Fundamentos de Desenvolvimento com C#

Aula 06: Classes

Professor: Rinaldo Ferreira Junior

E-mail: rinaldo.fjunior@prof.infnet.edu.br

- **Professor:** Rinaldo Ferreira Junior
- **Graduação:** Pós-graduado em Arquitetura de Softwares
- **Atuação:** .Net | C# | SQL | NoSQL | Engenheiro de Software
- **E-mail:** rinaldo.fjunior@prof.infnet.edu.br
- **Linkedin:** https://www.linkedin.com/in/rinaldo-ferreira-junior-787326a

Aula 06: Roteiro

- Classes
- Objetos

Aula 06: Classes

- Classes funcionam como templates ou protótipos para a criação de objetos
    - Classes e objetos são diferentes, embora os termos apareçam de forma intercambiável
- Classes possuem atributos e comportamento relacionado
- Classes são reference type.
- Objetos são instâncias de uma classe. Possuem dados reais amarrados aos comportamentos definidos na sua classe de origem.

Aula 06: Criando Classes

- Na estrutura de projeto, normalmente cada classe está em seu próprio arquivo .cs
- Clique o botão direito do mouse no local onde irá colocar a classe e selecione Adicionar -> Classe. Informe o nome da classe.
- Classes são criadas com a instrução class.

```
public class Person
{

}
```

Aula 06: Membros da Classes

- Os membros de uma classe são todos os atributos, métodos, constantes, propriedades e eventos de uma classe.
- Atributos (fields) são como variáveis, porém no escopo de uma classe. Esses atributos possuem modificadores de acesso, para definer o grau de visibilidade do atributo.
- Atributos são read-write por padrão
- Propriedades são métodos que agem como um atributo, mas permitem um maior controle sobre seu comportamento.

```
public class Person
{
    public string Nome = string.Empty;

    0 references
    public DateTime Nascimento { get; set; }
}
```

Aula 06: Membros da Classes

- É possível expandir uma propriedade, de forma que seja possível customizar os comportamentos de leitura e gravação da propriedade

```
public class Person
{
    public string Nome = string.Empty;

    0 references
    public DateTime Nascimento { get; set; }

    private string _endereco = string.Empty;
    0 references
    public string Endereco
    {
        get
        {
            return _endereco;
        }
        set
        {
            _endereco = value;
        }
    }
}
```

```
public class Address
{
    2 references
    public string Logradouro { get; set; } = string.Empty;

    1 reference
    public string Complemento { get; set; } = string.Empty;

    2 references
    public string Bairro { get; set; } = string.Empty;

    0 references
    public string Pais { get; } = "Brasil";

    1 reference
    public int Numero { get; set; }

    0 references
    public bool SemNumero
    {
        get
        {
            return this.Numero <= 0;
        }
    }
}
```

Aula 06: Objetos (Instâncias)

Instituto Infnet

- A criação de uma instância se dá por uma variável, do tipo da classe com a qual se quer trabalhar:

```
static void Main(string[] args)
{
    Person pessoa = new Person();
    pessoa.Nome = "Roberval Firmino";
    pessoa.Endereco = "Rua de Cima, nº 1";
    pessoa.Nascimento = new DateTime(1965, 04, 04);

    Console.WriteLine($"Nome: {pessoa.Nome}");
    Console.WriteLine($"Endereco: {pessoa.Endereco}");
    Console.WriteLine($"Nascimento: {pessoa.Nascimento:d}");
    Console.WriteLine();

    Person pessoa2 = new()
    {
        Nome = "Clarisbela de Jesus",
        Endereco = "Rua de Baixo, nº 2",
        Nascimento = new DateTime(1970, 02, 28)
    };

    Console.WriteLine($"Nome: {pessoa2.Nome}");
    Console.WriteLine($"Endereco: {pessoa2.Endereco}");
    Console.WriteLine($"Nascimento: {pessoa2.Nascimento:d}");

    Console.Read();
}
```

Aula 06: Objetos (Instâncias)

Instituto Infnet

- Propriedades podem fazer referências à outras instâncias

```
public class Person
{
    public string Nome = string.Empty;

    2 references
    public DateTime Nascimento { get; set; }

    3 references
    public Address? Endereco { get; set; }

}
```

```
static void Main(string[] args)
{
    Address endereco = new Address();

    endereco.Logradouro = "Rua Blah";
    endereco.Complemento = "Bloco 1";
    endereco.Bairro = "Centro";

    Person pessoa = new Person();
    pessoa.Nome = "Roberval Firmino";
    pessoa.Endereco = endereco;
    pessoa.Nascimento = new DateTime(1965, 04, 04);

    Console.WriteLine($"Nome: {pessoa.Nome}");
    Console.WriteLine($"Logradouro: {pessoa.Endereco.Logradouro} | Bairro: {pessoa.Endereco.Bairro}");
    Console.WriteLine($"Nascimento: {pessoa.Nascimento:d}");
    Console.WriteLine();

    Console.Read();
}
```