

Aula – Introdução à Programação Orientada a Objetos

Resultados esperados:

- Conhecer o programa da disciplina e o ambiente de desenvolvimento que será utilizado.
- Entender o que é programação orientada a objetos e comparar com outros paradigmas de programação
- Compreender o que são classes, objetos, atributos e métodos e como criá-los usando uma linguagem de programação. Compreender a forma básica de programar em Java (projetos java, uma classe por arquivo, etc.)

Apresentar a Disciplina, o Plano de Trabalho

Ambiente de Desenvolvimento

- Java
 - Instalar JDK
 - Verificar se foi adicionado no PATH usando javac
- VSCode + Extension Pack for Java

Introdução a Programação Orientada a Objetos

Paradigma Imperativo Procedural

- Instruções são “ordens” descrevendo o que o computador deve fazer.
- Instruções são dadas em sequência. As primeiras instruções são realizadas primeiro pelo computador.
- Programas são organizados em procedimentos que podem ser invocados para alcançar um determinado objetivo.
- O programa é pensado em termos de instruções e procedimentos.

Paradigma Orientado a Objetos

- O programa é pensado em termos de classes de objetos que possuem atributos (características/dados) e métodos (comportamentos/ações/funcionalidades).
- O paradigma OO sugere que devemos armazenar em componentes os dados e as instruções que manipulam esses dados de forma que seja possível reusar esses componentes. Ex.: Sistema de gestão de usuários em sites; Alunos, Professores, Disciplinas, Semestres, Matrículas em uma escola.
- Vantagens de usar POO
 - Reusabilidade: usar como exemplo um sistema de autorização que é comum a quase todos os sites.

- Manutenibilidade: Se existir um bug no sistema de autorização, é fácil identificar. Se o código for bem estruturado, é fácil estender sem estragar o que foi feito antes.
- Reduzir o gap semântico entre o problema do mundo real e a solução computacional
- Grande disponibilidade de materiais sobre engenharia de software usando este paradigma.

Vamos começar com um exemplo:

- Imagine que existem 3 produtos de um supermercado. Cada um destes itens é um objeto. Embora eles sejam objetos diferentes, eles possuem atributos comuns como nome, preço e quantidade em estoque. Existem operações que também são comuns a toda aquela **classe** de objetos, a classe dos produtos. Classes, em orientação a objetos, definem as características compartilhadas por um conjunto de objetos e as operações que estes objetos podem fazer.
- Fazer outro exemplo usando um jogo de luta. O jogo possui personagens. Perguntar aos alunos alguns atributos que os personagens possuem (nome, vida, posição na tela, figuras). Perguntar algumas ações que eles podem fazer. Andar, saltar, realizar ataques físicos e especiais.

Um exemplo inicial: ContaCorrente com atributos numero (String), saldo (double), limite (double) e métodos disponível, deposito e saque.

A minha conta corrente é diferente das contas correntes dos alunos. Cada conta corrente é um objeto. As contas correntes possuem características (número, saldo, limite) e comportamentos comuns (saque, deposito, obter valor disponível). A classe ContaCorrente define os atributos e métodos que todos os objetos daquele tipo possuem.

No exemplo, veremos:

- Como representar a classe usando um diagrama de classes UML.
- Compreender o que é uma classe, o que são atributos e o que são métodos.
- Como criar um projeto Java e como é sua estrutura.
 - É necessário criar um projeto para cada programa
 - Cada classe deve estar em um arquivo diferente.
 - Projetos podem possuir pacotes que são representados por pastas.
- Compreender a classe App que contém o método main e o seu significado.
- Como definir a classe ContaCorrente em seu arquivo. O nome do arquivo e da classe devem ser iguais e começar com letras maiúsculas.
- Como criar objetos (instâncias) do tipo ContaCorrente na classe App. Compreender o operador "new" e o seu significado.
- Como atribuir valores aos atributos usar os métodos.
- Exibir informações na tela usando a função System.out.println().

Exercício: Criar uma classe Círculo com atributos centro_x, centro_y e raio, e métodos area() e perimetro().

Funções matemáticas como pow e sqrt devem ser importadas da biblioteca Math. Para isto, adicionar no início do programa:

```
Import java.lang.Math;
```

E para usar:

```
Math.pow(2, 3); # 8
```