



Microsoft
.NET



Residência
em Software

POO - Dynamic e Object

Prof. Hélder Almeida



INSTITUIÇÃO EXECUTORA



UESC

COORDENADORA



APOIO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Dynamic e Object

Em C#, dynamic e object são ambos tipos que podem ser usados para representar dados cujo tipo exato é desconhecido em tempo de compilação. Aqui estão algumas semelhanças entre dynamic e object:

- **Flexibilidade de Tipo:**
 - Ambos permitem que você armazene dados de diferentes tipos na mesma variável.
 - São úteis quando o tipo exato dos dados só é conhecido em tempo de execução.
- **Compatibilidade com Métodos e Propriedades Dinâmicas:**
 - Ambos suportam acesso a membros (métodos, propriedades, campos) de forma dinâmica, sem a necessidade de conversões explícitas.
- **Uso em Cenários Dinâmicos:**
 - São frequentemente utilizados em situações em que a estrutura dos dados só é conhecida em tempo de execução, como interoperação com linguagens dinâmicas ou manipulação de dados dinâmicos.

Dynamic e Object

```
// Usando dynamic
dynamic dynamicVar = 10;
dynamicVar = "Olá, mundo!";
// Sem erro de compilação, mesmo sendo uma string
Console.WriteLine(dynamicVar.Length);

// Usando object
object objectVar = 42;
objectVar = "Hello, world!";
// Sem erro de compilação, mas precisa de conversão
// explícita para chamar métodos específicos
Console.WriteLine(objectVar.ToString());
```

Dynamic: Tempo de Execução

- O que é Dynamic?
 - O tipo dynamic em C# permite que os desenvolvedores trabalhem com objetos cujo tipo é determinado em tempo de execução.
 - Diferente de **var**, que é resolvido em tempo de compilação, dynamic é resolvido em tempo de execução.
- Quando Usar Dynamic?
 - Utilizado para interagir com APIs dinâmicas, como JavaScript.
 - Útil em situações onde o tipo exato só é conhecido durante a execução.

```
// Usando dynamic
dynamic dynamicVar = 10;
dynamicVar = "Olá, mundo!";
// Sem erro de compilação, mesmo sendo uma string
Console.WriteLine(dynamicVar.Length);
```

Object em C#: A Base de Todos os Tipos

- O que é Object?
 - O tipo object é a base de todos os tipos em C#.
 - Tudo em C# é implicitamente derivado do tipo object.
- Uso Ponderado de Object:
 - Apesar de ser a base, o uso indiscriminado de object pode levar à perda de informações.
 - Mais restrito, pois operações precisam ser convertidas explicitamente.

```
// Usando object
object objectVar = 42;
objectVar = "Hello, world!";
// Sem erro de compilação, mas precisa de conversão
// explícita para chamar métodos específicos
Console.WriteLine(objectVar.ToString());
```

Dynamic vs. Object: Diferenças Cruciais

- **Dynamic Oferece Flexibilidade:**
 - Permite operações dinâmicas em tempo de execução.
 - Não exige conversões explícitas.
- **Object Requer Conversões Explícitas:**
 - Restrito, pois operações precisam ser convertidas explicitamente.
 - Mais utilizado quando a informação específica do tipo é necessária.

```
dynamic dynamicVar = 10;
```

```
object objectVar = 10;
```

```
dynamicVar = dynamicVar + 5; // Operação dinâmica permitida
```

```
// objectVar = objectVar + 5; // Erro de compilação, precisa de conversão
```

Escolhendo a Ferramenta Certa para a Tarefa

- **Dynamic e Object têm Papéis Distintos:**
 - Use dynamic para flexibilidade em tempo de execução.
 - Use object quando informações específicas do tipo são necessárias.
- **Compreendendo as Diferenças é Fundamental:**
 - Escolha sabiamente entre dynamic e object dependendo dos requisitos da sua aplicação.
- **Prática Leva à Maestria:**
 - Ganhe experiência no uso de dynamic e object em cenários do mundo real.

Exercício

Crie uma classe chamada Calculadora que possui métodos para realizar operações básicas como soma, subtração, multiplicação e divisão. Permita que esses métodos aceitem parâmetros dinâmicos usando `dynamic` ou `object`. Implemente um programa principal que utiliza a classe Calculadora para realizar operações com diferentes tipos de dados (inteiros, doubles, strings) sem a necessidade de conversão prévia.