
Conceitos básicos de Orientação a Objetos

Prof. Gabriel M. Alves
IFSP - 2023

Motivação

- Softwares complexos e maiores
- Produtividade e manutenção
- Reutilização de código
- Facilidade de entender, corrigir e alterar
- Na programação estruturada
 - Procedimentos e funções implementados em blocos
 - O programa consiste no acionamento dos procedimentos e funções
- Na programação orientada a objetos
 - Dados, procedimentos e funções são **encapsulados** em uma só entidade, o **objeto**
 - O programa consiste na comunicação entre os objetos (troca de mensagens)

Classe e Objeto

- Objeto é uma entidade que materializa a nossa abstração do elemento no domínio do problema.
- O objeto guarda informações sobre o elemento abstraído.
- O objeto é uma maneira de diminuir o gap semântico, pois é uma entidade mais próxima da entidade no domínio do problema.
- Um objeto possui propriedades, também chamadas de características ou atributos, que são campos destinados ao armazenamento de informações.
- Um objeto possui comportamentos (métodos) que refletem as ações que ele pode realizar como, por exemplo, interagir com outros objetos.

Classe e Objeto

- Um objeto possui estado, comportamento e identidade.
- O estado indica os valores que os atributos assumem em um determinado momento.
- O comportamento pode alterar o estado do objeto e o relacionamento com outros objetos.
- A identidade significa que cada objeto é único.
- Exemplos de objetos:
 - Carro
 - Cachorro
 - Televisão, etc

Classe e Objeto

- Uma classe descreve um conjunto de objetos que possuem características e comportamentos semelhantes.
- O objeto é uma entidade concreta enquanto a classe é apenas uma abstração.
- Definir uma classe significa definir um Tipo Abstrato de Dados (TAD), enquanto declarar objetos significa criar variáveis do TAD definido.
- Criar um objeto a partir de uma classe é chamado de instanciar um objeto.
- É possível determinar o tipo de acesso aos membros de uma classe (atributos e métodos)

Tipos de acesso

- **Público:** o atributo ou método pode ser acessado por todas as demais entidades do programa.
- **Protegido:** o atributo ou método somente pode ser acessado por classes da mesma hierarquia.
- **Privado:** o atributo ou método pode ser acessado somente por métodos da própria classe.
- **Default:** o atributo ou método pode ser acessado por entidades do mesmo pacote (escopo de pacote).

Pilares da Programação Orientada a Objetos

- Abstração
- Encapsulamento
- Herança
- Polimorfismo
- Mensagens

Pilares da Programação Orientada a Objetos

- **Abstração:** consiste em identificar e selecionar detalhes relevantes do domínio do problema e representá-los em uma linguagem de programação.
- **Encapsulamento:** reunir dados e procedimentos correlacionados em uma mesma entidade.
 - O intuito é que o sistema não dependa da implementação interna mas da interface, ou seja, manter as informações ocultas o máximo possível.
 - Para outras entidades o importante é saber como se comunicar com o objeto.
- **Herança:** estabelece hierarquia entre classes de um sistema de modo que a classe mais especializada herda os membros de uma classe mais geral a fim de promover o reuso de software.

Pilares da Programação Orientada a Objetos

- **Polimorfismo:** capacidade de um método comportar-se diferente nas classes especializadas em uma herança de classes.
 - Uma subclasse pode sobrescrever o comportamento da superclasse.
 - Métodos com a mesma assinatura, mas com serviços diferentes
- **Mensagem:** é a transmissão por um objeto emissor para um objeto receptor para que este execute uma operação desejada.

Pacotes e bibliotecas

- **Pacote:** conjunto de classes relacionadas entre si, também recebe o nome de encapsulamento de nomeação.
- **Biblioteca:** conjunto de pacotes
 - Permite o reuso de código
 - API (Application Programming Interface)
 - Interessante verificar se já não há uma biblioteca que resolva o problema em questão, pois a confiabilidade é maior e pode reduzir o tempo de programação.