Quality software. On time. Every time.

Georgescu Claudiu, Dev Team Leader
www.essensys.ro

Microsoft GOLD CERTIFIED Partner

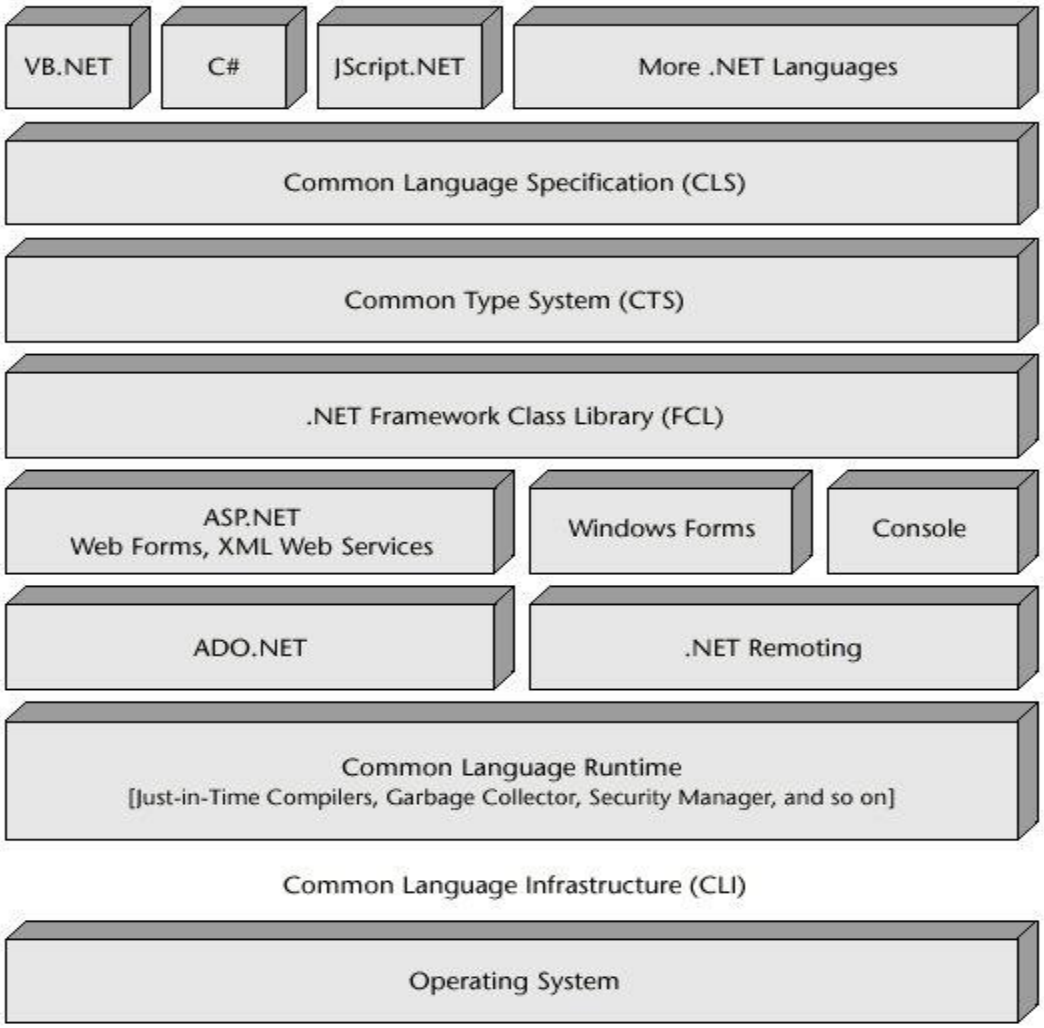
Data Management Solutions
Custom Development Solutions

Agenda

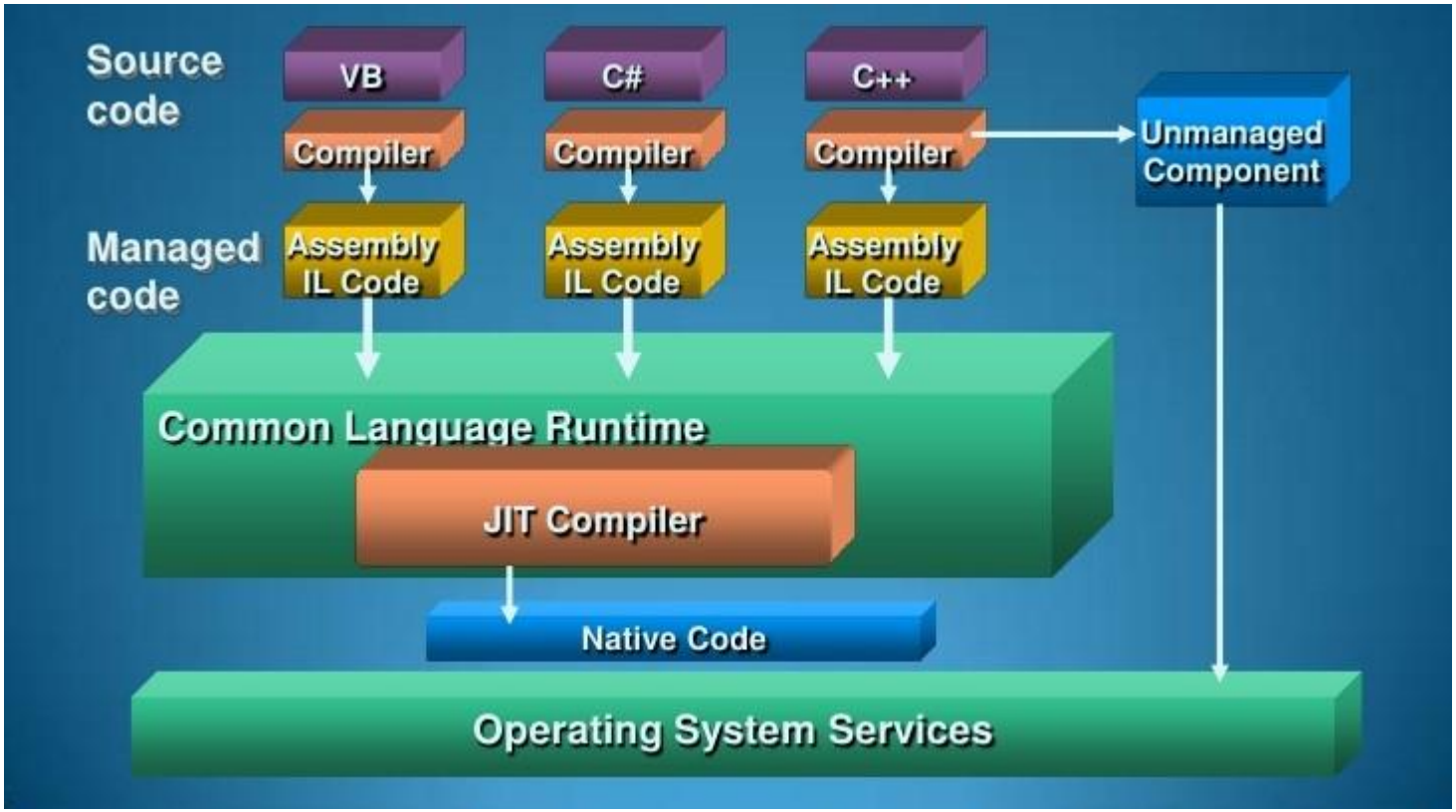
1. What is .NET?
2. How does it work?
3. Hello world
4. Visual Studio & NuGet
5. Data types
6. Namespaces
7. Classes/Constructors/Methods/Properties
8. Inheritance
9. Polymorphism
10. Encapsulation
11. Exception handling
12. Disposables
13. What is POCO & DTO?
14. What is N-Tier application?

1 What is .NET?

ESSENSYS™
SOFTWARE SOLUTIONS



2 How does it all work?



Working on:
C# 6
Visual Studio ’14’(2015)
.NET 4.5.2

Latest:
C# 7
Visual Studio ’15’(2017)
Stable: .NET 4.6.2
Cross platform: .NET Core 1.1

3 Hello World

ESSENSYS™
SOFTWARE SOLUTIONS

DEMO

4 Visual Studio & NuGet

ESSENSYS™ SOFTWARE SOLUTIONS

NuGet a package manager for the Microsoft development platform that provides the ability to consume or produce/distribute packages

5 Data types

ESSENSYS™ SOFTWARE SOLUTIONS

- Value types vs reference types
- Stack vs Heap

- Value types:
  Int, Boolean, float(single), double, decimal
  enums, custom structs
- Reference types:
  String(reference type that behaves like a value type)
  Classes
- Var

- Mutable vs immutable

- Casting: safe vs unsafe
- Out and Ref

6 Namespaces

ESSENSYS™ SOFTWARE SOLUTIONS

- Determines the fullname for types
- Represents a virtual path for types
- Assures no conflicts between types with same name
- When using types in another namespace
(even if child namespace) a using is required
- Intellisense can help with automatically adding usings

7 Classes/Constructors/Methods/Props

ESSENSYS™ SOFTWARE SOLUTIONS

- Classes: static/non-static
- Static method/constructor/property
  One method/constructor/property shared for all instances
- Non-static method/constructor/property
  One per instance
- Abstract classes/Abstract methods
- Virtual methods

8 Inheritance

- All classes inherit from object
- Vs C++: A class can only inherit from one class.
- But it can inherit any number of interfaces

(It makes for a simpler to understand inheritance model)

9 Polymorphism

ESSENSYS™ SOFTWARE SOLUTIONS

Code:

```
public abstract class Person
{
   public abstract string GetPersonType();
}

public class Student : Person
{
   public override string GetPersonType()
   {
      return nameof(Student); // "Student"
   }
}

public class Teacher : Person
{
   public override string GetPersonType()
   {
      return nameof(Teacher); // "Teacher"
   }
}
```

How to test in console app:

```
static void Main(string[] args)
{
   var persons = new Person[2];
   persons[0] = new Student();
   persons[1] = new Teacher();

   foreach (var person in persons)
   {
      Console.WriteLine(person.GetType());
   }
}
```

10 Encapsulation

ESSENSYS™ SOFTWARE SOLUTIONS

- **public**:
  Access is not restricted.

- **protected**:
  Access is limited to the containing class or types derived from the containing class

- **internal**:
  Access is limited to the current assembly

- **protected internal**:
  Access is limited to the current assembly or types derived from the containing class

- **private**:
  Access is limited to the containing type.

- They apply to: classes, properties, fields etc
- If nothing specified, lowest access modifier **available** is selected

Exception handling

- try, catch, throw and finally
- Catches can cascade from the more specific to the more general
- Only one catch will ever execute, the first one that fits from top to bottom
- If cascading, one should start catching specific errors and only then target more general ones
- Finally always gets executed
- Errors contain the stack trace which for a developer is sometimes more important than the actual message

12 Disposables

ESSENSYS™ SOFTWARE SOLUTIONS

For the majority of the objects that your app creates, you can rely on the .NET Framework's garbage collector to handle memory management. However, when you create objects that include **unmanaged resources, you must explicitly release those resources when you finish using them in your app**.

The most common types of unmanaged resource are objects that wrap operating system resources, such as files, windows, network connections, or database connections.

Although the garbage collector is able to track the lifetime of an object that encapsulates an unmanaged resource, it doesn't know how to release and clean up the unmanaged resource.

The caller disposes of unmanaged resources with the "Using" or try, catch, finally(same thing)

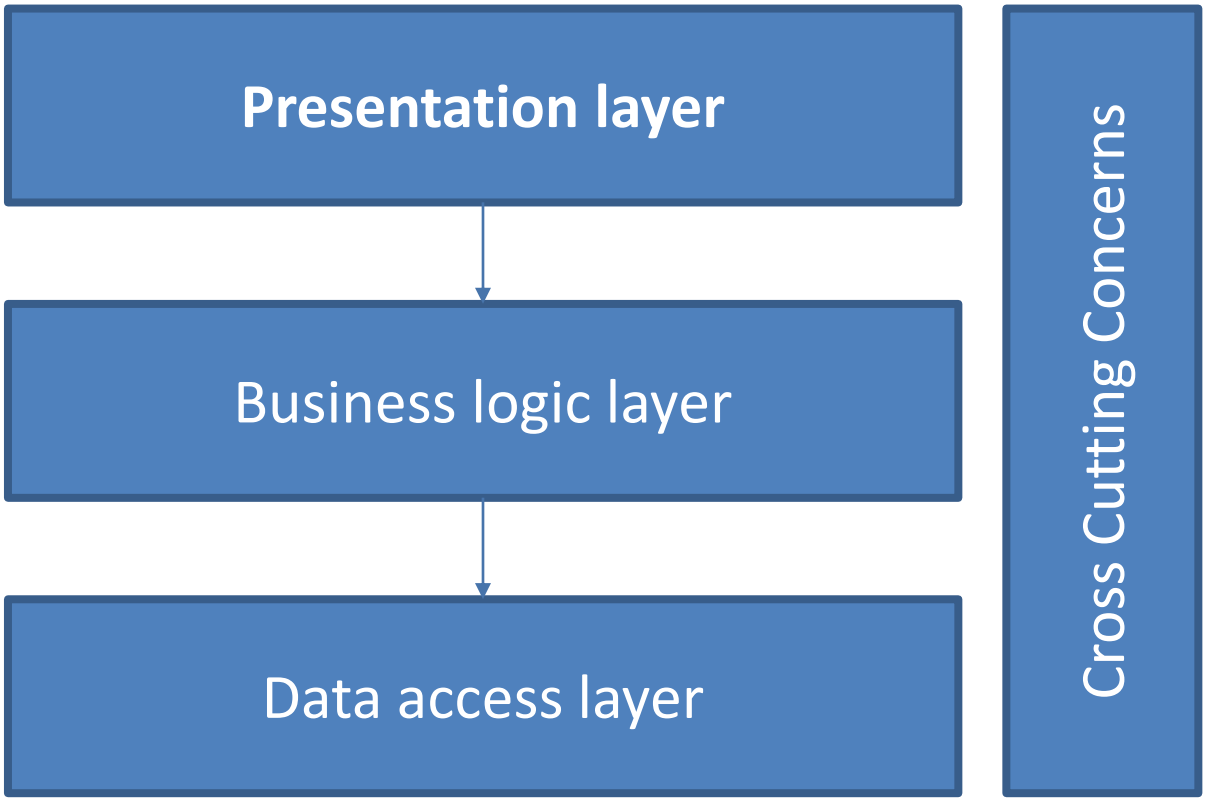
13 What is POCO & DTO?

ESSENSYS™ SOFTWARE SOLUTIONS

POCO - In software engineering, a **P**lain **O**ld **C**LR **O**bject is a simple object created in the Common Language Runtime (CLR) of the .NET Framework which is unencumbered by inheritance or attribute. It can contain state and some behavior.

DTO - A **D**ata **T**ransfer **O**bject is an object that carries data between processes; A DTO is a POCO subset. It can only contain state

What is N-Tier application?



Cross cutting concerns:

- Domain models
- Validation
- Exception handling
- Logging
- Common helpers

DEMO

Q & A

Exercitii:

- Download Visual Studio Community Edition de pe site-ul Microsoft
- Creati o solutie cu cel putin 2 proiecte: un proiect de consola si un class library referentiat in proiectul de consola
- Jucati-va cat mai mult: Clase, metode, constructori, statice/non-statice etc
- Sfat: Stack Overflow e cel mai bun prieten, dar intelegeti ce se intampla, nu copy/paste

ESSENSYS™
SOFTWARE SOLUTIONS

Quality software. On time. Every time.

Georgescu Claudiu, Dev Team Leader
www.essensys.ro

Microsoft GOLD CERTIFIED Partner

Data Management Solutions
Custom Development Solutions