

Introdução à Orientação a Objetos

Roland Teodorowitsch

Programação Orientada a Objetos - ECo - Curso de Engenharia de Computação - PUCRS

1 de setembro de 2021

Conceitos

Paradigma Programação

- É uma forma de se classificar as linguagens de programação a partir de suas funcionalidades
- É um padrão conceitual que orienta soluções de projeto e implementação
- Fornece e determina a visão que o programador possui sobre a estruturação e execução de seus programas
- Paradigmas explicam como os elementos que compõem um programa são organizados e como interagem entre si
- Exemplos: procedural, funcional, orientado a objetos
- Uma linguagem pode suportar mais de um paradigma
- Um programa pode NÃO aproveitar as funcionalidades de um paradigma

Programação Orientada a Objetos

- É um modelo de análise, projeto e programação de *software* baseado na composição e interação entre unidades chamadas de **objetos**
- É um estilo de programação que se baseia na modelagem de **objetos** do mundo real

Objetos

- São entidades que pode ser facilmente reconhecidas
- São abstrações de objetos do mundo real
- Um objeto é uma estrutura composta de:
 - **atributos**: características ou dados
 - **comportamentos**: operações, interface ou métodos
- Exemplos:
 - caneta
 - características: cor da tinta, quantidade de tinta, etc.
 - comportamento: escrever, recarregar, etc.
 - lâmpada
 - características: ligada (sim/não), potência, voltagem, etc.
 - comportamento: ligar, desligar, queimar, etc.

Programa Orientado a Objetos

- É estruturado como um conjunto de objetos que interagem entre si
- Cada objeto tem um papel a cumprir
- Cada objeto oferece um serviço ou realiza uma ação que é usada por outros membros do conjunto
- Exemplo: um computador e seus diversos componentes (teclado, mouse, vídeo, UCP, etc.)

Abordagem OO

- Princípios da abordagem OO
 - **Abstração**: representação de um objeto do mundo real, “abstraindo-se” os detalhes desnecessários, de forma que o objeto possa ser utilizado sem se preocupar com como ele foi implementado
 - **Encapsulamento**: detalhes da implementação ficam escondidos e a manipulação dos dados acontece através de uma interface pública
 - **Modularidade**: vários componentes que interagem
- Abstração, Encapsulamento, Herança e Polimorfismo são considerados os 4 pilares da POO

Encapsulamento

- É um dos conceitos básicos da OO
- A ideia é que cada objeto seja uma “caixa preta”:
 - Não é necessário saber os detalhes de seu funcionamento interno
 - Basta saber como utilizá-lo
- Encapsular é esconder como as coisas funcionam (*data hiding*) por trás de uma interface externa (pública)
- Exemplo: termostado
 - Aciona um dispositivo a partir de determinada temperatura
 - Como ele é implementado internamente?
 - Como pode ser usado?

Encapsulamento

- Um objeto é formado por:
 - Dados (Atributos):
 - São **privado**
 - Acesso aos dados deve ser feito através da chamada de um método
 - Operações (Métodos):
 - São **públicos**
 - Interface pública declara as operações permitidas

Benefícios do Encapsulamento

- A implementação interna de um objeto pode mudar e o resto do sistema não é afetado (desde que a interface de acesso não mude)
- Maior segurança ao proteger os atributos de um objeto de alterações indevidas por outros objetos
- Maior independência entre os objetos, pois eles só precisam conhecer a interface externa definida

Abstração

- Quando modelando um objeto, identificamos **somente** as operações e atributos que são essenciais
- Exemplo: caneta
 - Atributos não modelados: fabricante, etc.
 - Operações não modeladas: apontar, etc.

Classes

- Uma classe é uma “forma” para produzir objetos
- Determina um conjunto de objetos com:
 - Propriedades semelhantes
 - Comportamentos semelhantes
 - Relacionamentos comuns com outros objetos

Instâncias

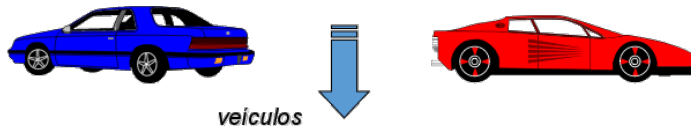
- **Instanciar** objetos significa gerar novos exemplares a partir de uma descrição abstrata de um objeto genérico (ou seja, de uma classe)
 - Objetos são instâncias de uma classe



Classes x Instâncias x Objetos

- Objetos são gerados a partir de classes
- Uma classe define as propriedades e o comportamento dos objetos gerados por ela
- Todo objeto é uma instância de uma classe

Classes x Instâncias x Objetos



<i>veículo</i>
<i>Marca</i> <i>Placa</i>
<i>ler_placa</i>

classe

Ferrari
IDN8810

objeto1

Fiat Uno
IAS3333

objeto2

instâncias

Exemplo em C++

```
#include <iostream>
#include <string>

using namespace std;
```

```
class Veiculo {
```

```
private:
```

```
    string marca;
    string placa;
```

➡ Atributos

```
public:
```

```
    Veiculo(string m, string p);
    string ler_placa();
```

➡ Métodos

```
};
```


Objetivo da POO

- Gerar programas que sejam:
 - Legíveis e de fácil compreensão
 - Reutilizáveis
 - De fácil manutenção
 - Obtidos de forma produtiva

Resumo: Conceitos de OO

- Objeto
 - Unidade básica de orientação a objetos
 - É uma entidade que tem atributos, comportamento e identidade
 - É um membro de uma classe e os atributos e o comportamento (métodos) de um objeto são definidos pela classe
- Classe
 - É uma descrição de um conjunto de objetos
 - Este conjunto de objetos compartilha atributos e comportamento em comum
 - Uma definição de classe descreve todos os atributos dos objetos membros da classe, bem como os métodos que implementam o comportamento destes membros

Resumo: Conceitos de OO

- Orientação a objetos
 - Um método de desenvolvimento de *software* que usa abstração com objetos, classes encapsuladas e comunicação por mensagens, hierarquia de classes e polimorfismo
- Abstração
 - Um modelo de um conceito ou objeto do mundo real
- Encapsulamento
 - Processo de esconder os detalhes internos de um objeto do mundo externo
- Atributo
 - Usado para armazenar o estado de um objeto
 - Pode ser simples como uma variável escalar (`int`, `char`, `double` ou `bool`) ou pode ser uma estrutura complexa tal como outro objeto

Resumo: Conceitos de OO

- Comportamento
 - Atividade de um objeto que é vista do ponto de vista do mundo externo
 - Inclui como um objeto responde a mensagens alterando seu estado interno ou retornando informação sobre seu estado interno
- Método
 - Uma operação ou serviço executado sobre o objeto, declarado como parte da estrutura da classe
 - Métodos são usados para implementar o comportamento do objeto
- Estado
 - Reflete os valores correntes de todos os atributos de um objeto e são o resultado do comportamento do objeto ao longo do tempo

Exemplo

Projetando Objetos

De uma forma simples, o **projeto orientado a objetos** de um sistema pode ser dividido em três etapas:

- 1 Identificar as abstrações/entidades envolvidas no problema
- 2 Identificar os serviços que cada uma destas entidades deve ser capaz de fornecer
- 3 Identificar os relacionamentos entre essas entidades

Exemplo

- Deseja-se criar um sistema de cadastro de produtos que são vendidos em um supermercado. Cada produto possui uma descrição e um valor de venda. O sistema permite a emissão de relatórios dos produtos disponíveis. Também, é permitido ao gerente aplicar reajustes de preços sobre o produto que desejar.

Exemplo

Interface do Produto:

- ✓ Visão externa da abstração Produto

Produto
+ inicializa (descrição:String, preço:double) + reajuste (percentagem:double) + informaDados():String

Classe do Produto:

Produto	nome da classe
- descrição:String - preço:double	} atributos
+ inicializa (descrição:String, preço:double) + reajuste (percentagem:double) + informaDados():String	} métodos

Instâncias do Produto:

prod1
- "banana" - 1.29
+ inicializa (descrição:String, preço:double) + reajuste (percentagem:double) + informaDados():String

prod2
- "café" - 5.40
+ inicializa (descrição:String, preço:double) + reajuste (percentagem:double) + informaDados():String

Créditos

Créditos

- Estas lâminas contêm trechos de materiais disponibilizados pelos professores Rafael Garibotti e Edson Moreno.