



LINGUAGEM DE PROGRAMAÇÃO

Orientada a objeto

Roteiro da aula

- Herança
- Polimorfismo
- Outros tipos de Classes e metodos
- Sobrecarga de metodos entre classes
- Exemplos

Orientação a Objeto

- Um dos grandes diferenciais está no conceito de **herança**
→ mecanismo através do qual definições existentes podem ser facilmente estendidas.
- E também no **polimorfismo**
→ que permite selecionar funcionalidades que um programa irá utilizar de forma dinâmica.

Herança

- Ferramenta para evitar a repetição ao definir classes diferentes que tem várias características em comum e estão muito claramente relacionadas entre si.
- Uma classe pode derivar no máximo de uma outra classe
- A sintaxe para declarar que uma classe herda de outra:

```
class PessoaFisica: Pessoa {  
    .....  
}
```

Exemplo: A classe PessoaFisica herda da classe Pessoa

Polimorfismo

- Sua principal função é tornar possível criar métodos de mesmo nome em classes diferentes
- Posso sobrepor os métodos usando o operador **virtual** ou **abstrato** no método da classe pai e **override** no método da classe filho
- Ou posso fazer com que os dois métodos com o mesmo nome não sobreponha, referenciando a classe pai na classe filho com: **base.nomeMetodo(parametros)**

Polimorfismo – sobreposição de métodos

```
class Pessoa
```

```
{  
    public virtual void gravar()  
    {  
        Console.WriteLine("Gravar Pessoa ");  
    }  
}
```

```
class pFisica : Pessoa
```

```
{  
    public override void gravar()  
    {  
        Console.WriteLine("Gravar Pessoa Fisica ");  
    }  
}
```

Polimorfismo – sem sobreposição

class Pessoa

```
{ public string nome, endereco;  
    public void gravar(string  
nome, string endereco)  
    {  
        this.nome = nome;  
        this.endereco = endereco;  
    }  
    public string getNome()...  
    public string getEndereco()...  
}
```

class pFisica : Pessoa

```
{ public string cpf;  
    public int idade;  
    public void gravar(string nome,  
string endereco, string cpf, int idade )  
    {  
        base.gravar(nome, endereco);  
        this.cpf = cpf;  
        this.idade = idade;  
    }  
  
    public string getCpf() ...  
    public int getIdade() ...
```

Classes ou metodos

- **Sealed** → classes ou métodos selados não podem ser herdados
sealed class pessoa
- **Abstract** → classes ou métodos só podem ser herdados, mas não podem ser instanciados
abstract class pai

Classes Abstrata

- As classes abstratas não permitem realizar instância.
- São classes feitas especialmente para serem modelos para suas classes derivadas.
- As classes derivadas, deverão sobrescrever os métodos para realizar a implementação dos mesmos.
- Caso um ou mais métodos abstratos estejam presentes nessa classe abstrata, a classe filha será, então, forçada a definir tais métodos, pois, caso contrário, a classe filha também se tornará abstrata.

Métodos Abstrato

- A funcionalidade dos métodos abstratos que são herdados pelas classes filha é atribuída de acordo com o objetivo dessas classes.
- É possível, não atribuímos uma funcionalidade a esses métodos abstratos. Neste caso, faz-se necessário, pelo menos, declarar tais métodos.
- Os métodos abstratos estão presentes somente em classes abstratas, e são aqueles que não possuem implementação.
- A sintaxe deste tipo de método é a seguinte:
 <modificador_de_acesso> abstract <tipo_retorno> <identificador> ;
 public abstract double Ganhos();

exemplo

- Sistema de folha de pagamento
- Pratica Aula 9