

# Module #1: Introduction to .NET

## 1) Задание 1 «Hello world console application»

```

MANIFEST
Find Find Next
// Metadata version: v4.0.30319
.assembly extern mscorlib
{
    .publickeytoken = ( 87 78 5C 56 19 34 E0 89 ) // .NET 4.0...
    .ver 4:0:0:0
}
.assembly task1
{
    .custom instance void [mscorlib]System.Runtime.CompilerServices.CompilationRelaxationsAttribute::ctor(int32) = ( 01 00 08 00 00 00 00 00 )
    .custom instance void [mscorlib]System.Runtime.CompilerServices.RuntimeCompatibilityAttribute::ctor() = ( 01 00 01 00 54 02 16 57 72 61 70 4E 6F 6E 45 78 // ....T.WrapNonEx
        63 65 70 74 69 6F 6E 54 68 72 6F 77 73 01 ) // ceptionThroush.

    // --- The following custom attribute is added automatically, do not uncomment -----
    // .custom instance void [mscorlib]System.Diagnostics.DebuggableAttribute::ctor(valueType [mscorlib]System.Diagnostics.DebuggableAttribute/DebuggingModes) = ( 01 00 07 01 00 00 00 00 )

    .custom instance void [mscorlib]System.Reflection.AssemblyTitleAttribute::ctor(string) = ( 01 00 05 74 61 73 68 31 00 00 ) // ...task1..
    .custom instance void [mscorlib]System.Reflection.AssemblyDescriptionAttribute::ctor(string) = ( 01 00 00 00 00 00 )
    .custom instance void [mscorlib]System.Reflection.AssemblyConfigurationAttribute::ctor(string) = ( 01 00 00 00 00 00 )
    .custom instance void [mscorlib]System.Reflection.AssemblyCompanyAttribute::ctor(string) = ( 01 00 0C 45 50 41 4D 20 53 79 73 74 65 6D 73 00 // ...EPAM Systems.
        00 )
    .custom instance void [mscorlib]System.Reflection.AssemblyProductAttribute::ctor(string) = ( 01 00 05 74 61 73 68 31 00 00 ) // ...task1..
    .custom instance void [mscorlib]System.Reflection.AssemblyCopyrightAttribute::ctor(string) = ( 01 00 1E 43 6F 70 79 72 69 67 68 74 20 C2 A9 20 // ...Copyright ..
        45 50 41 4D 20 53 79 73 74 65 6D 73 20 32 30 32 // EPAM Systems 202
        30 00 00 ) // 0..
    .custom instance void [mscorlib]System.Reflection.AssemblyTrademarkAttribute::ctor(string) = ( 01 00 00 00 00 00 )
    .custom instance void [mscorlib]System.Runtime.InteropServices.ComVisibleAttribute::ctor(bool) = ( 01 00 00 00 00 00 )
    .custom instance void [mscorlib]System.Runtime.InteropServices.GuidAttribute::ctor(string) = ( 01 00 24 37 61 62 35 32 39 39 34 2D 34 35 37 65 // ...$7ab52994-457e
        2D 34 33 66 33 2D 61 62 39 34 2D 63 63 34 63 38 // -43f3-ab94-cc4c8
        35 63 61 63 36 33 33 00 00 ) // 5cac633..
    .custom instance void [mscorlib]System.Reflection.AssemblyFileVersionAttribute::ctor(string) = ( 01 00 07 31 2E 30 2E 30 2E 30 00 00 ) // ...1.0.0.0..
    .custom instance void [mscorlib]System.Runtime.Versioning.TargetFrameworkAttribute::ctor(string) = ( 01 00 10 2E 4E 45 54 46 72 61 6D 65 77 6F 72 6D // ...NETFramework
        2E 56 65 72 73 69 6F 6E 3D 76 34 2E 37 2E 32 01 // .Version=v4.7.2.
        00 54 0E 14 46 72 61 6D 65 77 6F 72 6D 44 69 72 // .T.FrameworkDIs
        70 6C 61 79 4E 61 6D 65 14 2E 4E 45 54 20 46 72 // playName=.NET Fr
        61 6D 65 77 6F 72 6D 20 34 2E 37 2E 32 ) // anework 4.7.2

    .hash algorithm 0x00000004
    .ver 11:1:1:1
}
module task1.exe
// GUID: {F129B58E-D923-4D06-A364-489D87CF0D0C}
.imagebase 0x00400000
.file alignment 0x00000200
.stackreserve 0x00100000
.subsystem 0x0003 // WINDOWS_CUI
.corflags 0x00020003 // ILONLY 32BITPREFERRED
// Image base: 0x00360000

```

### TypeDef #1 (02000002)

```

-----
TypeDefName: task1.Program (02000002)
Flags       : [Public] [AutoLayout] [Class] [AnsiClass] [BeforeFieldInit] (00100001)
Extends     : 01000010 [TypeRef] System.Object
Method #1 (06000001) [ENTRYPOINT]

```

```

-----
MethodName: Main (06000001)
Flags       : [Private] [Static] [HideBySig] [ReuseSlot] (00000091)
RVA         : 0x00002050
ImplFlags   : [IL] [Managed] (00000000)
CallConvntn: [DEFAULT]
ReturnType: Void
1 Arguments
    Argument #1: SZArray String
1 Parameters
    (1) ParamToken : (08000001) Name : args flags: [none] (00000000)

```

### Method #2 (06000002)

```

-----
MethodName: .ctor (06000002)
Flags       : [Public] [HideBySig] [ReuseSlot] [SpecialName] [RTSpecialName] [.ctor] (00001886)
RVA         : 0x0000205e
ImplFlags   : [IL] [Managed] (00000000)
CallConvntn: [DEFAULT]
hasThis
ReturnType: Void
No arguments.

```

Вывод: не зависимо от типа сборки, статическая или динамическая, любая сборка содержит коллекцию данных с описанием того, как ее элементы связаны друг с другом. Эти метаданные содержатся в манифесте сборки. Манифест сборки содержит все метаданные, необходимые для задания требований сборки к версиям, а также все метаданные, необходимые для определения области действия сборки и разрешения ссылок на ресурсы и классы.

## 2) Задание 2 «Method GetCurentTime()»

### TypeDef #1 (02000002)

```
-----
TypeDefName: task1.Program (02000002)
Flags       : [Public] [AutoLayout] [Class] [AnsiClass] [BeforeFieldInit] (00100001)
Extends     : 01000010 [TypeRef] System.Object
Method #1 (06000001) [ENTRYPOINT]
```

```
-----
MethodName: Main (06000001)
Flags       : [Private] [Static] [HideBySig] [ReuseSlot] (00000091)
RVA        : 0x00002050
ImplFlags   : [IL] [Managed] (00000000)
CallConvntn: [DEFAULT]
ReturnType: Void
1 Arguments
    Argument #1: SZArray String
1 Parameters
    (1) ParamToken : (08000001) Name : args flags: [none] (00000000)
```

### Method #2 (06000002)

```
-----
MethodName: GetCurentTime (06000002)
Flags       : [Private] [Static] [HideBySig] [ReuseSlot] (00000091)
RVA        : 0x00002064
ImplFlags   : [IL] [Managed] (00000000)
CallConvntn: [DEFAULT]
ReturnType: ValueClass System.DateTime
No arguments.
```

### Method #3 (06000003)

```
-----
MethodName: .ctor (06000003)
Flags       : [Public] [HideBySig] [ReuseSlot] [SpecialName] [RTSpecialName] [.ctor] (00001886)
RVA        : 0x0000207b
ImplFlags   : [IL] [Managed] (00000000)
CallConvntn: [DEFAULT]
hasThis
ReturnType: Void
No arguments.
```

Вывод: при добавление нового метода GetCurentTime() соответственно обновились и методанные типа Program. Добавилась метаинформации этого метода, такая как: MethodName – название метода; Flags и ImplFlags - битовые маски, описывающие метод (например, является ли метод статическим, приватным или публичным); RVA -

позволяют среде выполнения вычислить начальный адрес в памяти IL кода, определяющего этот метод; Return Type – тип, который возвращает этот метод; и др.

### 3) Задание 3 «Custom DLL»

```
// Metadata version: v4.0.30319
.assembly extern mscorlib
{
    .publickeytoken = (B7 7A 5C 56 19 34 E0 89 ) // .z\U.4..
    .ver 4:0:0:0
}
.assembly extern ClassLibrary1
{
    .ver 1:0:0:0
}
.assembly task1
{
    .custom instance void [mscorlib]System.Runtime.CompilerServices.CompilationRelaxationsAttribute::.ctor(int32) = ( 01 00 08 00 00 00 00 00 )
    .custom instance void [mscorlib]System.Runtime.CompilerServices.RuntimeCompatibilityAttribute::.ctor() = ( 01 00 01 00 54 02 16 57 72 61 70 4E 6F 6E 45 78 // ....T..WrapNonEx
    63 65 70 74 69 6F 6E 54 68 72 6F 77 73 01 ) // ceptionThrows.

    // --- The following custom attribute is added automatically, do not uncomment -----
    // .custom instance void [mscorlib]System.Diagnostics.DebuggableAttribute::.ctor(valueType [mscorlib]System.Diagnostics.DebuggableAttribute/DebuggingModes) = ( 01 00 07 01 00 00 00 00 )

    .custom instance void [mscorlib]System.Reflection.AssemblyTitleAttribute::.ctor(string) = ( 01 00 05 74 61 73 68 31 00 00 ) // ...task1..
    .custom instance void [mscorlib]System.Reflection.AssemblyDescriptionAttribute::.ctor(string) = ( 01 00 00 00 00 )
    .custom instance void [mscorlib]System.Reflection.AssemblyConfigurationAttribute::.ctor(string) = ( 01 00 00 00 00 )
    .custom instance void [mscorlib]System.Reflection.AssemblyCompanyAttribute::.ctor(string) = ( 01 00 0C 45 50 41 4D 20 53 79 73 74 65 6D 73 00 // ...EPAM Systems.
    00 )
    .custom instance void [mscorlib]System.Reflection.AssemblyProductAttribute::.ctor(string) = ( 01 00 05 74 61 73 68 31 00 00 ) // ...task1..
    .custom instance void [mscorlib]System.Reflection.AssemblyCopyrightAttribute::.ctor(string) = ( 01 00 1E 43 6F 70 79 72 69 67 68 74 20 C2 A9 20 // ...Copyright ..
    45 50 41 4D 20 53 79 73 74 65 6D 73 20 32 30 32 // EPAM Systems 202
    30 00 00 ) // 0..
    .custom instance void [mscorlib]System.Reflection.AssemblyTrademarkAttribute::.ctor(string) = ( 01 00 00 00 00 )
    .custom instance void [mscorlib]System.Runtime.InteropServices.ComVisibleAttribute::.ctor(bool) = ( 01 00 00 00 00 )
    .custom instance void [mscorlib]System.Runtime.InteropServices.GuidAttribute::.ctor(string) = ( 01 00 24 37 61 62 35 32 39 39 34 2D 34 35 37 65 // ..$7ab52994-b57e
    20 34 33 66 33 2D 61 62 39 34 2D 63 63 34 63 38 // -43f9-ab94-cc4c8
    35 63 61 63 36 33 33 00 00 ) // 5cac633..
    .custom instance void [mscorlib]System.Reflection.AssemblyFileVersionAttribute::.ctor(string) = ( 01 00 07 31 2E 30 2E 30 2E 30 00 00 ) // ...1.0.0.0..
    .custom instance void [mscorlib]System.Runtime.Versioning.TargetFrameworkAttribute::.ctor(string) = ( 01 00 1C 2E 4E 45 54 46 72 61 6D 65 77 6F 72 68 // ....NETFramework
    2C 56 65 72 73 69 6F 6E 3D 76 34 2E 37 2E 32 01 // ,Version=v4.7.2.
    00 54 0E 14 46 72 61 6D 65 77 6F 72 68 44 69 73 // .I..FrameworkDis
    70 6C 61 79 4E 61 6D 65 14 2E 4E 45 54 20 46 72 // playName..NET Fr
    61 6D 65 77 6F 72 68 20 34 2E 37 2E 32 ) // anework 4.7.2

    .hash algorithm 0x00000004
    .ver 11:1:1:1
}
.module task1.exe
// GUID: {91644BD9-FE98-4A2F-B0D8-1E42358C18A6}
.imagebase 0x00400000
.file alignment 0x00000200
.stackreserve 0x00100000
.subsystem 0x0003 // WINDOWS_CUI
.corflags 0x00020003 // ILONLY 32BITPREFERRED
// Image base: 0x09E40000
```

Вывод: ссылки на внешние библиотеки (.dll) хранятся в манифесте сборки. Инструкция .publickeytoken указывается только тогда, когда подключаемая библиотека имеет строгую форму имени. Лексема .ver обозначает числовую идентификацию версии этой библиотеки.

### 4) Задание 4 «Debug vs Release»

```

.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    // Code size      76 (0x4c)
    .maxstack 5
    .locals init ([0] int32 i,
                  [1] int32 j,
                  [2] bool V_2,
                  [3] bool V_3)

    IL_0000: nop
    IL_0001: ldc.i4.0
    IL_0002: stloc.0
    IL_0003: br.s      IL_0042
    IL_0005: nop
    IL_0006: ldc.i4.0
    IL_0007: stloc.1
    IL_0008: br.s      IL_0034
    IL_000a: nop
    IL_000b: ldstr      "{0} * {1} = {2}"
    IL_0010: ldloc.0
    IL_0011: box        [mscorlib]System.Int32
    IL_0016: ldloc.1
    IL_0017: box        [mscorlib]System.Int32
    IL_001c: ldloc.0
    IL_001d: ldloc.1
    IL_001e: mul
    IL_001f: box        [mscorlib]System.Int32
    IL_0024: call       string [mscorlib]System.String::Format(string,
                                                                object,
                                                                object,
                                                                object)

    IL_0029: call       void [mscorlib]System.Console::WriteLine(string)
    IL_002e: nop
    IL_002f: nop
    IL_0030: ldloc.1
    IL_0031: ldc.i4.1
    IL_0032: add
    IL_0033: stloc.1
    IL_0034: ldloc.1
    IL_0035: ldc.i4.s    10
    IL_0037: clt
    IL_0039: stloc.2
    IL_003a: ldloc.2
    IL_003b: brtrue.s    IL_000a
    IL_003d: nop
    IL_003e: ldloc.0
    IL_003f: ldc.i4.1
    IL_0040: add
    IL_0041: stloc.0
    IL_0042: ldloc.0
    IL_0043: ldc.i4.s    10
    IL_0045: clt
    IL_0047: stloc.3
    IL_0048: ldloc.3
    IL_0049: brtrue.s    IL_0005
    IL_004b: ret
} // end of method Program::Main

```

```

.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    // Code size      62 (0x3e)
    .maxstack 5
    .locals init ([0] int32 i,
                  [1] int32 j)
    IL_0000: ldc.i4.0
    IL_0001: stloc.0
    IL_0002: br.s      IL_0038
    IL_0004: ldc.i4.0
    IL_0005: stloc.1
    IL_0006: br.s      IL_002f
    IL_0008: ldstr      "{0} * {1} = {2}"
    IL_000d: ldloc.0
    IL_000e: box        [mscorlib]System.Int32
    IL_0013: ldloc.1
    IL_0014: box        [mscorlib]System.Int32
    IL_0019: ldloc.0
    IL_001a: ldloc.1
    IL_001b: mul
    IL_001c: box        [mscorlib]System.Int32
    IL_0021: call      string [mscorlib]System.String::Format(string,
                                                             object,
                                                             object,
                                                             object)
    IL_0026: call      void [mscorlib]System.Console::WriteLine(string)
    IL_002b: ldloc.1
    IL_002c: ldc.i4.1
    IL_002d: add
    IL_002e: stloc.1
    IL_002f: ldloc.1
    IL_0030: ldc.i4.s    10
    IL_0032: blt.s     IL_0008
    IL_0034: ldloc.0
    IL_0035: ldc.i4.1
    IL_0036: add
    IL_0037: stloc.0
    IL_0038: ldloc.0
    IL_0039: ldc.i4.s    10
    IL_003b: blt.s     IL_0004
    IL_003d: ret
} // end of method Program::Main

```

Вывод: в конфигурации Debug код компилируется с полной информацией, необходимой для отладки программы, и без оптимизации. Оптимизация усложняет отладку, поскольку усложняется связь между исходным кодом и сгенерированными инструкциями. В случае конфигурация Release, код не имеет этой отладочной информации и полностью оптимизирован.