

## ΑΝΑΛΥΣΗ PROJECT

### 1. Τι είναι ένα .NET C# Project

Ένα **.NET project** είναι ένας φάκελος που περιέχει:

- τον κώδικα της εφαρμογής
- τις ρυθμίσεις του project
- τις βιβλιοθήκες που χρησιμοποιούνται
- τα αρχεία που χρειάζονται για build και εκτέλεση

Με απλά λόγια:

**Project = όλη η δομή ενός προγράμματος .NET.**

---

### 2. Παράδειγμα Project που δημιουργήσαμε

Όταν γράψαμε:

```
dotnet new console -n Project1
```

δημιουργήθηκε κάτι σαν αυτό:

```
Project1
```

```
|
|— Program.cs
|— Project1.csproj
|— bin
|— obj
```

Ας τα δούμε ένα-ένα.

---

### 3. Το αρχείο Program.cs

Program.cs

Αυτό είναι το **κύριο αρχείο του προγράμματος**.

Εδώ ξεκινά η εκτέλεση.

Παράδειγμα:

```
Console.WriteLine("Hello World");
```

Ο compiler ξεκινά πάντα από την **Main method** ή το **top-level program**.

Στο .NET 6 γράφουμε απλά:

```
Console.WriteLine("Hello");
```

χωρίς class και main.

---

### 4. Το αρχείο Project1.csproj

Αυτό είναι το **αρχείο ρυθμίσεων του project**.

Παράδειγμα:

```
<Project Sdk="Microsoft.NET.Sdk">
```

```
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
```

```
  </PropertyGroup>
```

```
</Project>
```

#### Ανάλυση

#### Sdk

Microsoft.NET.Sdk

λέει στο .NET ποιο SDK χρησιμοποιείται.

---

#### OutputType

Exe

σημαίνει ότι το project παράγει:  
εκτελέσιμο πρόγραμμα

---

### TargetFramework

net6.0

λέει σε ποια έκδοση .NET τρέχει το πρόγραμμα.

---

### ImplicitUsings

επιτρέπει αυτόματα:

```
using System;
```

```
using System.Collections.Generic;
```

χωρίς να τα γράφουμε.

---

### Nullable

ενεργοποιεί τον έλεγχο null reference errors.

---

## 5. Ο φάκελος bin

bin

Εδώ μπαίνουν τα **compiled αρχεία**.

Παράδειγμα:

bin

└─ Release

    └─ net6.0

ή

bin

└─ Debug

Μέσα βρίσκουμε:

Project1.dll

Project1.exe

ή τα αρχεία publish.

---

## 6. Ο φάκελος obj

obj

Είναι **ενδιάμεσα αρχεία build**.

Τα χρησιμοποιεί ο compiler για:

- caching
- intermediate compilation
- metadata

Δεν τα επεξεργαζόμαστε ποτέ.

---

## 7. Η διαδικασία Build

Όταν γράφουμε:

```
dotnet run
```

γίνεται η εξής διαδικασία:

Source Code (.cs)

↓

Compiler (C#)

↓

Intermediate Language (IL)

↓

.NET Runtime

↓

Program Execution

---

## 8. Τι κάνει το .NET Runtime

Το runtime:

- φορτώνει το πρόγραμμα
- διαχειρίζεται μνήμη
- εκτελεί IL code
- κάνει garbage collection

Είναι ουσιαστικά η **μηχανή εκτέλεσης της εφαρμογής**.

---

### 9. Δημιουργία εκτελέσιμου

Όταν γράψαμε:

```
dotnet publish -c Release -r win-x64 -p:PublishSingleFile=true
```

το .NET δημιούργησε:

```
Project1.exe
```

στον φάκελο:

```
bin\Release\net6.0\win-x64\publish
```

Αυτό είναι το **τελικό πρόγραμμα**.

---

### 10. Δομή ενός μεγαλύτερου project

Σε μεγαλύτερα projects βλέπουμε κάτι σαν αυτό:

```
Project
```

```
|
|— Program.cs
|— Models
|— Controllers
|— Services
|— Data
|— Utilities
|— Project.csproj
```

Κάθε φάκελος έχει συγκεκριμένο ρόλο.

---

### 11. Πλεονεκτήματα του Project System

Τα projects επιτρέπουν:

- οργάνωση κώδικα
  - διαχείριση βιβλιοθηκών
  - build automation
  - debugging
  - δημιουργία executable
-