

Análise e Projeto Orientado a Objetos

Classes e Objetos

2021/01

Sumário

Classes e Objetos

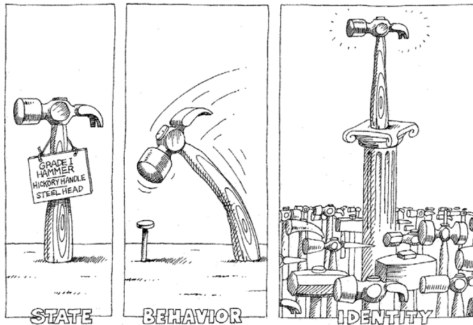
Objetos e Relacionamentos

Exemplo

Referências

Classes e Objetos

- Um objeto tem estado, comportamento e identidade. A estrutura e o comportamento de objetos similares são definidos em sua classe.



Estado

- ▶ O estado de um objeto é determinado pelo conjunto de suas propriedades (estático) e pelos respectivos valores de cada propriedade (dinâmicos).

```
public class Funcionario {  
    private String nome;  
    private Integer cpf;  
    private String departamento;  
    private Double salario;  
}
```

Comportamento

- ▶ O comportamento de um objeto define como o mesmo age ou reage, em termos de mudanças de estado e de envio/recebimento de mensagens.

```
List<Funcionario> funcionarios = new ArrayList<>();  
System.out.println(funcionarios.length());  
  
funcionarios.add(joao);  
funcionarios.add(maria);  
System.out.println(funcionarios.length());  
  
funcionarios.remove(0);  
System.out.println(funcionarios.length());
```

Operações

- ▶ Modificadores: alteram o estado do objeto.
- ▶ Seletores: acessam, sem alteração, o estado do objeto.
- ▶ Construtores: permitem a criação de objetos e inicialização de estado.

Identidade

- Identidade é a propriedade que distingue um objeto dos demais.

```
public static void movimenta(Ponto ponto) {  
    ponto.setX(ponto.getX() + 1);  
    ponto.setY(ponto.getY() + 1);  
  
    ponto = new Ponto(10, 20);  
}  
  
public static void main(String[] args) {  
    Ponto p1 = new Ponto(2, 3);  
    movimenta(p1);  
    p1.imprime(); // x = 3, y = 4  
}
```

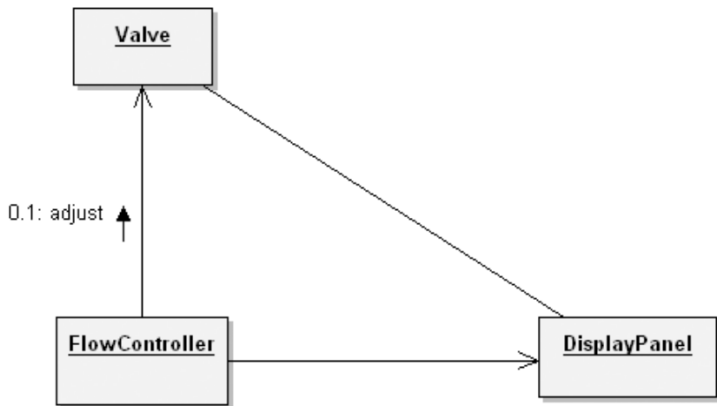
Objetos e Relacionamentos

- ▶ Objetos cooperam para a execução de tarefas complexas, cada um exercendo um papel específico.
- ▶ Relacionamentos: links (cliente/servidor) e agregações (todo/parte).

Links

- ▶ Conexões físicas/conceituais entre objetos.
- ▶ *Controllers*: objetos que invocam operações de outros objetos.
- ▶ *Servers*: objetos cujas operações são invocadas por outros objetos.
- ▶ *Proxies*: objetos que exercem os papéis de atores e servidores.

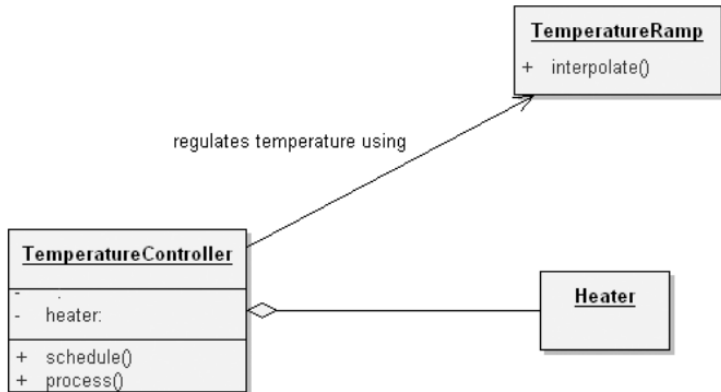
Links



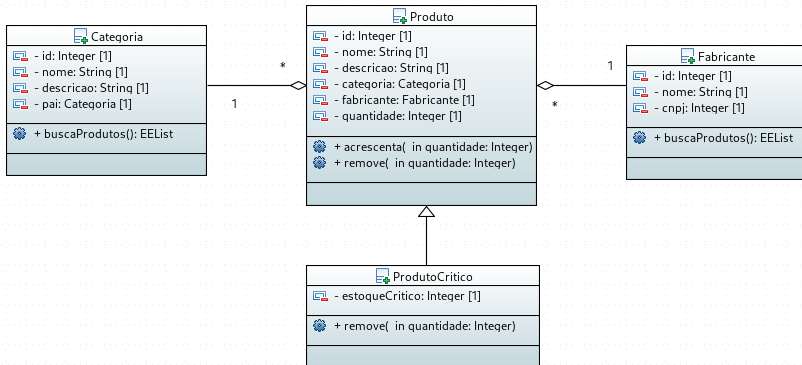
Agregação

- ▶ Implementa a hierarquia *todo/parte*.
- ▶ Navegação implícita do *todo* para a *parte*.
- ▶ O objeto *parte* compõe o estado do objeto *todo*.

Agregação



Exemplo



Referências

- ▶ Booch, G. *et al.*. *Object-oriented analysis and design with applications*, 3ª edição. Boston, MA, USA, Pearson Education.
- ▶ Page-Jones, M. *Fundamentos do desenho orientado a objeto com UML*. São Paulo, Makron Books, 2001.
- ▶ Sommerville, I. *Engenharia de Software*, 9ª edição. São Paulo, Pearson Prentice Hall, 2011.