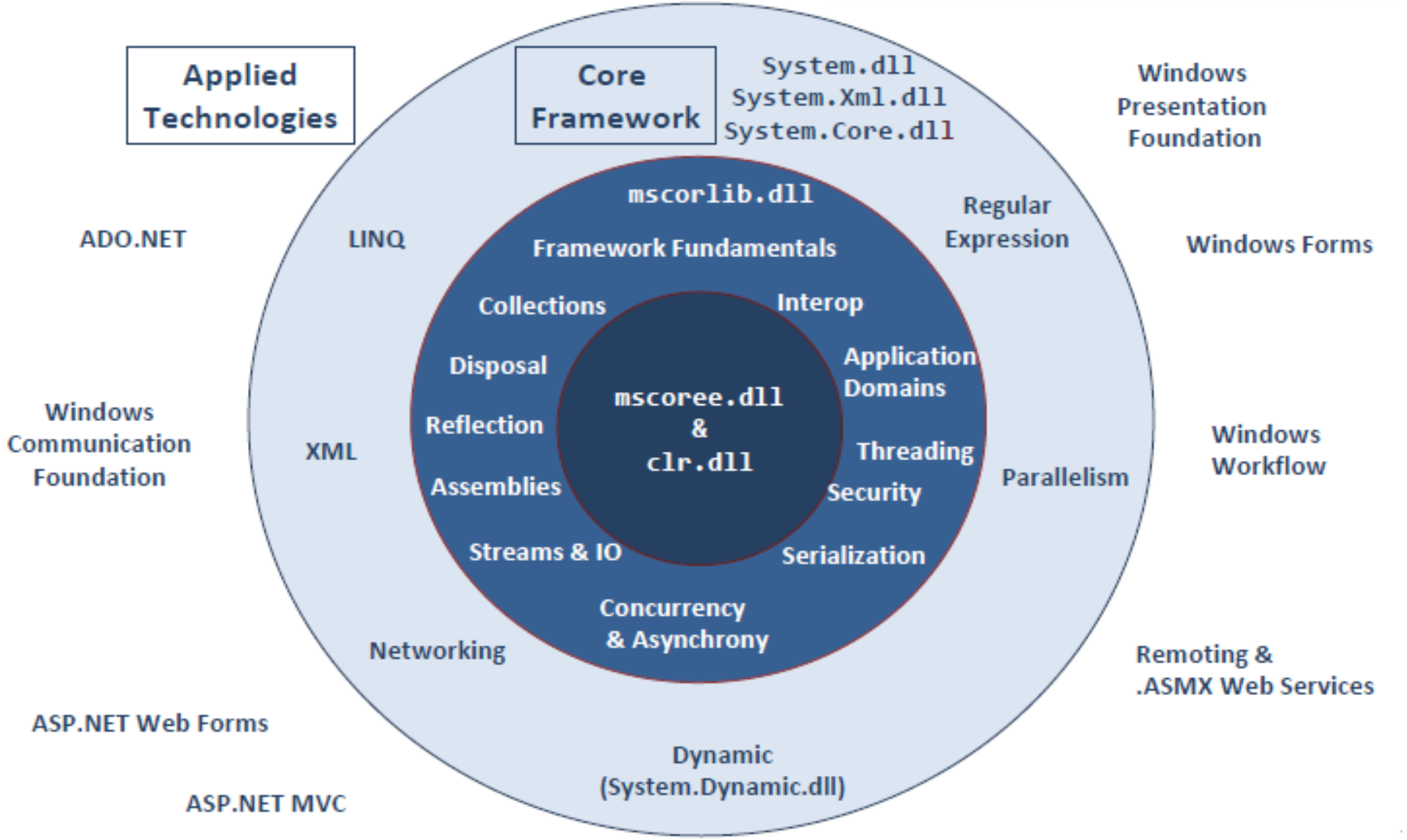
- **ядро среды выполнения CLR**, реализовано в виде библиотеки **mscoree.dll**.

- **библиотека базовых классов**. Библиотека разбита на множество отдельных сборок, главная сборка библиотеки базовых классов представлена файлом **mscorlib.dll.**

исходный код:
http://referencesource.microsoft.com/#mscorlib/mscorlib.csproj

Платформа .NET Framework

http://referencesource.microsoft.com/



Библиотека классов .NET (.NET FCL) предоставляет собой уже готовые к использованию типы (классы, интерфейсы), отображающие широкую предметную область.

Common Language Runtime (CLR) представляет собой систему, которая управляет выполнением пользовательских программ и делает программы переносимыми, поддерживает многоязыковое программирование и обеспечивает безопасность.

Common Language Specification (CLS) описывает набор свойств, которыми одновременно должны обладать различные языки.

Common Type System описывает все возможные типы данных и программные конструкции, поддерживаемые средой выполнения, и определяет, как одни типы данных могут взаимодействовать с другими.

Библиотека классов .NET (.NET FCL)

-содержит более 7 000 типов: классов, структур, интерфейсов, перечислений;

- некоторые классы FCL содержат до 100 методов, свойств и других членов,

- библиотека классов по смысловому содержанию типов .NET разделена на иерархические пространства имен;

-физически библиотека классов представляет собой набор DLL.

Исходный код: http://referencesource.microsoft.com/

Microsoft .NET Framework v2.0

Commonly Used Types and Namespaces

Microsoft Visual Studio 2005

windows.forms

System.Drawing
- Bitmap
- Brush
- Brushes
- Color
- Font
- FontFamily
- Graphics
- Icon
- Image
- ImageAnimator
- Pen
- Pens
- Region

System.Drawing.Design
- UITypeEditor

System.Drawing.Printing
- PageSettings
- PrintDocument
- PrinterSettings

System.Media NEW!
- SoundPlayer NEW!
- SystemSounds NEW!

System.Windows.Forms
- BindingSource NEW!
- Button
- CheckBox
- CheckedListBox
- ColorDialog
- ComboBox
- ContextMenuStrip NEW!
- DataGridView NEW!
- DateTimePicker
- FlowLayoutPanel NEW!
- FolderBrowserDialog
- FontDialog
- Form
- GroupBox
- ImageList
- Label A
- LinkLabel A
- ListBox
- ListView
- MaskedTextBox NEW!
- MenuStrip NEW!
- MessageBox
- MonthCalendar
- NumericUpDown
- OpenFileDialog
- PictureBox
- PrintDialog
- ProgressBar
- RadioButton
- RichTextBox
- SplitContainer NEW!
- StatusStrip NEW!
- TabControl
- TableLayoutPanel NEW!
- TextBox
- TextRenderer
- ToolStrip NEW!
- ToolStripContainer NEW!
- TrackBar
- TreeView
- UserControl
- WebBrowser NEW!

System.Windows.Forms.Design
- ControlDesigner
- DocumentDesigner
- UITypeEditor

System.Windows.Forms.VisualStyles
- VisualStyleRenderer NEW!

asp.net

System.Web
- HttpApplication
- HttpContext
- HttpCookie
- HttpRequest
- HttpResponse
- HttpRuntime
- HttpServerUtility
- HttpWorkerRequest
- IHttpHandler I
- IHttpModule I

System.Web.Caching
- Cache
- SqlCacheDependency NEW!

System.Web.Compilation
- BuildProvider NEW!
- ClientBuildManager NEW!

System.Web.Configuration
- WebConfigurationManager NEW!

System.Web.Hosting
- ApplicationManager NEW!
- HostingEnvironment NEW!
- VirtualPathProvider NEW!

System.Web.Management
- WebBaseEvent NEW!

System.Web.Profile NEW!
- ProfileBase NEW!
- SqlProfileProvider NEW!

System.Web.Security
- FormsAuthentication
- FormsAuthenticationTicket
- FormsIdentity
- Membership NEW!
- Roles NEW!

System.Web.Services
- WebMethodAttribute A
- WebService
- WebServiceAttribute A

System.Web.Services.Description
- ServiceDescription

System.Web.Services.Discovery
- DiscoveryClientProtocol

System.Web.Services.Protocols
- SoapDocumentMethodAttribute A
- SoapHttpClientProtocol

System.Web.SessionState
- HttpSessionState

System.Web.UI
- ClientScriptManager NEW!
- Control
- ICallbackEventHandler NEW! I
- IDataSource NEW! I
- Page
- MasterPage NEW!
- UserControl

System.Web.UI.HtmlControls
- HtmlControl
- HtmlForm
- HtmlButton
- HtmlInputControl

System.Web.UI.WebControls
- AccessDataSource NEW!
- Content NEW!
- ContentPlaceHolder NEW!
- CreateUserWizard NEW!
- DetailsView NEW!
- FormView NEW!
- GridView NEW!
- Login NEW!
- LoginName NEW!
- LoginStatus NEW!
- LoginView NEW!
- Menu NEW!
- ObjectDataSource NEW!
- SiteMapDataSource NEW!
- SiteMapPath NEW!
- SqlDataSource NEW!
- TreeView NEW!
- Wizard NEW!
- XmlDataSource NEW!

System.Web.UI.WebControls.WebParts NEW!
- SqlPersonalizationProvider NEW!
- WebPartManager NEW!
- WebPartZone NEW!
- WebPart NEW!
- WebPartConnection NEW!

tools

Microsoft.Build.BuildEngine NEW!
- Engine NEW!
- Project NEW!

Microsoft.Build.Framework NEW!
- ITaskItem NEW! I
- OutputAttribute NEW! A
- RequiredAttribute NEW! A
- ILogger NEW! I
- IEventSource NEW! I

Microsoft.Build.Utilities NEW!
- Logger NEW!
- Task NEW!
- TaskItem NEW!
- TaskLoggingHelper NEW!
- ToolTask NEW!

exceptions

- System.Exception
  - System.SystemException
    - BadImageFormatException
    - System.ArgumentException
      - System.ArgumentNullException
      - System.ArgumentOutOfRangeException
    - System.FormatException
      - System.IndexOutOfRangeException
    - System.InvalidOperationException
      - System.ObjectDisposedException
      - System.Net.NetworkInformation.PingException NEW!
      - System.Net.WebException
    - System.NotSupportedException
      - System.PlatformNotSupportException
    - System.OperationCanceledException NEW!
    - System.OutOfMemoryException
    - System.TimeoutException NEW!
    - System.UnauthorizedAccessException
    - System.Collections.Generic.KeyNotFoundException NEW!
    - System.Configuration.ConfigurationException
    - System.Data.DataException
      - System.Data.ConstraintException
      - System.Data.NoNullAllowedException
      - System.Data.ReadOnlyException
      - System.Data.RowNotInTableException
    - System.Drawing.Printing.InvalidPrinterException NEW!
    - System.EnterpriseServices.RegistrationException
    - System.IO.InvalidDataException NEW!
    - System.IO.IOException
      - System.IO.DirectoryNotFoundException
      - System.IO.FileNotFoundException
      - System.IO.FileLoadException
      - System.IO.PathTooLongException
    - System.Management.ManagementException
    - System.Runtime.InteropServices.InvalidComObjectException
    - System.Runtime.InteropServices.ExternalException
      - System.Data.Common.DbException
      - System.Data.Odbc.OdbcException
      - System.Data.OleDb.OleDbException
      - System.Data.OracleClient.OracleException
      - System.Data.SqlClient.SqlException
      - System.Web.HttpParseException
      - System.Web.HttpUnhandledException
    - System.ComponentModel.Win32Exception
      - System.Net.NetworkInformation.NetworkInformationException NEW!
      - System.Net.Sockets.SocketException
    - System.Runtime.Remoting.RemotingException
    - System.Runtime.Serialization.SerializationException
    - System.Security.SecurityException
    - System.Security.Authentication.AuthenticationException
    - System.Security.Cryptography.CryptographicException
    - System.Threading.ThreadAbortException
    - System.Transactions.TransactionException
      - System.Transactions.TransactionAbortedException
    - System.Xml.XmlException
      - System.Xml.Schema.XmlSchemaException

data and xml

System.Data
- DataColumn
- DataRelation
- DataRow
- DataSet
- DataTable
- DataView

System.Data.Common
- DbCommand
- DbCommandBuilder NEW!
- DbConnection
- DbConnectionStringBuilder
- DbDataAdapter
- DbDataReader NEW!
- DbProviderFactories NEW!
- DbProviderFactory NEW!

System.Data.Odbc
- OdbcCommand
- OdbcConnection
- OdbcConnectionStringBuilder NEW!
- OdbcDataAdapter
- OdbcDataReader
- OdbcParameter
- OdbcPermission
- OdbcTransaction

System.Data.OleDb
- OleDbCommand
- OleDbConnection
- OleDbConnectionStringBuilder NEW!
- OleDbDataAdapter
- OleDbDataReader
- OleDbParameter
- OleDbPermission
- OleDbTransaction

System.Data.OracleClient
- OracleCommand
- OracleConnection
- OradeConnectionStringBuilder NEW!
- OracleDataAdapter
- OracleDataReader
- OracleDataReader
- OraclePermission
- OracleTransaction

System.Data.Sql
- SqlNotification NEW!
- SqlDataSourceEnumerator NEW!

System.Data.SqlClient
- SqlBulkCopy NEW!
- SqlClientPermission
- SqlCommand
- SqlConnection
- SqlConnectionStringBuilder NEW!
- SqlDataAdapter
- SqlDataReader
- SqlDependency NEW!
- SqlParameter
- SqlTransaction

System.Messaging
- IMessageFormatter I
- Message
- MessageQueue

System.Xml
- XmlAttribute
- XmlDocument
- XmlElement
- XmlReader
- XmlReaderSettings NEW!
- XmlWriter
- XmlWriterSettings NEW!

System.Xml.Schema
- XmlSchemaSet NEW!

System.Xml.Serialization
- IXmlSerializable I
- XmlSerializer

System.Xml.Xpath
- XPathNavigator
- XPathExpression

System.Xml.Xsl
- XslCompiledTransform NEW!
- XsltArgumentList

fundamentals

System
- Action NEW! G
- AppDomain
- Array
- Attribute
- BitConverter
- Boolean
- Byte
- Char
- CLSCompliantAttribute A
- Comparison NEW! G
- Console
- ConsoleKeyInfo NEW!
- Convert
- Converter NEW! G
- DateTime
- Decimal
- Delegate
- Double
- Enum
- Environment
- EventArgs
- EventHandler
- EventHandler NEW! G
- Exception
- FlagsAttribute A
- IComparable I
- IComparable NEW! G I
- IConvertible I
- IDisposable I
- IEquatable NEW! G I
- IFormattable I
- Int32
- Int64
- Math
- Nullable NEW!
- Nullable NEW! G
- Object
- ObsoleteAttribute A
- Predicate NEW! G
- Random
- SerializableAttribute A
- String
- Type
- Uri
- UriBuilder
- UriParser NEW!
- Version

System.CodeDom
- CodeExpression
- CodeObject
- CodeStatement

System.CodeDom.Compiler
- CodeCompiler
- CodeGenerator
- CodeParser

System.Collections
- ArrayList
- Hashtable
- IComparer I
- IEnumerable I
- IList I
- SortedList

System.Collections.Generic NEW!
- Dictionary NEW! G
- EqualityComparer NEW! G
- ICollection NEW! G I
- IDictionary NEW! G I
- IEnumerable NEW! G I
- IEqualityComparer NEW! G I
- IList NEW! G I
- LinkedList NEW! G
- List NEW! G
- SortedList NEW! G

System.Collections.ObjectModel
- Collection NEW! G
- ReadOnlyCollection NEW! G

System.ComponentModel
- BackgroundWorker NEW!
- BindingList NEW! G
- Component
- TypeConverter
- TypeDescriptor

System.Configuration
- Configuration NEW!
- ConfigurationManager NEW!

System.Deployment.Application NEW!
- ApplicationDeployment NEW!

System.Diagnostics
- Debug
- EventLog
- PerformanceCounter
- Process
- Stopwatch NEW!
- Trace
- TraceListener
- TraceSource

System.DirectoryServices
- DirectoryEntry
- DirectorySearcher

System.DirectoryServices.ActiveDirectory NEW!
- ActiveDirectorySite NEW!
- DirectoryServer NEW!
- Domain NEW!
- Forest NEW!

System.DirectoryServices.Protocols
- DirectoryConnection NEW!
- DirectoryRequest NEW!
- DirectoryResponse NEW!

System.EnterpriseServices
- ContextUtil
- ServiceConfig
- ServicedComponent
- ServiceDomain

System.Globalization
- Calendar
- CompareInfo
- CultureAndRegionInfoBuilder NEW!
- CultureInfo
- DateTimeFormatInfo
- NumberFormatInfo
- RegionInfo
- TextInfo

System.IO
- Directory
- DriveInfo NEW!
- DriveType NEW!
- File
- FileStream
- Path
- Stream
- StreamReader
- StreamWriter
- UnmanagedMemoryStream NEW!

System.IO.Compression NEW!
- DeflateStream NEW!
- GZipStream NEW!

System.IO.IsolatedStorage
- IsolatedStorage

System.IO.Ports NEW!
- SerialPort NEW!

System.Net
- Dns
- FileWebRequest
- FtpWebRequest NEW!
- HttpListener NEW!
- HttpWebRequest
- WebClient

System.Net.Mail NEW!
- MailMessage NEW!
- SmtpClient NEW!

System.Net.NetworkInformation NEW!
- NetworkChange NEW!
- NetworkInterface NEW!
- Ping NEW!

System.Net.Security NEW!
- NegotiateStream NEW!
- SslStream NEW!

System.Net.Sockets
- NetworkStream
- Socket
- TcpClient
- TcpListener
- UdpClient

System.Reflection
- Assembly
- ConstructorInfo
- EventInfo
- FieldInfo
- GenericParameterAttributes NEW!
- MemberInfo
- MethodBase
- MethodInfo
- ParameterInfo
- PropertyInfo

System.Reflection.Emit
- AssemblyBuilder
- ConstructorBuilder
- EnumBuilder
- EventBuilder
- FieldBuilder
- MethodBuilder
- ParameterBuilder
- PropertyBuilder
- DynamicMethod NEW!
- TypeBuilder

System.Resources
- ResourceManager
- UltimateResourceFallbackLocation NEW!

System.Runtime.CompilerServices
- InternalsVisibleToAttribute NEW! A

System.Runtime.InteropServices
- ComVisibleAttribute A
- SafeHandle NEW!

System.Runtime.Serialization
- ISerializable I
- OptionalFieldAttribute NEW! A
- OnDeserializingAttribute NEW! A
- OnDeserializedAttribute NEW! A
- OnSerializingAttribute NEW! A
- OnSerializedAttribute NEW! A

System.Runtime.Serialization.Formatters.Binary
- BinaryFormatter

System.Security
- SecurityManager
- SecureString NEW!

System.Security.AccessControl NEW!
- AccessRule NEW!
- FileSecurity NEW!
- FileSystemAccessRule NEW!
- ObjectSecurity NEW!
- RegistryAccessRule NEW!
- RegistrySecurity NEW!

System.Security.Cryptography
- SHA1
- MD5
- TripleDES
- Rijndael

System.Security.Cryptography.Pkcs NEW!
- EnvelopedPkcs7 NEW!
- SignedPkcs7 NEW!

System.Security.Cryptography.X509Certificates
- X509Chain NEW!
- X509Store NEW!

System.Security.Permissions
- SecurityPermission

System.Security.Principal
- NTAccount NEW!
- SecurityIdentifier NEW!
- WindowsIdentity

System.ServiceProcess
- ServiceBase

System.Text
- Encoding
- StringBuilder
- UnicodeEncoding
- UTF8Encoding

System.Text.RegularExpressions
- Match
- Regex

System.Threading
- EventWaitHandle NEW!
- Interlocked
- Monitor
- Mutex
- ReaderWriterLock
- Semaphore NEW!
- Thread
- WaitHandle

System.Transactions NEW!
- TransactionScope NEW!
- CommittableTransaction NEW!
- Transaction NEW!

key NEW! New for .NET Framework 2.0 · A attribute · G indicates generic types · Generic type syntax is language specific. C#/J#: public class List<T> Visual Basic: Public Class List(Of T) C++: generic<typename T> public ref class List · I interface

Microsoft .net Framework

http://msdn.microsoft.com/netframework/

Microsoft

Система CLR осуществляет управление жизненным циклом и выполнением кода приложения. Такой код называется управляемым кодом (***managed code***)

Во время работы программы среда CLR следит за тем, чтобы выполнялись только разрешенные операции, осуществляет распределение и очистку памяти и обрабатывает возникающие ошибки. Это многократно повышает безопасность и надежность программ.

Достоинства:

современные методы управления памятью;

возможность использовать различные языки;

улучшенная безопасность;

поддержка управления версиями и четкая организация взаимодействия программных компонентов.

Недостатки:

- наличие специального компилятора, который должен создавать MSIL-файл, предназначенный для работы под управлением CLR-системы,
- использование компилятором библиотек среды .NET Framework

Сборщик мусора (garbage collector) является элементов CLR и предназначен для очистки памяти.

Основная идея:

- вся динамически выделяемая память (для приложений .NET CLR поддерживает собственную управляемую кучу) распределяется в области кучи;

- при обнаружении .NET, что управляемая куча данного процесса заполнилась и нуждается в приведении в порядок, он вызывает **сборщик мусора**;

- сборщик мусора проходит по всем переменным, находящимся в контексте нашего кода, и проверяет **ссылки на объекты**, расположенные в области кучи, чтобы идентифицировать, какие из них доступны из кода, т.е. узнать на какие объекты существуют ссылки;

- любой объект с отсутствием ссылок будет удален так, как к нему не будет осуществляться доступ из кода.

Common Type System (CTS)

Common Type System (*CTS*) - общая система типов, *полностью описывает все доступные для использования типы данных и конструкции языка, которые поддерживаются CLR*

Понятие типа существует 5 типов – это класс, структура, перечисление, интерфейс и делегат.

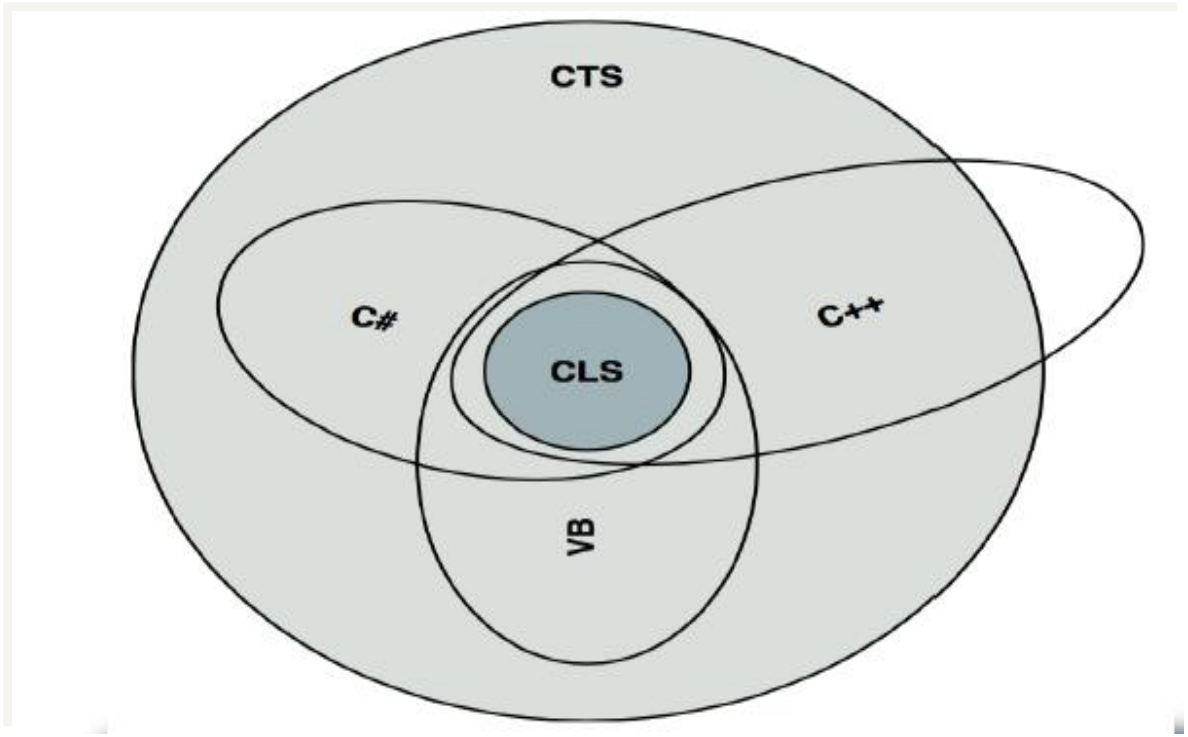
Члены типа (конструктор, финализатор, статический конструктор, вложенный тип, метод, свойство, индексатор, поле, поле только для чтения, константа, событие)

Механизм обмена сообщений основан на делегатах – функциональном типе (в C# встроен механизм событий, полностью согласованный с возможностями CLR)

C# поддерживает так же и **перегрузку операций.**

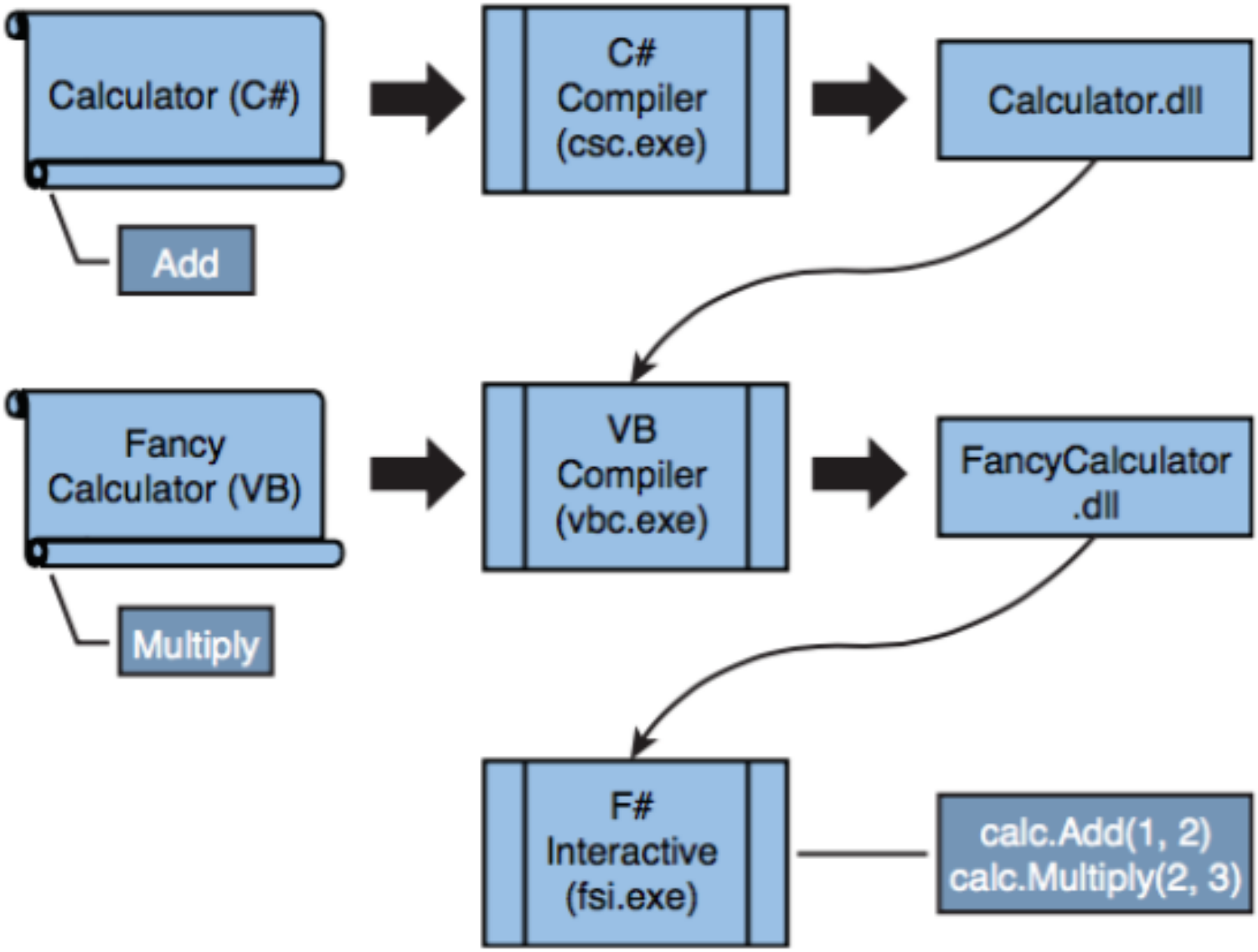
CTS устанавливает четко определенный набор фундаментальных типов данных.

Тип данных CTS	Ключевое слово VB	Ключевое слово C#	Ключевое слово C++/CLI
System.Byte	Byte	byte	unsigned char
System.SByte	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.Single	Single	float	float
System.Double	Double	double	double
System.Object	Object	object	object^
System.Char	Char	char	wchar_t
System.String	String	string	String^
System.Decimal	Decimal	decimal	Decimal
System.Boolean	Boolean	bool	bool



Спецификация CLS - это набор правил, подробно описывающих минимальное и полное множество характеристик, которые должен поддерживать отдельный компилятор .NET, чтобы генерировать программный код, обслуживаемый средой CLR.

Правила CLS применяются только к тем частям типа, которые видны извне определяющей сборки


Calculator (C#)
Add
C#
Compiler
(csc.exe)
Calculator.dll
Fancy
Calculator (VB)
Multiply
VB
Compiler
(vbc.exe)
FancyCalculator
.dll
F#
Interactive
(fsi.exe)
calc.Add(1, 2)
calc.Multiply(2, 3)

Компиляция IL в платформенно-зависимые инструкции

Компилятор времени выполнения (just-in-time compiler, JIT) предназначен для перевода CIL в платформенно-зависимые инструкции и входит в состав CLR.

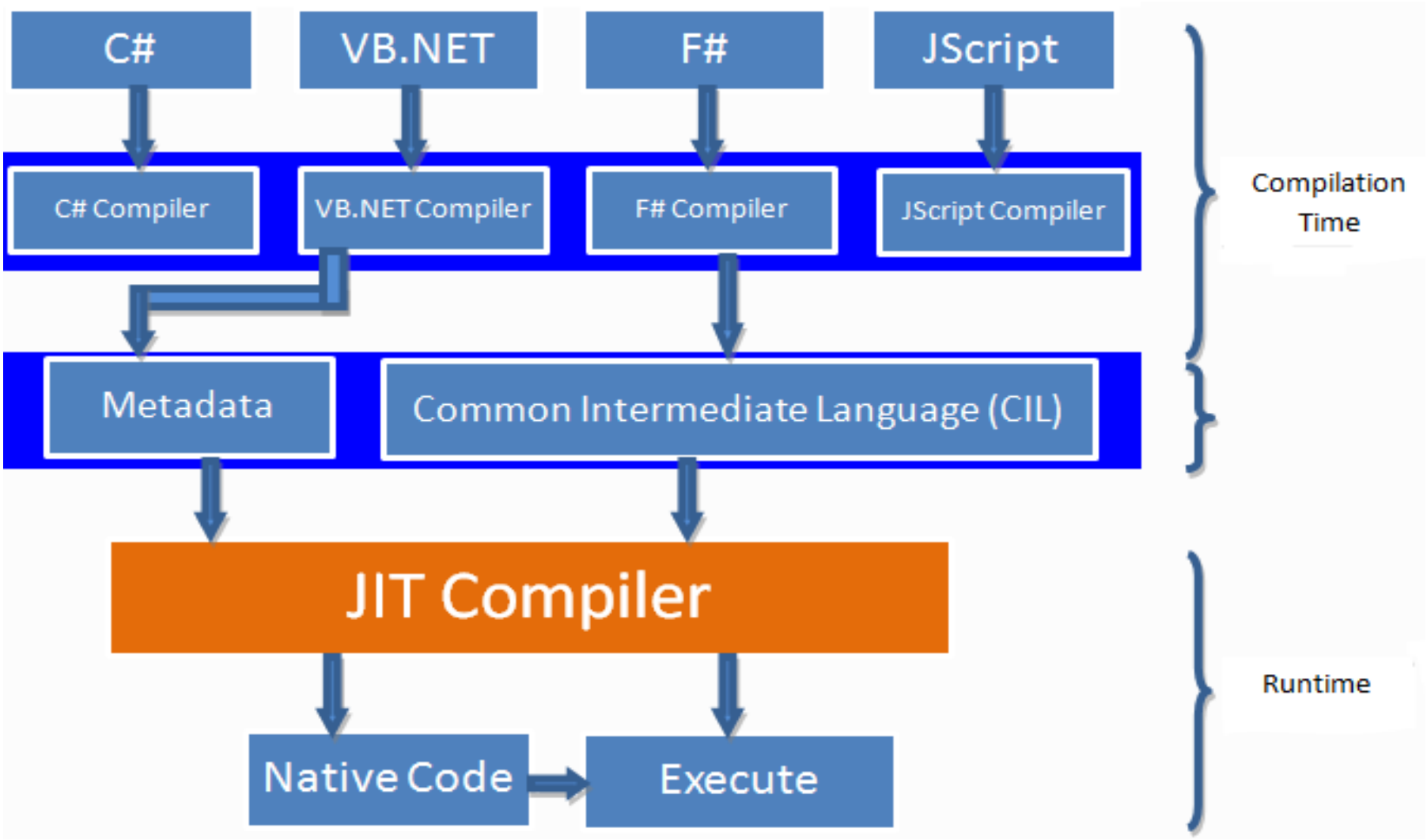
Особенности:

- В процессе выполнения программы компилируются только те ее части, которые требуется выполнить в данный момент
- Используя код CIL, разработчики могут не думать об особенностях архитектуры CPU данного компьютера - эти особенности будут учтены JIT.
- Откомпилированные из MSIL платформенно-зависимые инструкции JIT помещает в кэш-памяти, что очень сильно ускоряет работу приложения. При **ПОВТОРНЫХ** вызовах этого метода JIT уже не будет заниматься компиляцией, а просто возьмет уже готовый откомпилированный код из КЭШа в оперативной памяти.

Ядро CLR производит следующие действия:

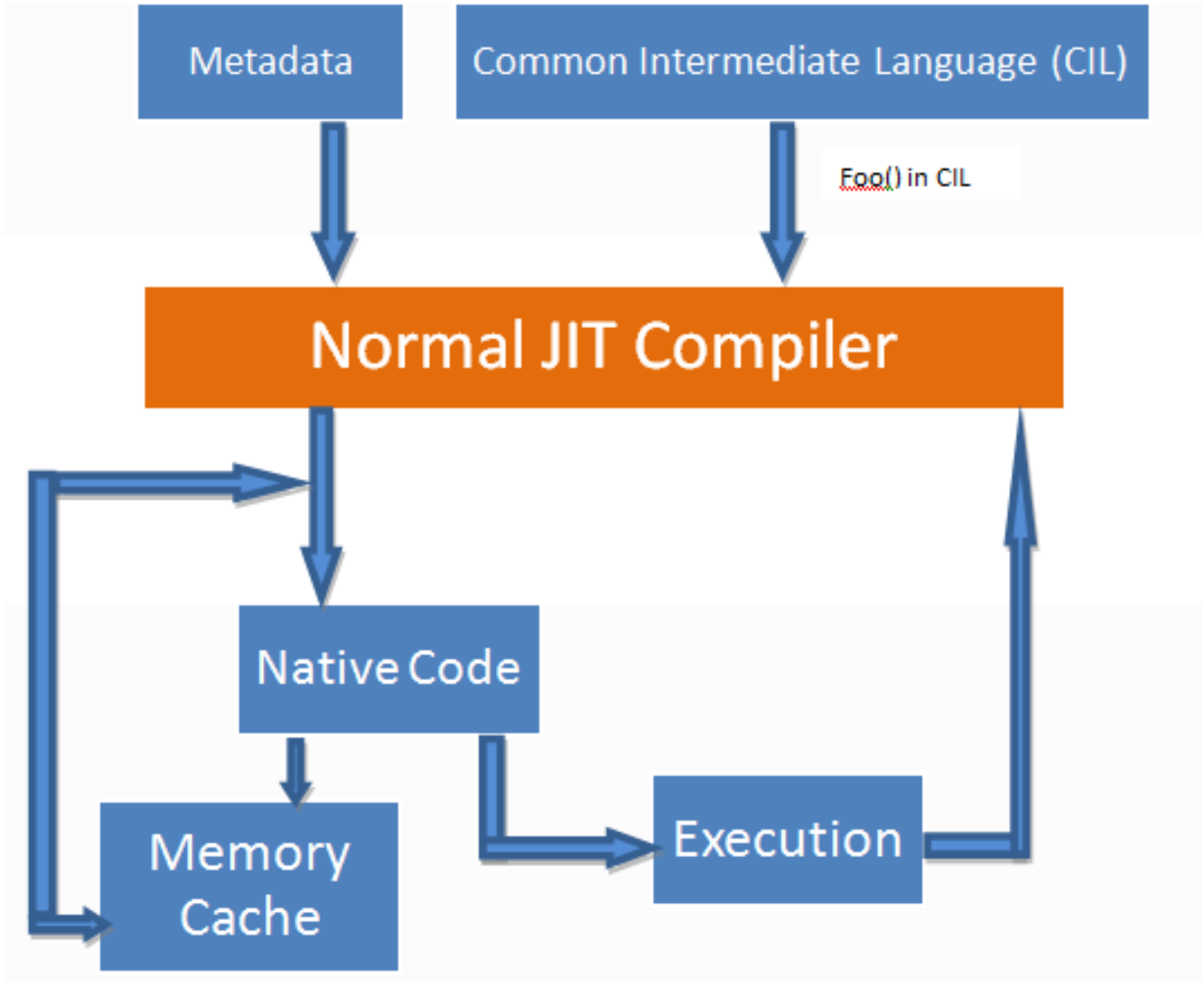
- загружает сборку в память;
- производит анализ содержимого сборки (выявляет классы, структуры, интерфейсы);
- производит анализ метаданных;
- обеспечивает компиляцию кода на промежуточном языке (MSIL) в платформозависимые инструкции (ассемблерный код);
- выполняет проверки, связанные с обеспечением безопасности;
- используя основной поток приложения, передает управление преобразованному в команды процессора фрагменту кода сборки.

JIT-компиляция

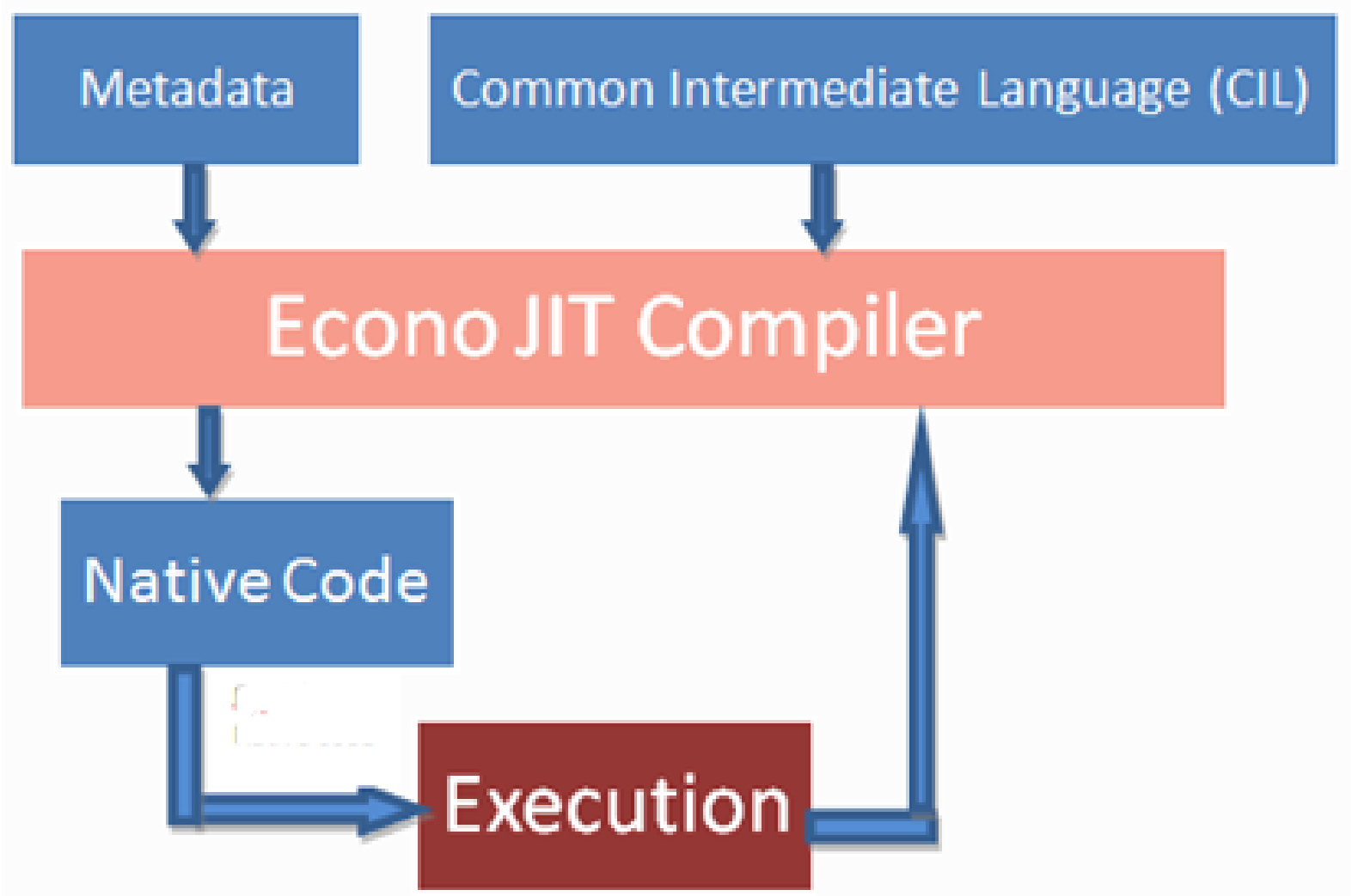


http://www.techstrikers.com/DotNet/dotnet-just-in-time-compiler.php

Виды компиляции. Normal JIT-компиляция

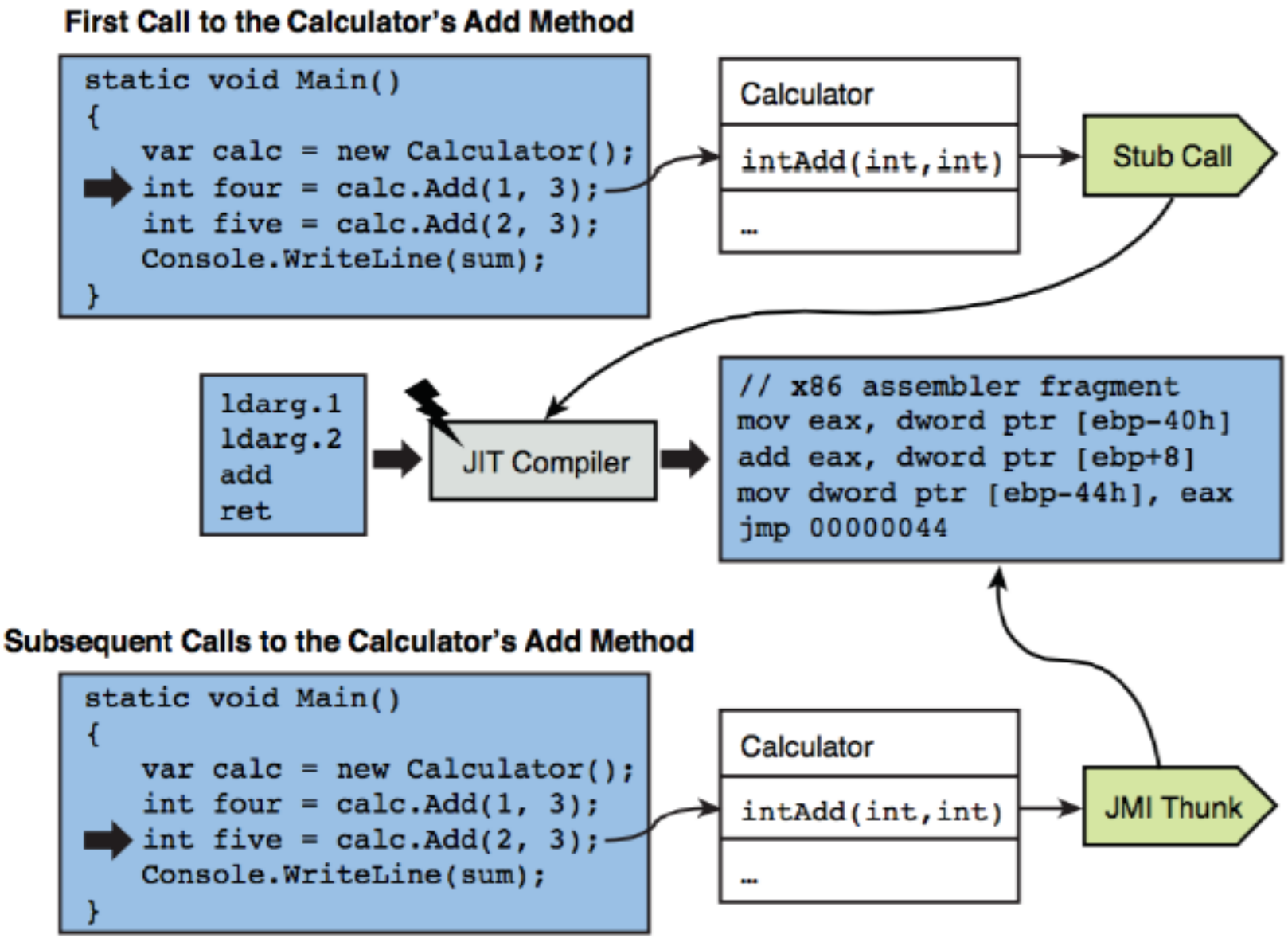


Виды компиляции. Econo JIT-компиляция



Особенности: не использует сохранение в Cache

Как CLR загружает, компилирует и запускает сборки



Работа с ILDASM.EXE

C:\Program Files\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools

Для просмотра метаданных типов, которые содержатся в загруженной в текущий момент сборке, необходимо нажать комбинацию клавиш <Ctrl+M>

Рефлекторы и дотфускаторы.

Что такое рефлектор?

Необходимость использования рефлектора.

Обзор существующих рефлекторов.

Что такое дотфускатор?

Необходимость использования дотфускаторов.

Обзор существующих дотфускаторов.

Обфускация или запутывание кода - приведение исходного текста или исполняемого кода программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

Обзор обфускаторов для .NET
http://habrahabr.ru/post/97062/