

Лекція 1

АРХІТЕКТУРА .NET



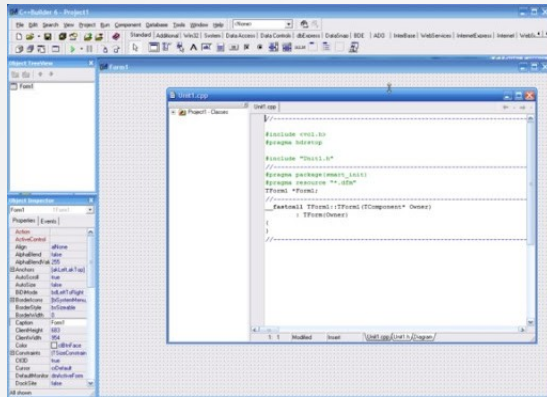
Лекція 1. Архітектура .NET

План

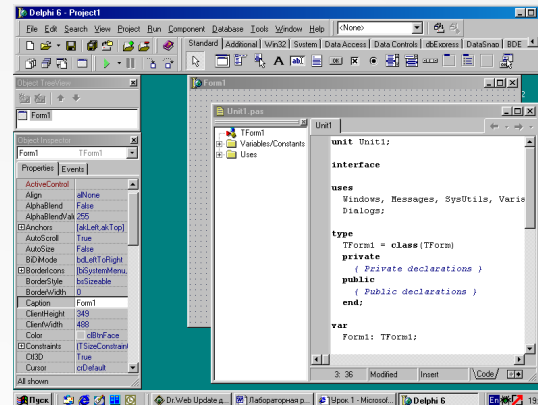
1. Історія створення платформи .NET
2. Архітектура .NET
3. Бібліотека базових класів
4. Мови програмування .NET
5. Загальномовне середовище виконання
6. IDE для .NET
7. Література

1. Історія створення платформи .NET

IDE Borland C++ Builder
(мова програмування C++)



IDE Borland Delphi
(Object Pascal)



Бібліотека візуальних компонентів VCL
(Visual Component Library)

Додаток Windows

У середині **90-х** років минулого століття фірма **Borland** випустила два інтегровані середовища розробки (**IDE** – Integrated Development Environment) для створення програм мовами програмування **C++** та **Object Pascal (Delphi)**. Вони базувалися на використанні єдиної бібліотеки графічних компонентів **VCL**, призначеної для написання різних типів програм для ОС Windows.

1. Історія створення платформи .NET

У **1995** році фірма **Sun Microsystems** розробила високорівневу об'єктно-орієнтовану мову програмування **Java**. Її особливістю є те, що вона спочатку транслюється в спеціальний **байт-код**, який може бути виконаний **віртуальною java-машиною (JVM)** на будь-якій комп'ютерній архітектурі, де вона є в наявності.



Такий підхід виявився дуже вдалим, тому що вдалося досягти реальної **кросплатформенності**. Він став основою для створеної пізніше специфікації загальномовної інфраструктури **CLI (Common Language Infrastructure)**.



У **1999** році фірма **Microsoft** розпочала розробку своєї альтернативи Java і в **2002** році було оголошено про створення нової платформи програмування, яка отримала назву **.NET Framework**. Основою .NET є загальномовне середовище виконання **CLR (Common Language Runtime)**, в якому можуть працювати різні мови програмування, такі, наприклад, як C#, F#, J#, Ada, C++ та ін.

2. Архітектура .NET

З погляду програміста платформа .NET є **середовищем виконання** і повнофункціональною об'єктно-орієнтованою **бібліотекою базових класів**, що реалізують весь необхідний функціонал для розробки додатків різних типів у Windows.

Загальномовне середовище виконання CLR (Common Language Runtime) призначене для автоматичного виявлення, завантаження та управління об'єктами.

Загальна система типів CTS (Common Type System) описує всі можливі типи даних та всі програмні конструкції, що підтримуються CLR. Крім того, у CTS показано, як ці сутності можуть взаємодіяти один з одним, і як вони представлені у форматі метаданих .NET.

Загальномовна специфікація CLS (Common Language Specification) описує підмножину загальних типів та програмних конструкцій, які повинні підтримуватися усіма мовами програмування для .NET.

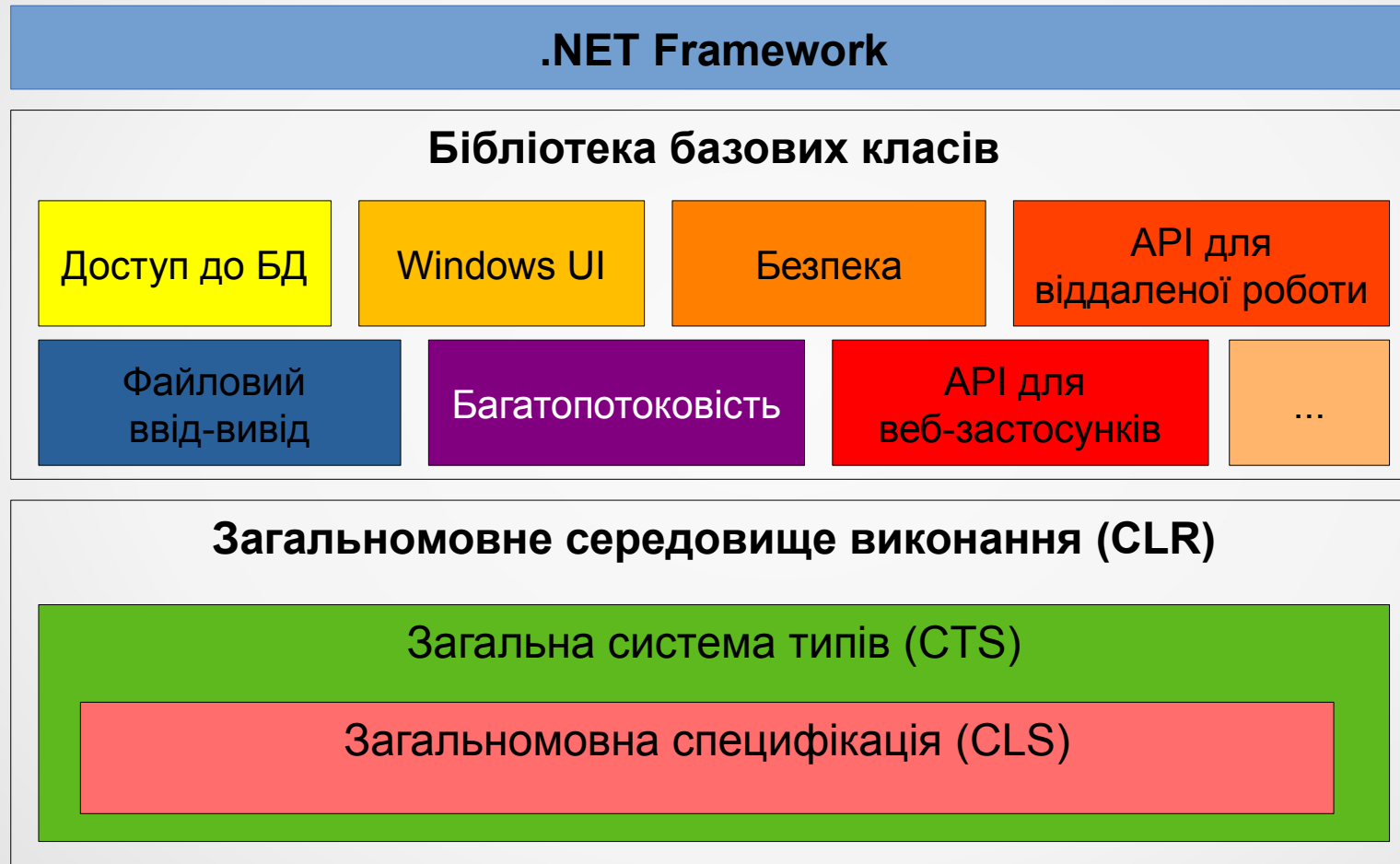
2. Архітектура .NET

Додатково до середовища CLR та специфікацій CTS/CLS платформа .NET надає **бібліотеку базових класів**, яка доступна всім мовам програмування .NET. Ця бібліотека не тільки інкапсулює різноманітні примітиви, такі, наприклад, як **потоки виконання** (thread), **потоки вводу-виводу** (stream), дво- та тривимірну **графіку** та механізми взаємодії з різними зовнішніми пристроями, але також забезпечує підтримку для численних служб, потрібних більшості реальних додатків.

Бібліотеки базових класів визначають типи, які можна використовувати для побудови програм будь-якого виду. Наприклад, **ASP.NET** можна застосовувати для побудови веб-сайтів, **WCF** – для створення мережеских служб, **WPF** – для написання настільних додатків з графічним інтерфейсом користувача і т.д.

Крім того, бібліотеки базових класів надають типи для взаємодії з XML-та JSON-документами, файловими системами, комунікації з базами даних (**ADO.NET**) і багато іншого.

2. Архітектура .NET

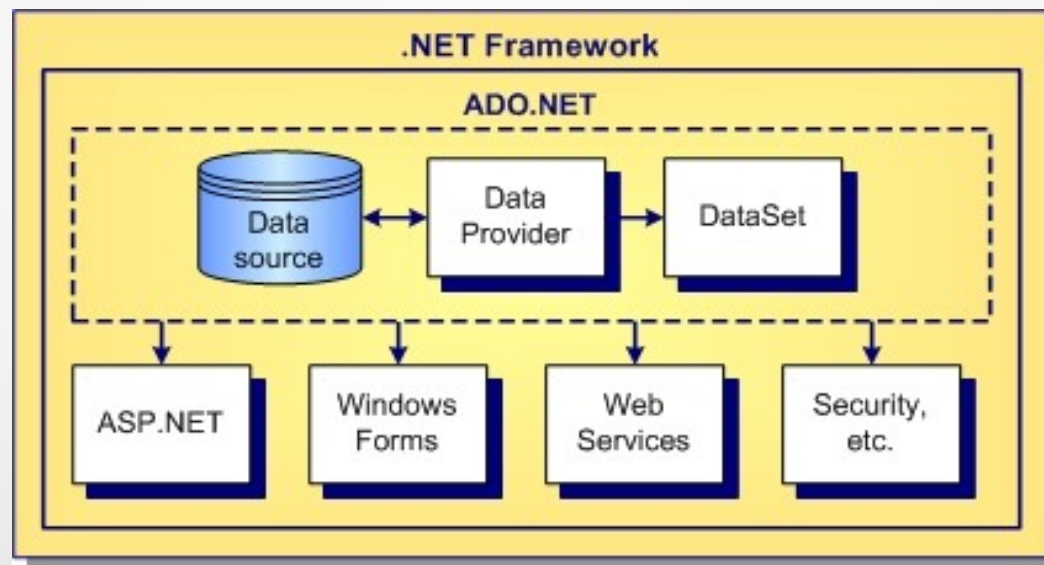


Загальна структура платформи .NET Framework

3. Бібліотека базових класів



ADO.NET (ActiveX Data Object для .NET) – набір класів та бібліотек для платформи .NET Framework, які дозволяють отримати доступ і керувати інформацією, що зберігається в базах даних або інших джерелах (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, текстові файли).



3. Бібліотека базових класів



ASP.NET (Active Server Pages для .NET) – набір класів для розробки веб-застосунків для платформи .NET Framework.

Включає такі компоненти:

- **ASP.NET Web Forms** – інфраструктура конструювання веб-сторінок;
- **ASP.NET MVC** – фреймворк для створення веб-застосунків, що реалізує шаблон Модель-Вигляд-Контролер (MVC – Model-View-Controller);
- **ASP.NET Dynamic Data** – платформа створення веб-застосунків, керованих даними;
- **ASP.NET Web API** – інфраструктура для простої взаємодії між клієнтом та сервером, заснована на ASP.NET MVC і використовує контролери та маршрутизацію.



3. Бібліотека базових класів



WCF (Windows Communication Foundation) – це функціональна комунікаційна технологія, за допомогою якої можна організовувати взаємодію як на основі **REST**, так і на основі **SOAP** з усіма можливостями, характерними для стандартизованих веб-служб, включаючи підтримку захисту, транзакцій, одно- та двонаправлених комунікацій, маршрутизації, виявлення тощо.



WWF (Windows Workflow Foundation) – інфраструктура, що призначена для визначення, виконання та керування робочими процесами. WWF підтримує три типи процесів:

- 1) послідовний процес;
- 2) скінченний автомат;
- 3) процес, керований правилами.

3. Бібліотека базових класів



Windows Forms

WF (Windows Forms) – набір класів, призначений для створення графічного інтерфейсу користувача (**GUI** – Graphical User Interface) і є частиною Microsoft .NET Framework. Реалізує доступ до стандартних елементів інтерфейсу MS Windows за рахунок створення обгортки для існуючого **Win32 API** в керованому коді. Заснований на піксельній графіці.



WPF (Windows Presentation Foundation) – нова інфраструктура створення GUI для платформи .NET Framework, що базується на векторній графіці. При побудові додатків інфраструктура WPF спирається на використання **XAML** (eXtensible Application Markup Language) – мові розмітки додатків, що розширюється.

4. Мови програмування .NET



Нативною (природною) для платформи .NET є спеціально розроблена в 1998-2001 рр. фірмою Microsoft об'єктно-орієнтована мова **C# (C Sharp)**.

Програма «Hello World»

```
using System;

namespace hello
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

4. Мови програмування .NET



F# – розроблена у 2005 році мультипарадигмальна мова програмування сімейства .NET Framework, яка підтримує **функціональне, імперативне та об'єктно-орієнтоване** програмування.

Програма «Hello World»

// Learn more about F# at <http://fsharp.org>

open System

[<EntryPoint>]

let main argv =

printfn "Hello World from F#!"

0 // return an integer exit code

Програма обчислення факторіала

// Learn more about F# at <http://fsharp.org>

open System

let rec fac n =

if n < 2 then 1

else n * fac(n - 1)

[<EntryPoint>]

let main argv =

let v = fac 5

printfn "Factorial %d." v

0 // return an integer exit code

4. Мови програмування .NET



Visual Basic .NET – розроблена у **2001** році об'єктно-орієнтована мова програмування сімейства, що є розвитком **Visual Basic**.

Консольна програма «Hello World»

```
Module Module1
    Sub Main()
        Console.WriteLine("Hello, World!")
    End Sub
End Module
```

Програма «Hello World» з GUI

```
Private Sub HW()
    Dim helloworld As New Label
    With helloworld
        .Text = "Hello, World"
        .AutoSize = vbYes
        .Font = New Font(.Text, 48)
        .Location = new point(100, 100)
        .Visible = True
    End With
    Me.Controls.Add(helloworld)
End sub
```

4. Мови програмування .NET



C++/CLI – розроблений у **2002** році розвиток мови C++ для середовища програмування Microsoft .NET. Є частиною загальномовної інфраструктури CLI (Common Language Infrastructure), а також подальшим розвитком **Managed C++**.

Програма «Hello World»

```
#include "pch.h"
#include <iostream>

using namespace System;

int main(array<System::String ^> ^args)
{
    std::cout << "Hello world!" << std::endl;

    return 0;
}
```


4. Мови програмування .NET



JScript.NET – мова програмування (розвиток JavaScript), що підтримує об'єктно-орієнтовану, імперативну та функціональну парадигми. Використовується як вбудована мова для програмного доступу до об'єктів програм (наприклад, як мова сценаріїв для надання інтерактивності веб-сторінкам).

Програма «Hello World»

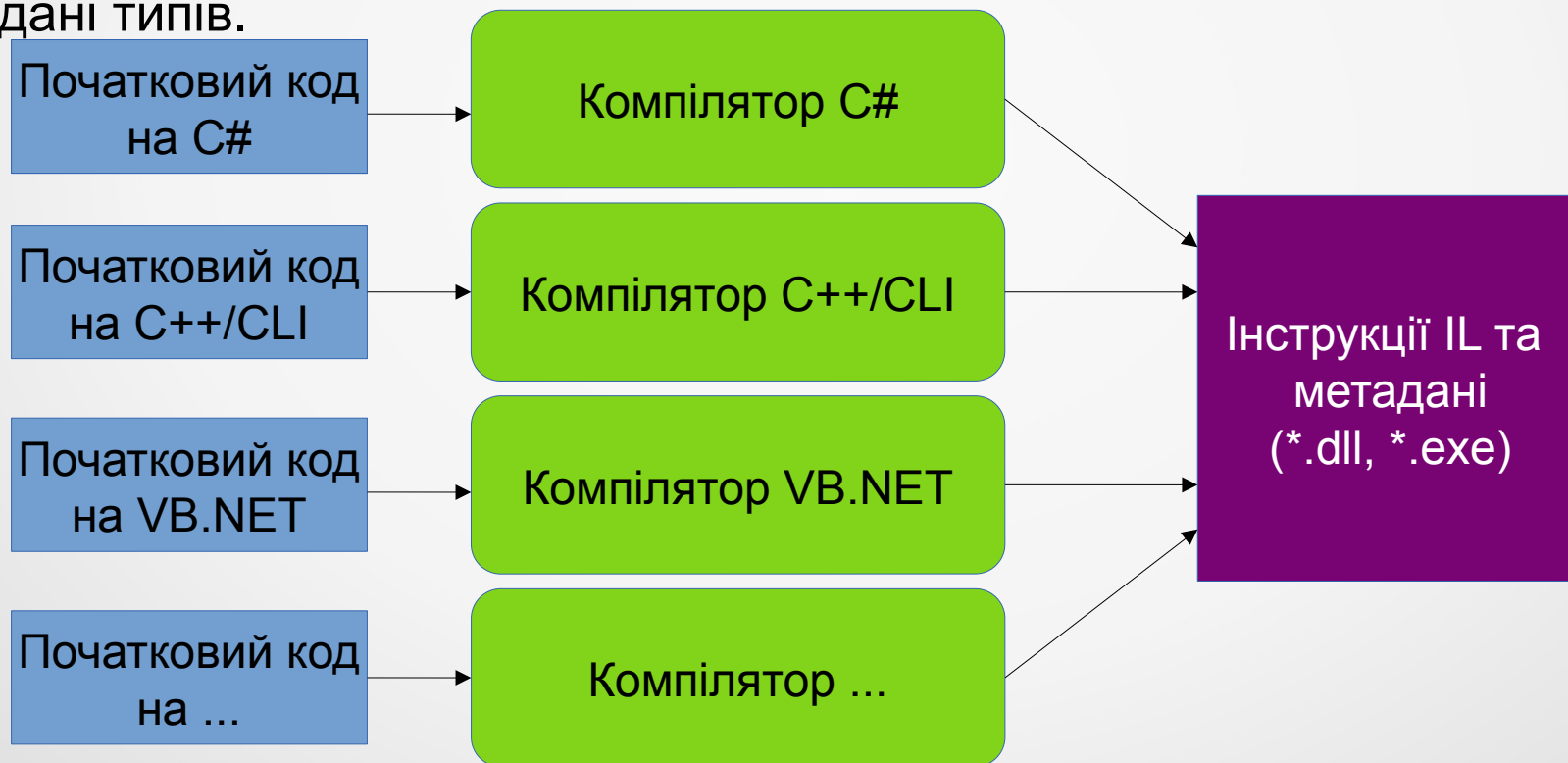
(фрагмент)

```
'use strict';  
var debug = require('debug');  
var express = require('express');  
var path = require('path');  
var favicon = require('serve-favicon');  
var logger = require('morgan');  
var cookieParser = require('cookie-parser');  
var bodyParser = require('body-parser');  
  
var routes = require('./routes/index');  
var users = require('./routes/users');
```

```
var app = express();  
  
// view engine setup  
app.set('views', path.join(__dirname, 'views'));  
app.set('view engine', 'pug');  
// uncomment after placing your favicon in /public  
//app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));  
app.use(logger('dev'))
```


5. Загальномовне середовище виконання

Програми, написані будь-якою мовою .NET, мають стандартні для Windows розширення (*.dll або *.exe), але внутрішньо вони влаштовані зовсім по-іншому. Зокрема, двійкові модулі .NET містять не специфічні, а незалежні від платформи інструкції проміжною мовою **IL** (Intermediate Language) та метадані типів.

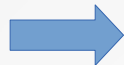


Будь-який .NET-компілятор генерує інструкції IL та метадані 17

5. Загальномовне середовище виконання

C#

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

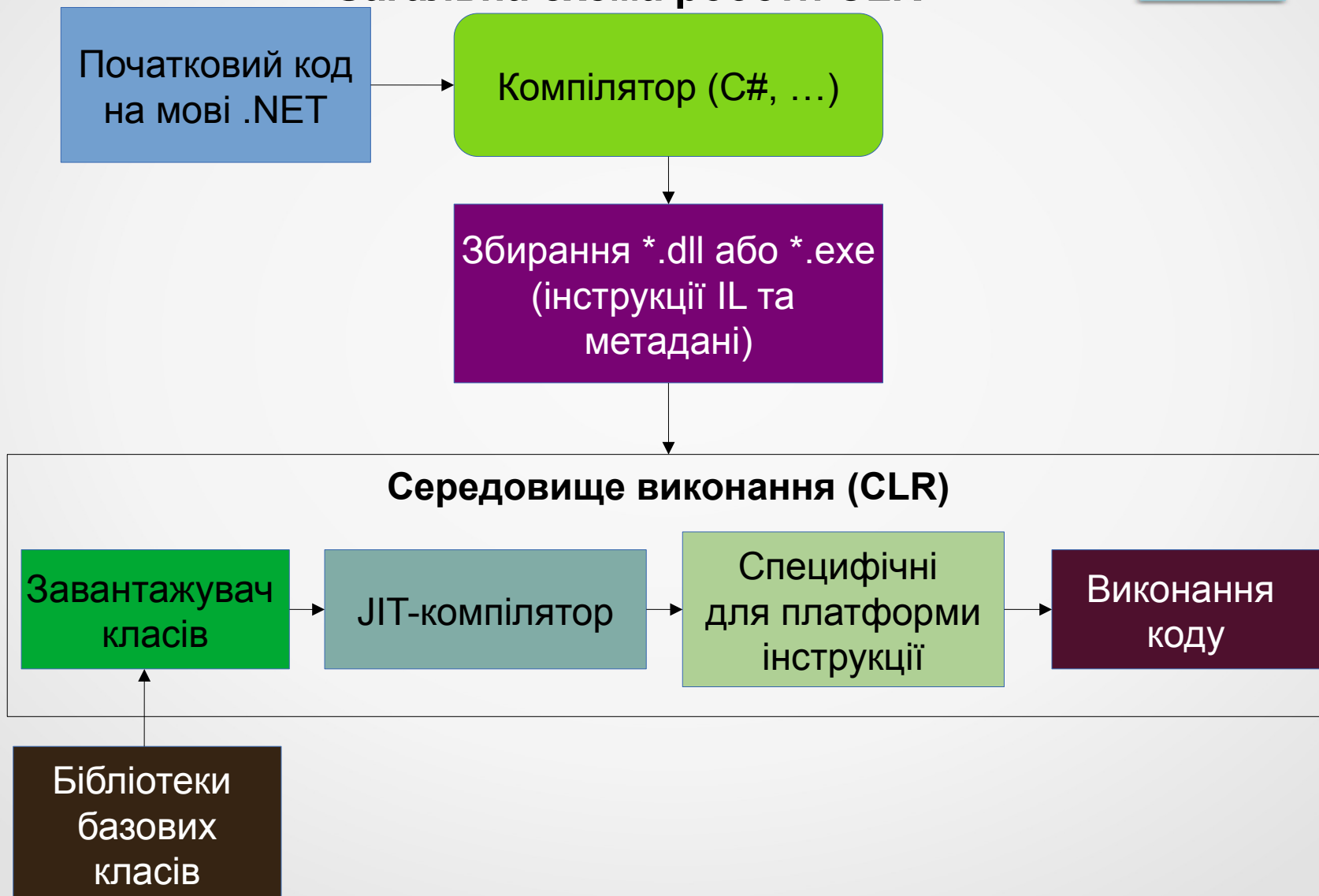


IL

```
.method public hidebysig instance int32 Add(int32
x,
int32 y) cil managed
{
    // Code size 9 (0x9)
    .maxstack 2
    .locals init (int32 V_0)
    IL_0000: nop
    IL_0001: ldarg.l
    IL_0002: ldarg.2
    IL_0003: add
    IL_0004: stloc.O
    IL_0005: br.s IL_0007
    IL_0007: ldloc.O
    IL_0008: ret
} // end of method Calc::Add
```

5. Загальномовне середовище виконання

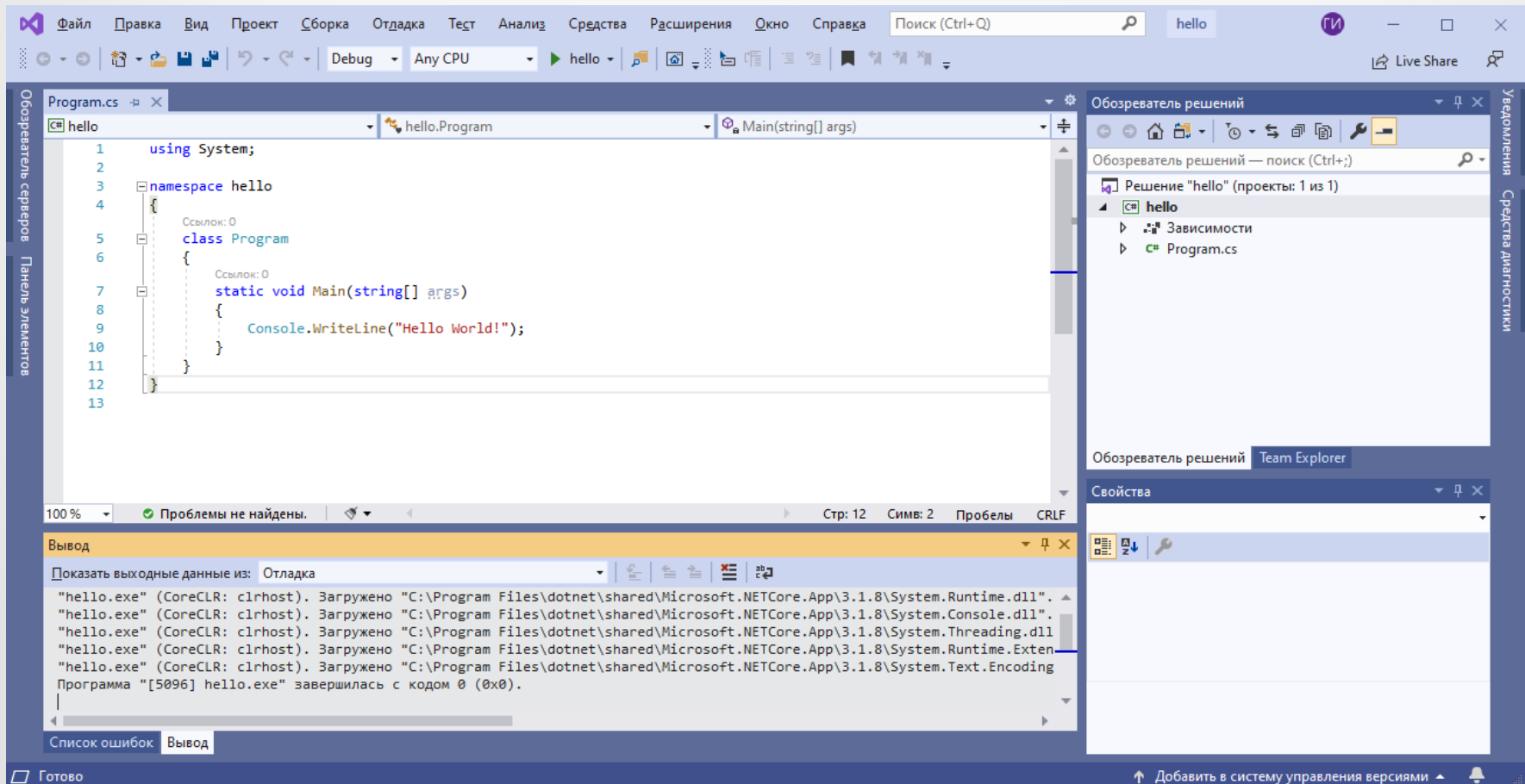
Загальна схема роботи CLR



6. IDE для .NET



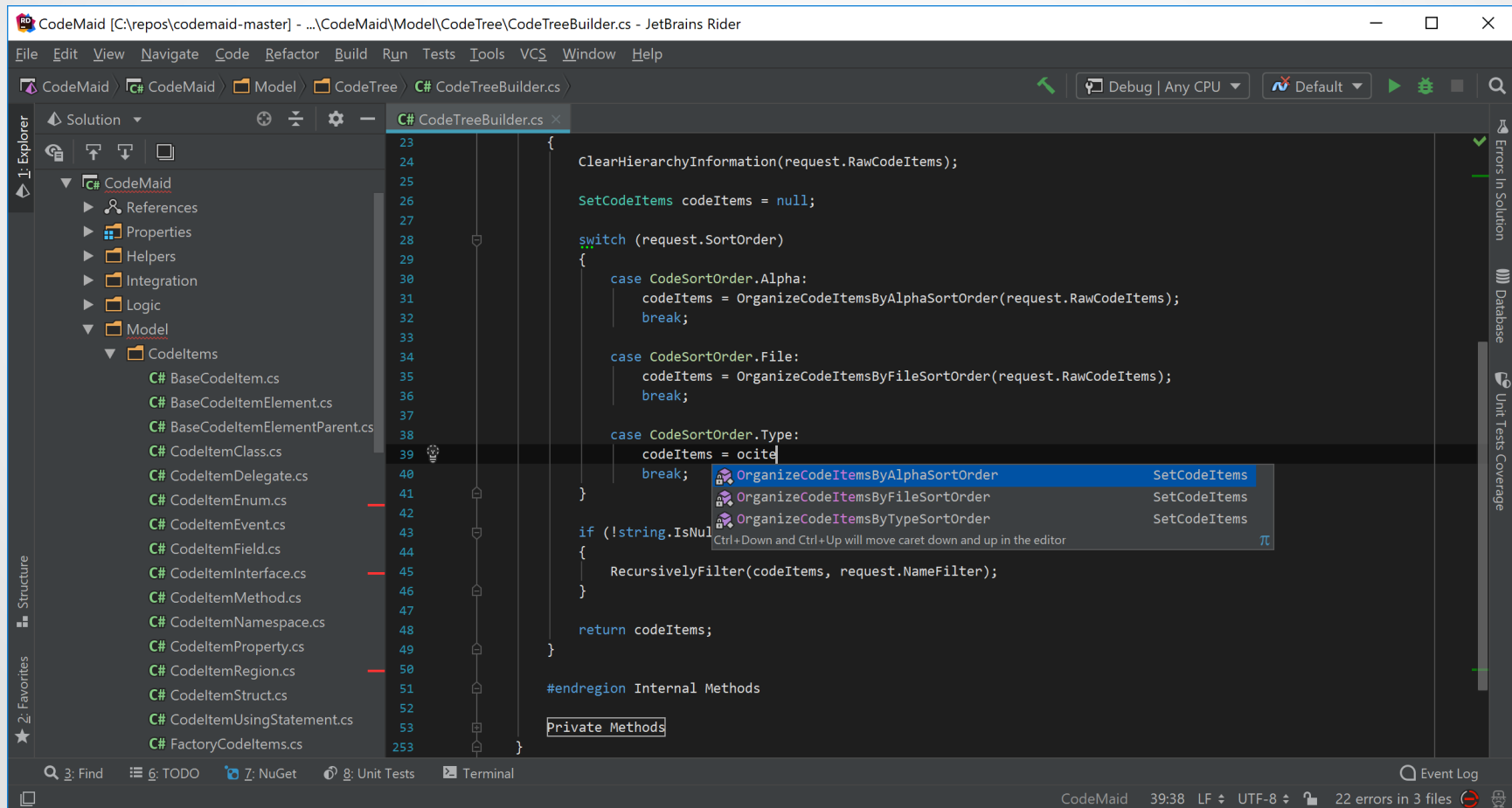
Microsoft Visual Studio



6. IDE для .NET



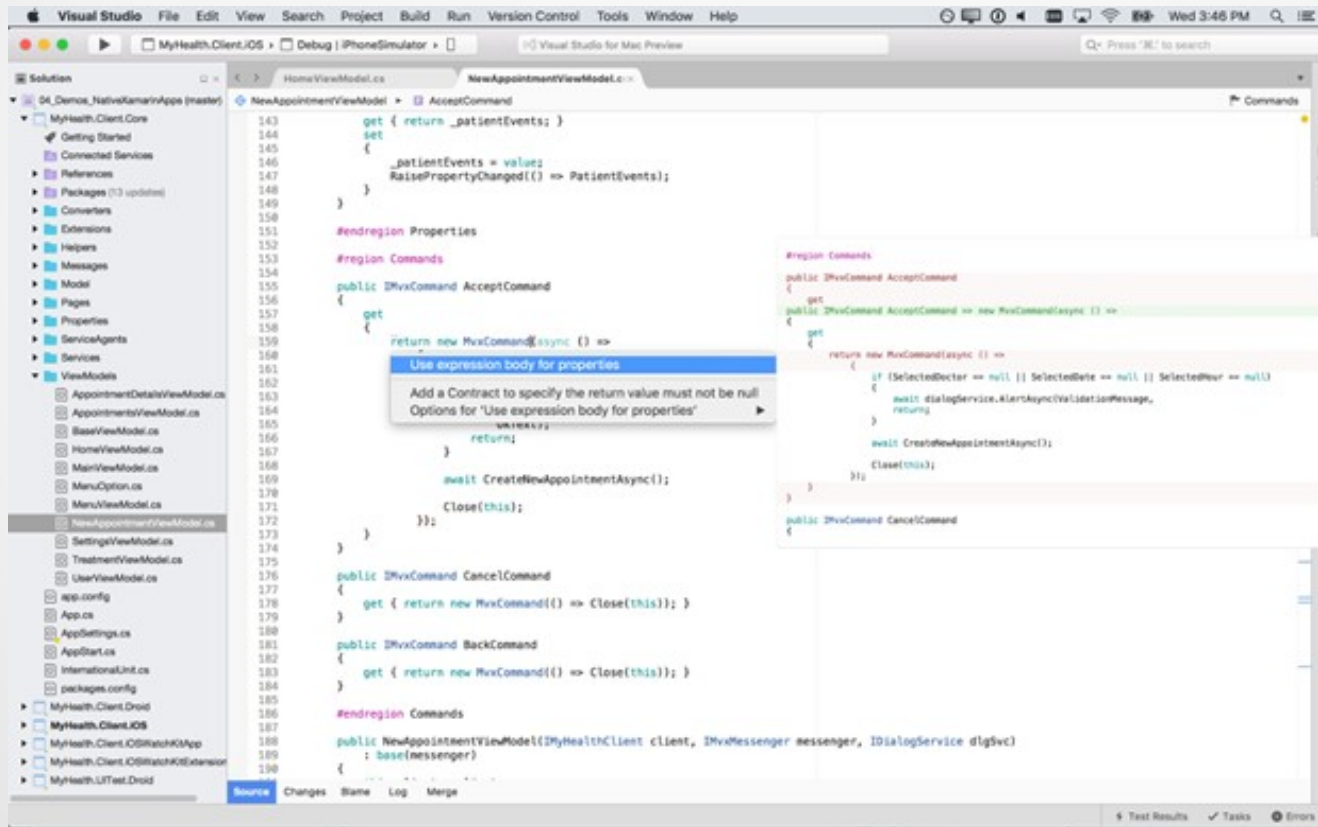
JetBrains Project Rider



6. IDE для .NET



MonoDevelop



5. Литература

1. Nagel C., Glynn J., Skinner M. Professional C# 5.0 and .NET 4.5.1. John Wiley & Sons, 2014. 1560 p.
2. Richter J. CLR via C#. Microsoft Press, 2012. 896 p.
3. C# Programming Guide. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/?redirectedfrom=MSDN>
(дата звернення: 09.08.2023).
4. Бармак О.В. Visual studio. NET. Створення Windows-програм. Хмельницький: ХНУ, 2006. 90 с.
5. Home – Ukrainian .NET Developer Community. URL: <https://dotnet.city/>
(дата звернення: 09.08.2023).