

PROGRAMAÇÃO ORIENTADA A OBJETOS

**Paradigmas de Programação e
Introdução ao Paradigma
Orientado a Objetos**

ROTEIRO

- **Paradigmas de Programação**
- **Programação Orientada a Objetos (POO)**
- **A importância da POO**
- **Realidade x Modelo**
- **Conceitos-Chave da POO**

Paradigmas de Programação

- Trata-se de uma forma de classificar as linguagens de programação baseada em suas funcionalidades. Um paradigma fornece e determina a visão que o programador possui sobre a estruturação e execução do programa.

• Alguns exemplos de paradigmas de programação

- **Programação Funcional:** Programadores abstraem o programa como uma sequência de funções executadas de modo empilhado. Trata da computação como uma avaliação de funções matemáticas, evitando estados ou dados mutáveis
- **Programação declarativa:** Programador modela o problema por meio de assertivas em relação aos objetos do universo em questão: Exemplo: SQL e HTML

Paradigmas de Programação

• Alguns exemplos de paradigmas de programação

- **Programação Procedural:** É baseado no conceito de chamadas a procedimentos, conhecidos como rotinas, sub-rotinas, métodos. Um conjunto de passos computacionais a serem executados. Procedimentos podem ser chamados a qualquer momento durante a execução do programa, inclusive por outros procedimentos.
- **Programação Orientada a Objetos:** Muito conhecido pela sigla POO, é um modelo de análise, projeto e programação de software, baseado na composição e interação entre várias unidades conhecidas como objetos. **É neste paradigma que vamos nos concentrar nesta disciplina.**

POO – Programação Orientada a Objetos

- Neste paradigma de programação todos os componentes de um programa são considerados objetos
 - Todos os objetos possuem estado, identidade, comportamento
- A principal mensagem que a POO nos fornece é:
 - Reduzir os programas em partes resulta em metas mais fáceis de se alcançar
 - Os programas são basicamente interações entre objetos, o que faz com que surjam padrões comuns
 - *O que é um padrão comum, pode ser reutilizado*

A importância da POO

- Escrever código não é tão complicado...
 - **Dificuldade de compreender o seu funcionamento, em especial quando há centenas de milhares ou até milhões de linhas de código**
- Facilidade no entendimento está associado a:
 - **Projetar o código**
 - **Preocupação com organização**
- Exemplos
 - Em linguagens diferentes da POO era comum o comando goto
 - bloco de comandos;
 -
 - goto saida;
 - saida: comandos

A importância da POO

- O comando goto realiza o desvio da execução para o comando que possui o rótulo indicado.
- Qualquer boa técnica de programação atual não recomenda o seu uso.
 - Em linguagens não orientadas a objetos, seu uso é recomendado apenas para processar exceções, desviando para uma área específica caso ocorra algum erro grave na execução de algum comando.

A importância da POO

- Exemplo de programa com goto

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int a;
```

```
    int b;
```

```
    while (1)
```

```
    {
```

```
        printf("Valores: ")
```

```
        scanf("%d %d", &a);
```

```
        if(a == 0)
```

```
        {
```

```
            goto fim;
```

```
        }
```


A importância da POO

Continua do slide anterior...

- ***Fim:***
 - ***printf("Fim da execução do programa\n");***

.Usar goto torna o projeto ruim, e dificulta a sua manutenção no futuro, pois códigos tornam-se complexos.

- ***A análise de problemas ficará prejudicada***

Realidade x Modelo

- Quando desenvolvemos um código temos que ter em mente a abstração possível em relação ao mundo real
- Chamamos de *gap semântico* a diferença entre o quão completo um modelo representa a realidade e a realidade em si.
- Quanto menor o gap, será mais fácil de entender e modificar um sistema
- Como todo programa de computador representa objetos reais, entender o gap é muito importante

Conceitos-Chave da POO

- **Abstração**

- É um conceito no qual escondemos detalhes não essenciais, embora mostramos o que é mais relevante
- Apresenta diversas camadas de abstração, dependendo do que é relevante.
- **Exemplo: Um carro**
 - O carro possui: motor, câmbio, volante, sistema multimídia, pneus, acelerador, freios, etc.
 - O que realmente importa para o usuário?
 - Saber virar o volante ou saber como é o processo de funcionamento da troca de marchas entre a embreagem e o restante do motor?

Conceitos-Chave da POO

- ***Encapsulamento***
 - É um conceito em que a parte interna de um código não fica acessível para outras partes, com o intuito de proteger e manter a sua integridade.
 - Todas as operações a serem feitas pelo usuários devem ser feitas por uma interface bem definida
 - Apresenta diversas camadas de abstração, dependendo do que é relevante.
 - **Exemplo: Carro**

Conceitos-Chave da POO

- O carro possui um motor cujo funcionamento e a interação interna entre as peças fiquem inacessíveis ao motorista
- Uma interface mais intuitiva torna o uso do carro mais simples.
- O motorista não precisa entender de mecânica para dirigir um carro, mas precisa ter acesso a algumas interfaces disponibilizadas pelo carro para que ele possa acelerar, frear e trocar as marchas (carro manual)

Referências

1. **Java Como Programar: Paul Deitel & Harvey Deitel - 10ª Edição**
2. **Java Como Programar: Paul Deitel & Harvey Deitel - 8ª Edição**
3. **Wikipedia – Paradigma de Programação - https://pt.wikipedia.org/wiki/Paradigma_de_programação**
4. **Wikipedia – Java (Plataforma de Software) - [https://pt.wikipedia.org/wiki/Java_\(plataforma de software\)](https://pt.wikipedia.org/wiki/Java_(plataforma_de_software))**

PROGRAMAÇÃO ORIENTADA A OBJETOS

**Paradigmas de Programação e
Introdução ao Paradigma
Orientado a Objetos**