





Alex Sander Resende de Deus

A 25 anos ensinando programação a jovens e adultos.

Apaixonado por tecnologia é atualmente coordenador de cursos na ETEC Albert Einstein. Na FIAP atua como professor da FIAP School, lecionando C#, SQLServer e Desenvolvimento Mobile

AULA 8

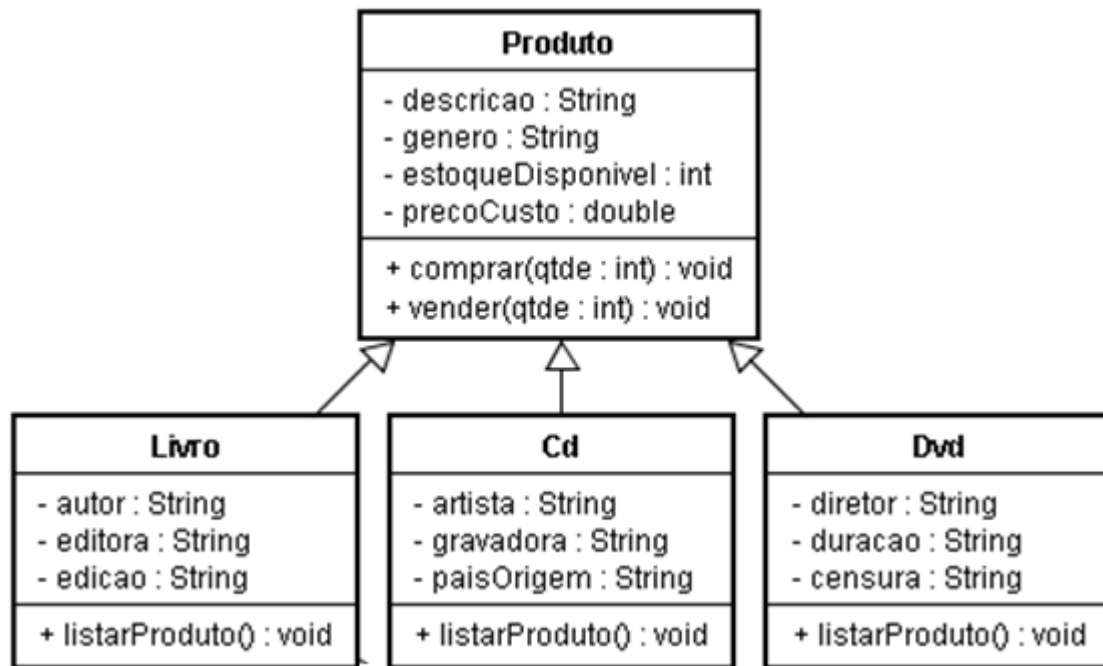
Herança



Herança

- É um princípio da Programação Orientada a Objetos que permite que as classes compartilhem atributos e métodos comuns baseados em um relacionamento.
- A herança possibilita a aplicação de vários conceitos de orientação a objetos que não seriam viáveis sem a organização e estruturação alcançada por sua utilização.

Projeto: Livraria



Classe Produto

```
class Produto
{
    public string descricao { get; set; }
    public string genero { get; set; }
    public int estoqueDisponivel { get; set; }
    public double precoCusto { get; set; }

    public Produto()
    {
        this.descricao = "";
        this.genero = "";
        this.estoqueDisponivel = 0;
        this.precoCusto = 0;
    }
}
```

Classe Produto

```
public void comprar(int qtde)
{
    estoqueDisponivel += qtde;
    MessageBox.Show("Quatidade adicionada ao estoque");
}

public void vender(int qtde)
{
    if (qtde <= this.estoqueDisponivel)
    {
        estoqueDisponivel -= qtde;
        MessageBox.Show("Quantidade retirada do estoque");
    }
    else
    {
        MessageBox.Show("Estoque indisponível");
    }
}
```

Classe Livro

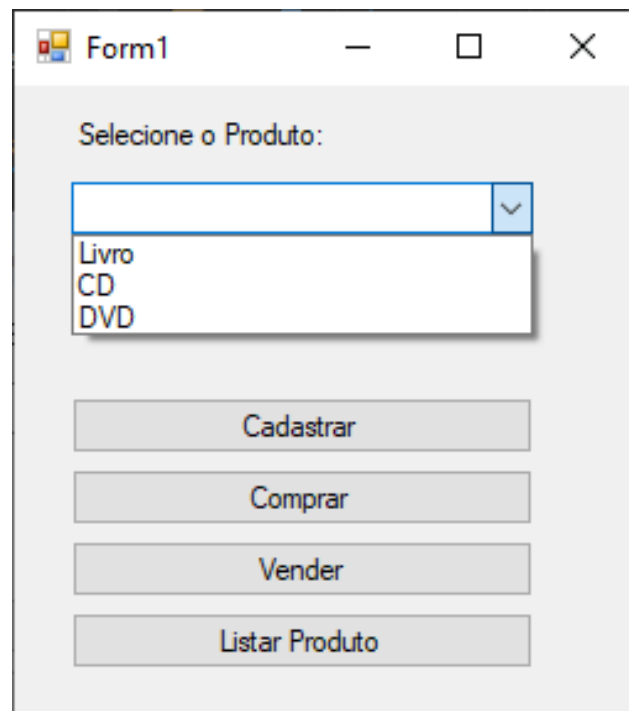
```
class Livro: Produto
{
    public string autor { get; set; }
    public string editora { get; set; }
    public string edicao { get; set; }

    public Livro()
    {
        this.descricao = "";
        this.genero = "";
        this.estoqueDisponivel = 0;
        this.precoCusto = 0;
        this.autor = "";
        this.editora = "";
        this.edicao = "";
    }
}
```


Classe Livro

```
public void listarProduto()
{
    MessageBox.Show("Dados do Livro:" +
        "\n\nDescrição: " + this.descricao +
        "\nGênero: " + this.genero +
        "\nAutor: " + this.autor +
        "\nEditora: " + this.editora +
        "\nEdição:" + this.edicao +
        "\nPreço de Custo: " + this.precoCusto.ToString() +
        "\nEstoque Disponível: " + this.estoqueDisponivel.ToString());
}
```

Formulário



A screenshot of a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area has a light gray background. At the top, the text "Selecione o Produto:" is displayed. Below it is a dropdown menu with a blue border and a blue arrow button on the right. The dropdown is open, showing a list of three items: "Livro", "CD", and "DVD". Below the dropdown are four buttons stacked vertically, each with a light gray background and a thin border. The buttons are labeled "Cadastrar", "Comprar", "Vender", and "Listar Produto" from top to bottom.

Código do botão cadastrar

```
private void btnCadastrar_Click(object sender, EventArgs e)
{
    switch (cmbProduto.SelectedIndex)
    {
        case 0:

            livro.descricao = Interaction.InputBox("Digite o título do livro");
            livro.genero = Interaction.InputBox("Digite o gênero do livro");
            livro.autor = Interaction.InputBox("Digite o autor do livro");
            livro.editora = Interaction.InputBox("Digite a editora do livro");
            livro.edicao = Interaction.InputBox("Digite a edição do livro");
            livro.precoCusto = Convert.ToDouble(Interaction.InputBox("Digite o preço do livro"));
            MessageBox.Show("Livro cadastrado com sucesso!");
            break;

        case 1:
            //CD
            break;

        case 2:
            //DVD
            break;
    }
}
```

Código do botão comprar

```
private void btnComprar_Click(object sender, EventArgs e)
{
    switch (cmbProduto.SelectedIndex)
    {
        case 0:
            int qtde = Convert.ToInt32(Interaction.InputBox("Digite a qtde"));
            livro.comprar(qtde);
            break;
        case 1:
            //CD
            break;
        case 2:
            //DVD
            break;
    }
}
```

Código do botão vender

```
private void btnVender_Click(object sender, EventArgs e)
{
    switch (cmbProduto.SelectedIndex)
    {
        case 0:
            int qtde = Convert.ToInt32(Interaction.InputBox("Digite a qtde"));
            livro.vender(qtde);
            break;
        case 1:
            //CD
            break;
        case 2:
            //DVD
            break;
    }
}
```

Código do botão listar produto

```
private void btnListar_Click(object sender, EventArgs e)
{
    switch (cmbProduto.SelectedIndex)
    {
        case 0:
            livro.listarProduto();
            break;
        case 1:
            //CD
            break;
        case 2:
            //DVD
            break;
    }
}
```



Momento Hands On



Que tal terminar nosso projeto livraria de forma que agora possamos cadastrar, comprar e vender CDs e DVDs?

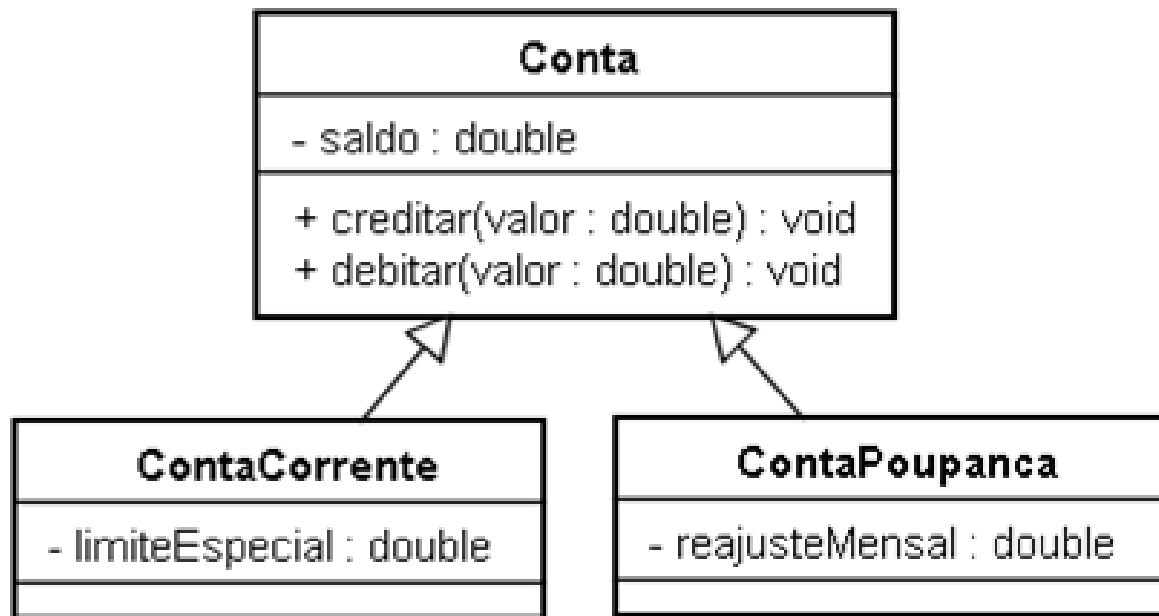
Então, HANDS ON!



Para treinar mais



Exercício 01



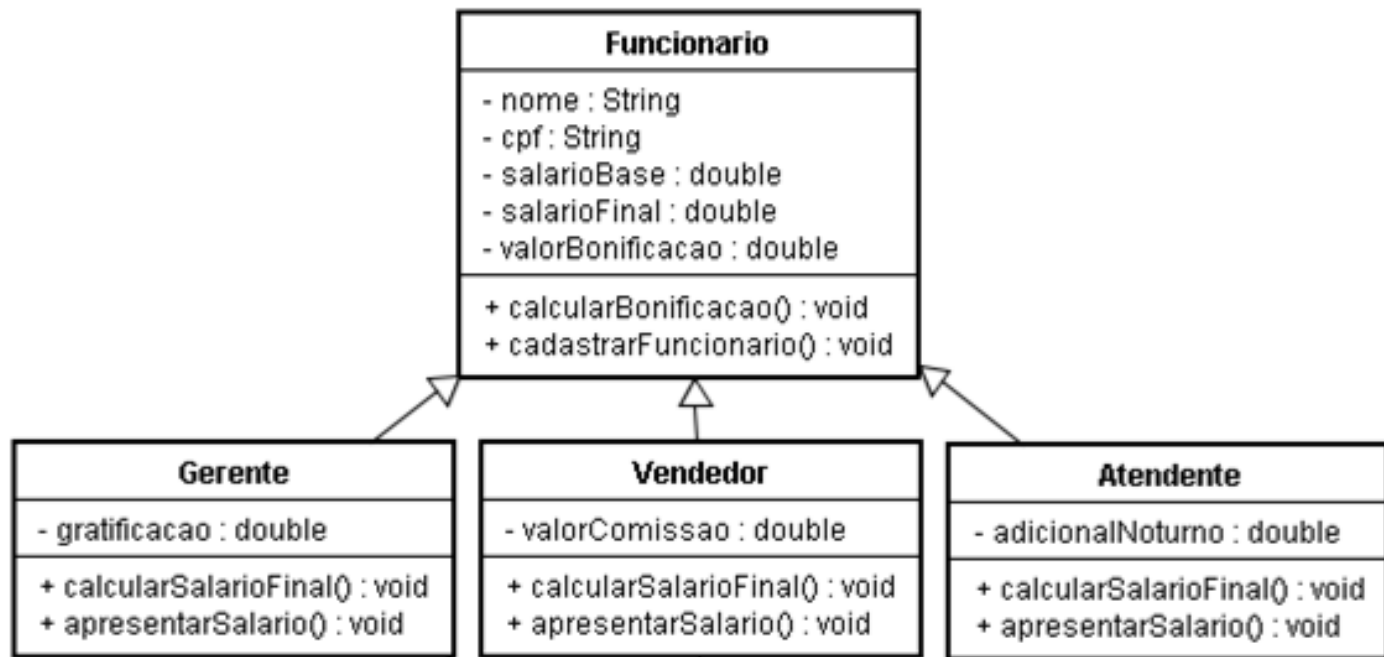
Classe: Conta	
Métodos	creditar: Recebe um valor por parâmetro e soma ao atributo saldo
	debitar: Recebe um valor por parâmetro e subtrai do atributo saldo desde que haja saldo suficiente, caso não tenha, apresenta mensagem de saldo insuficiente.

Classe: ContaCorrente (subclasse de Conta)
Obs.: O atributo limiteEspecial será <u>implementado</u> mas não será utilizado por enquanto.

Classe: ContaPoupanca (subclasse de Conta)
Obs.: O atributo reajusteMensal será <u>implementado</u> mas não será utilizado por enquanto.

Formulário	
	<ul style="list-style-type: none"> Instanciar um objeto do tipo ContaCorrente chamado cc1 com saldo inicial de 500 e limite especial de 1000.
	<ul style="list-style-type: none"> Instanciar um objeto do tipo ContaPoupanca chamado cp1 com saldo inicial de 5000 e reajusteMensal de 1% (0.01)
	<ul style="list-style-type: none"> Apresentar um menu com as opções: <ul style="list-style-type: none"> 1 – Conta corrente 2 – Poupanca 0 – Sair
	<ul style="list-style-type: none"> Apresentar outro menu com as opções: <ul style="list-style-type: none"> 1 – Depositar 2 – Sacar 3 – Consultar saldo 0 – Sair
	<ul style="list-style-type: none"> Realizar as chamadas aos métodos de acordo com as opções do usuário.

Exercício 02



Classe: Funcionario

Métodos	cadastrarFuncionario: Lê o nome, CPF e salário base e armazena em seus respectivos atributo. Inicializa com zero os atributo salarioFinal e valorBonificacao.
	calcularBonificacao: Calcula 5% do salário base do funcionário e insere no atributo valorBonificacao através do métodos set. Cálculo: $\text{bonificacao} = \text{salário base} * 0.05$

Subclasse: Gerente

Métodos	calcularSalarioFinal: Lê o valor da gratificação e armazena no atributo gratificacao. Soma o atributo gratificação, o salário base e a bonificação e armazena no atributo salarioFinal.
	apresentarSalario: Apresentar o nome do funcionário, o valor do salário base, o valor da bonificação, o valor da gratificação e o valor do salário final.

Subclasse: Vendedor

Métodos	calcularSalarioFinal: Lê o valor da comissão e armazena no atributo valorComissao. Soma o atributo valorComissao, o salário base e a bonificação e armazena no atributo salarioFinal.
	apresentarSalario: Apresentar o nome do funcionário, o valor do salário base, o valor da bonificação, o valor da comissão e o valor do salário final.

Subclasse: Atendente

Métodos	calcularSalarioFinal: Lê o valor do adicional noturno e armazena no atributo adicionalNoturno. Soma o atributo adicionalNoturno, o salário base e a bonificação e armazena no atributo salarioFinal.
	apresentarSalario: Apresentar o nome do funcionário, o valor do salário base, o valor da bonificação, o valor do adicional noturno e o valor do salário final.

Formulário	
	• Instanciar um objeto do tipo Gerente chamado gerente.
	• Instanciar um objeto do tipo Vendedor chamado vendedor.
	• Instanciar um objeto do tipo Atendente chamado atendente.
	• Apresentar um menu com as opções: 1 – Gerente 2 – Vendedor 3 – Atendente 0 – Sair
	• Apresentar um menu com as opções: 1 – Cadastrar <u>funcionario</u> 2 – Calcular salário 3 – Calcular bonificação 4 – Apresentar salário final 0 – Sair

OBRIGADO



profalex.deus@fiap.com.br



[linkedin.com/in/alexanderresende](https://www.linkedin.com/in/alexanderresende)

FIAP MBA⁺

Copyright © 2019 | Professor (a) Nome do Professor

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

FIAP