

POO

Msc. Édimo Sousa Silva.

Sistemas de Informação
Faculdade Paraíso (FAP)



Aula 01 - O que é OO

- **Orientação a Objetos**
- **Abstração**
- **Classe**
- **Encapsulamento**
- **Herança**
- **Polimorfismo**
- **Comparação entre POO e Programação Estruturada**

Orientação a objetos

- **Definição:** Um paradigma de programação que representa as entidades como objetos para solucionar problemas.
- **Origem:** na década de 1960, na Noruega. Década de 1970 SmallTalk, década 1980 C++ e na Década de 90 Java.
- **Motivação:** Usar uma lógica similar a do mundo real para resolver problemas computacionais.

Principais características da OO

- **Abstração**
- **Polimorfismo**
- **Herança**
- **Encapsulamento**

Abstração

• **Definição:** Habilidade de identificar os aspectos principais de um contexto para a resolução de um problema, ignorando características menos importantes.

Classe

- Uma abstração de entidades existentes no domínio do sistema.
- Unidade básica de de trabalho em OO.
- Representa um modelo a partir do qual instâncias são criadas.
- É uma coleção de dados (atributos) e operações (métodos) que manipulam tais dados.

Classe - part 2

- Atributos:

- São dados que caracterizam objetos da classe.
- São armazenados em variáveis.
- Constituem o estado do objeto.

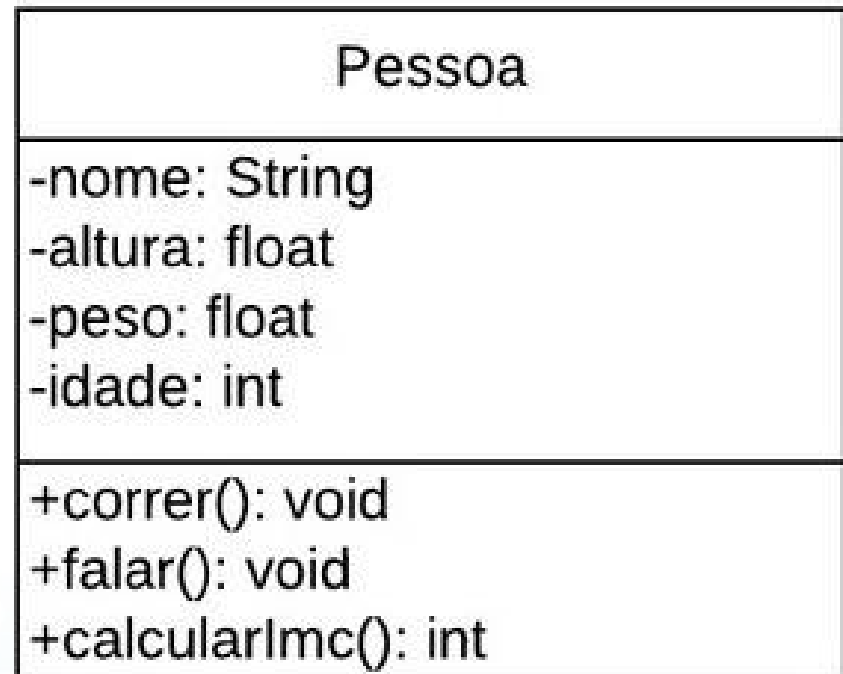
- Operações:

- São os métodos (procedimentos ou funções) que manipulam os dados.

Classe - part 3

```
public class Pessoa {  
    String nome;  
    float altura, peso;  
    int idade;  
  
    void correr () {  
        ...  
    }  
    void falar () {  
        ...  
    }  
    int calcularIMC () {  
        ...  
    }  
}
```

Representação UML



Encapsulamento

- **Definição:** Consiste na separação dos aspectos internos e externos de um objeto.
- É utilizado para impedir acesso direto ao estado de um objeto (atributos).
- Permite ignorar detalhes de implementação (como as coisas funcionam internamente), permitindo ao desenvolvedor um nível mais alto de abstração.
- É um termo formal que descreve a junção de métodos e dados dentro de um objeto.

Herança

- **Definição:** É o mecanismo pelo qual uma classe pode estender outra aproveitando seus comportamentos (métodos) e estados possíveis (atributos);
- Permite que elementos mais específicos incorporem a estrutura e o comportamento de elementos mais genéricos;
- Quando um objeto da classe filho é criado ele herda todas as propriedades da classe pai, além das propriedades definidas na própria classe filho;

Polimorfismo

- **Definição:** Permite que uma mesma operação possa ser definida para diferentes tipos de classes, e cada uma delas a implementa como quiser;
- Um mesmo objeto se manifestar de diferentes formas;

PE vs POO (Reutilização de código)

- **PE:** É possível reutilizar códigos na programação estruturada, porém em muitos casos você será obrigado a utilizar "CTRL C + CTRL V".
- **POO:** é possível elaborar um relacionamento entre diversos componentes, estabelecendo comunicação entre eles e facilitando assim, e muito a reutilização de código, além da facilidade de se herdar atributos e comportamentos de outros objetos.

PE vs POO (Manutenção de código)

- **PE:** A manutenção do código se baseia no que o programador que criou o código fez para deixar de comentários no próprio código ou se escreveu um roteiro sobre o que o programa faz.
- **POO:** se o código seguir os padrões de construção, qualquer programador que conheça os padrões pode facilmente encontrar problemas, utilizar o código escrito ou até mesmo melhorá-lo se assim se fizer necessário.

PE (Vantagens e desvantagens)

- **Vantagens PE:**

- Provê um melhor controle sobre o fluxo de execução do código, quando comparada com a POO.
- É de fácil compreensão, sendo amplamente usada em cursos introdutórios de programação.

- **Desvantagens PE:**

- Ainda se foca em como a tarefa deve ser feita e não em o que deve ser feito.
- Tende a gerar códigos confusos, onde tratamento dos dados são misturados como comportamento do programa.

POO (Vantagens e desvantagens)

- **Vantagens POO:**

- Provê uma melhor organização do código.
- Contribui para o reaproveitamento de código.

- **Desvantagens POO:**

- Pode não possuir o mesmo desempenho de códigos estruturados similares.
- Seus conceitos são de difícil compreensão se comparados aos conceitos da Programação estruturada.

Bibliografia

- P. Boratti. Programação Orientada a Objetos em Java. 2007.
- P. Deitel, H. Deitel. Java Como Programar. 2010.