

Apostila Interativa - Programação Orientada a Objetos com C#

Unidade 1 - Introdução à POO

A Programação Orientada a Objetos (POO) é baseada em objetos que simulam o mundo real. Cada objeto tem atributos (características) e métodos (comportamentos).

```
public class Pessoa {  
    public string nome;  
    public int idade;  
  
    public void Apresentar() {  
        Console.WriteLine($"Olá, meu nome é {nome} e tenho {idade} anos.");  
    }  
}
```

Unidade 2 - Estrutura de uma Classe

Uma classe define os atributos e métodos que seus objetos terão. Pode conter construtores, que ajudam a inicializar os dados.

```
public class Produto {  
    public string nome;  
    public double preco;  
  
    public Produto(string nome, double preco) {  
        this.nome = nome;  
        this.preco = preco;  
    }  
  
    public void Exibir() {  
        Console.WriteLine($"{nome} custa R${preco}");  
    }  
}
```

Unidade 3 - Encapsulamento

Encapsulamento protege os dados da classe e permite acesso controlado por meio de métodos públicos.

```
public class Conta {  
    private double saldo;
```

Apostila Interativa - Programação Orientada a Objetos com C#

```
public void Depositar(double valor) {  
    if (valor > 0) saldo += valor;  
}  
  
public double ConsultarSaldo() {  
    return saldo;  
}  
}
```

Unidade 4 - Herança

Herança permite criar novas classes baseadas em uma classe existente.

```
public class Animal {  
    public void Dormir() {  
        Console.WriteLine("Zzz...");  
    }  
}  
  
public class Gato : Animal {  
    public void Miar() {  
        Console.WriteLine("Miau!");  
    }  
}
```

Unidade 5 - Polimorfismo

Polimorfismo permite que métodos com o mesmo nome se comportem de maneiras diferentes.

```
public class Animal {  
    public virtual void Falar() {  
        Console.WriteLine("Som genérico");  
    }  
}  
  
public class Cachorro : Animal {  
    public override void Falar() {  
        Console.WriteLine("Au au!");  
    }  
}
```

Apostila Interativa - Programação Orientada a Objetos com C#

Unidade 6 - Abstração

Abstração mostra apenas os detalhes relevantes de um objeto, escondendo a complexidade.

```
public abstract class CartaoDeCredito {  
    public abstract void Pagar(double valor);  
}  
  
public class Visa : CartaoDeCredito {  
    public override void Pagar(double valor) {  
        Console.WriteLine($"Pagamento de R${valor} com Visa.");  
    }  
}
```

Exercícios Propostos

1. Crie uma classe Aluno com atributos nome, nota1, nota2 e um método CalcularMedia().
2. Crie uma herança com Funcionario e Gerente, onde o gerente tem um método CalcularBonus().