

# Programação Orientada a Objeto

---

## OO2 - Abstração

### Sumário

- Vantagens da OO
- Abstração
- Classes e Objetos
- Atributos e Métodos

Dr. Márcia L. Aguená Castro  
[marciaaguená@gmail.com](mailto:marciaaguená@gmail.com)

# Vantagens da OO

---

## Sumário

- Vantagens da OO
- Abstração
- Classes e Objetos
- Atributos e Métodos

## Vantagens da OO

---

- Facilita o mapeamento entre problema do mundo real e o código fonte, já que o mundo real também é composto por objetos que interagem entre si;
- A orientação a objetos aproxima a documentação e o código fonte de um sistema;
- Por exemplo, um diagrama de classes é **mapeado diretamente** para código fonte.

## Vantagens da OO

---

- Os princípios da orientação a objeto facilitam:
  - A **coesão** (código claro e escopo delimitado);
  - O **baixo acoplamento do código** (a menor dependência possível entre módulos de um código);

# Definição de Abstração

---

- **Abstração** é a representação do real considerando suas características e comportamentos;
- Na abstração são considerados os aspectos **essenciais** de um contexto **ignorando os detalhes** que não são importantes nele;
- Em um projeto de OO, um sistema é composto por parte menores. Cada nível de abstração deve ignorar os detalhes menores ou maiores para ser resolvido.

# Exemplos de Abstração

---

- Exemplos:
  - Um carro do ponto de vista de um motorista:
    - Características: quilometragem, potência, assentos, modelo...
    - Comportamento: acelerar, frear, limpar o pára-brisa, baixar os vidros, mudar de marcha...
  - Um carro do ponto de vista químico:
    - Características: quantidade de polímeros, metais, vidros, borrachas, papéis, combustível.
    - Comportamento: queima combustível, produz CO, aquece o radiador, gera energia através da bateria

## Evolução das Classes e Objetos:

- Dados primitivos isolados e ligados somente por código.  
Ex: 3 inteiros para representar uma data;
- Estruturas (struct) que agrupavam dados. Ex: Dia, mês e ano fazem mas sentidos juntos;
- Tipos Abstratos de Dados (TAD) combinando armazenamento e procedimentos de manipulação. Ex: pilha, fila, lista, árvore;
- Combinação de armazenamento e comportamento – Origem de OO.

## OO2 - Abstração

# Classes e Objetos

---

- Uma **classe** é um Tipo Abstrato de Dados que define uma **estrutura padrão**, por meio de atributos e métodos, que todos os seus objetos seguem;
- Uma classe é declarada por meio da palavra-chave `class`;
- Um **objeto** é a **instância de uma classe**, ou seja, ele segue toda a estrutura que a classe determina em valores e comportamento;
- Um objeto é **criado** por meio da palavra-chave `new`



Uma analogia para facilitar a compreensão:

- **forma de bolo = classe**
  - Determina uma forma, sem determinar a massa;
- **bolos = objetos**
  - A forma pode ser preenchida por bolos de limão, chocolate, milho ou cenoura. Todos serão bolos e terão a mesma forma.

## OO2 - Abstração

# Classes e Objetos

---

- Definição de uma classe Pessoa em java:

```
1  public class Pessoa{  
2  }
```

- Criação de um Objeto da classe Pessoa:

```
1  public class TestaPessoa{  
2      public static void main (String[] args) {  
3          Pessoa p1 = new Pessoa();  
4          System.out.println(p1);  
5      }  
6  }
```

# Classes e Objetos - Exercícios

---

**Atenção:** Não copie ou cole nenhum exercício. A repetição é intencional para criar fluência na linguagem.

**Observação:** Guarde o diretório sistema, pois incrementaremos as classes com cada conceito de OO aprendido durante esse treinamento.

1. Crie um projeto java chamado sistema.

# Classes e Objetos - Exercícios

2. Dentro do diretório sistema, crie as seguintes classes (todas como públicas):
  - Banco,
  - Agencia,
  - Conta,
  - AplicacaoFinanceira.
3. Dentro da classe AplicacaoFinanceira, crie o método main de Java;

# Classes e Objetos - Exercícios

4. Dentro do método `main`, faça as seguintes operações:
  - Declare e crie um objeto `banco1` do tipo `Banco`,
  - Declare e crie um objeto `agencia1` do tipo `Agencia`,
  - Declare e crie um objeto `conta1` do tipo `Conta`,
5. Compile todas as classes para verificar se existe algum erro.
6. Teste o método `System.out.println(objeto);`