Microsoft .NET

Residência em Software

Propriedades

Prof. Hélder Almeida

INSTITUIÇÃO EXECUTORA
cepedi UESC

COORDENADORA
MCTI FUTURO Softex

APOIO
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Residência em Software

O que são Properties?

Properties são membros de classe que oferecem uma maneira de ler ou escrever valores em campos privados.

São essenciais para a implementação do encapsulamento em C#.

Alternativa *elegante* para acessar atributos privados.

https://learn.microsoft.com/pt-br/dotnet/csharp/programming-guide/classes-and-structs/properties

Residência em Software

Vantagens do Uso de Properties

Acesso Controlado: Proporciona controle sobre a leitura e escrita de dados.

Encapsulamento: Facilita o esconder de detalhes de implementação.

Maior Flexibilidade: Permite validações e lógica personalizada durante a leitura ou escrita.

Residência em Software

Exemplo

```
public class Exemplo{
    2 references
    private int _idade;

    0 references
    public int Idade{
        get { return _idade; }
        set { _idade = value; }
    }
}
```

Residência em Software

Propriedade automática

Simplifica a sintaxe quando não é necessário lógica personalizada no **get** ou **set**.

```
0 references
public class Exemplo
{
    0 references
    public int Idade { get; set; }
}
```

Residência em Software

Propriedades Somente Leitura e Somente Escrita

=> é usado para propriedades somente leitura (expression-bodied property).

```
0 references
public class Exemplo
{
    // Somente Leitura
    0 references
    public int AnoAtual => DateTime.Now.Year;

    // Somente Escrita
    0 references
    public string Mensagem { set { Console.WriteLine(value); } }
}
```

Residência em Software

Propriedades com Lógica Personalizada

Permite a adição de validações e lógica personalizada no set.

```
0 references
public class Pessoa
{
    2 references
    private int _idade;

    0 references
    public int Idade
    {
        get { return _idade; }
        set
        {
            if (value > 0)
                _idade = value;
            else
                throw new ArgumentException("Idade deve ser maior que zero.");
        }
    }
}
```

Residência em Software

Resumo

Properties são fundamentais para o encapsulamento em C#.

Proporcionam controle sobre a leitura e escrita de dados.

Permitem a adição de lógica personalizada durante essas operações.

Residência em Software

Exercícios

1. Crie uma classe chamada **Veiculo** com propriedades para representar características básicas de um veículo, como **Modelo**, **Ano**, e **Cor**. Implemente um programa principal que cria uma instância da classe **Veiculo**, define valores para suas propriedades e exibe essas informações no console.

2. Modifique a classe **Veiculo** do exercício anterior para incluir uma propriedade somente leitura chamada **IdadeVeiculo**. Esta propriedade deve calcular a idade do veículo com base no ano atual e no ano do veículo.

3. Crie uma classe **ContaBancaria** com uma propriedade **Saldo**. Adicione lógica personalizada na propriedade **Saldo** para garantir que o saldo não seja negativo. Se uma tentativa de definir um saldo negativo for detectada, lance uma exceção **ArgumentException** com a mensagem "Saldo não pode ser negativo".

4. Crie uma classe **Aluno** com propriedades automáticas para representar o nome e a idade de um aluno. No construtor da classe, inicialize essas propriedades com valores padrão. Implemente um programa principal que cria uma instância da classe **Aluno** sem fornecer valores iniciais e exibe as propriedades no console.

5. Crie uma classe **Agenda** que represente uma lista de contatos. Adicione uma propriedade indexada chamada **Contatos** que permite acessar os contatos pelo índice. Implemente um programa principal que cria uma instância da classe **Agenda**, adiciona alguns contatos e exibe suas informações usando a propriedade indexada.