Orientação a objetos

Prof. Nelson Bellincanta Filho

Orientação a objetos

- Técnica de programação baseada na construção e utilização de objetos;
- Um objeto combina dados e operações específicas, o que define um conjunto particular de responsabilidades;
- Um sistema OO é um conjunto de objetos que se relacionam para produzir os resultados desejados;

Orientação a objetos

- A construção de sistemas OO é muito eficiente, pois a delimitação das responsabilidades dos objetos permite que suas classes sejam utilizadas em diferentes situações (reusabilidade);

- Além disso, a manutenção do código é simplificada, pois, dependendo da qualidade do projeto, podem ser feitas modificações em elementos específicos sem que isso implique alterações nas demais partes do sistema.

Classes e objetos

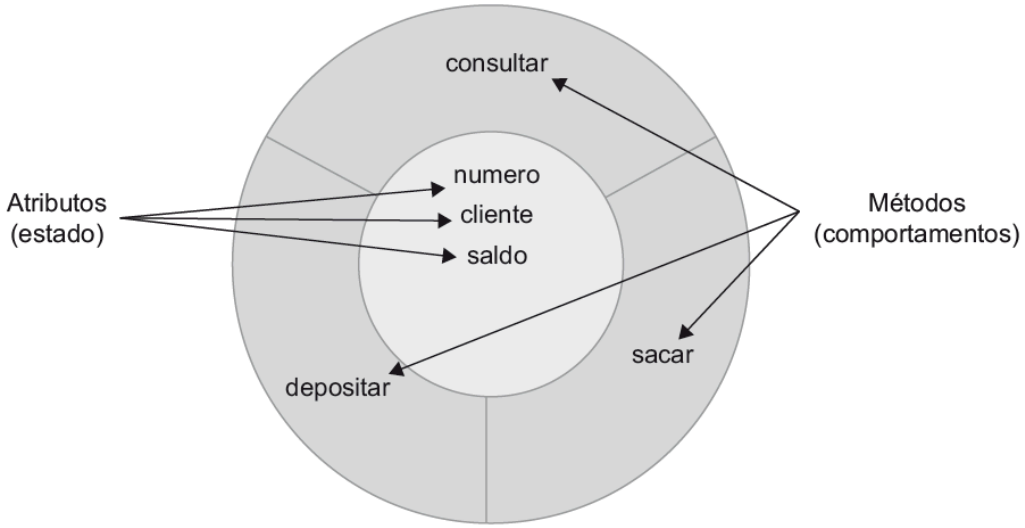
Classes e objetos

- Todo objeto é de um tipo específico, ou seja, pertence a uma classe (class);
- Na programação OO, a criação e utilização de objetos requer modelos para descrevê-los;
- Uma classe é um modelo definido pelo programador para um novo tipo de objeto, que relaciona seus atributos (características e estados) e seus comportamentos (funcionalidades), podendo representar uma entidade real ou abstrata.

Classes e objetos

- As classes são utilizadas para definir novos tipos de objetos;
- Uma conta-corrente, um automóvel, uma camisa ou um cachorro são exemplos de classes, pois cada um desses tipos de objetos encerra um propósito, um conjunto de características e um funcionamento ou comportamento específicos;
- Em um banco existem muitas contas-corrente diferentes, de propriedade de indivíduos distintos, com saldos e históricos de transações próprios, ao mesmo tempo que todas as contas-corrente têm um mesmo funcionamento e características comuns.

Classes e objetos



Objeto contaBancaria

Declaração de classes e variáveis de instância

Declaração de classes e variáveis de instância

- A API Java é constituída de um grande número de classes, cada uma com características e funcionalidades próprias, que podem ser utilizadas na construção de sistemas OO;

- A implementação de classes em Java é simples e emprega a palavra reservada class:

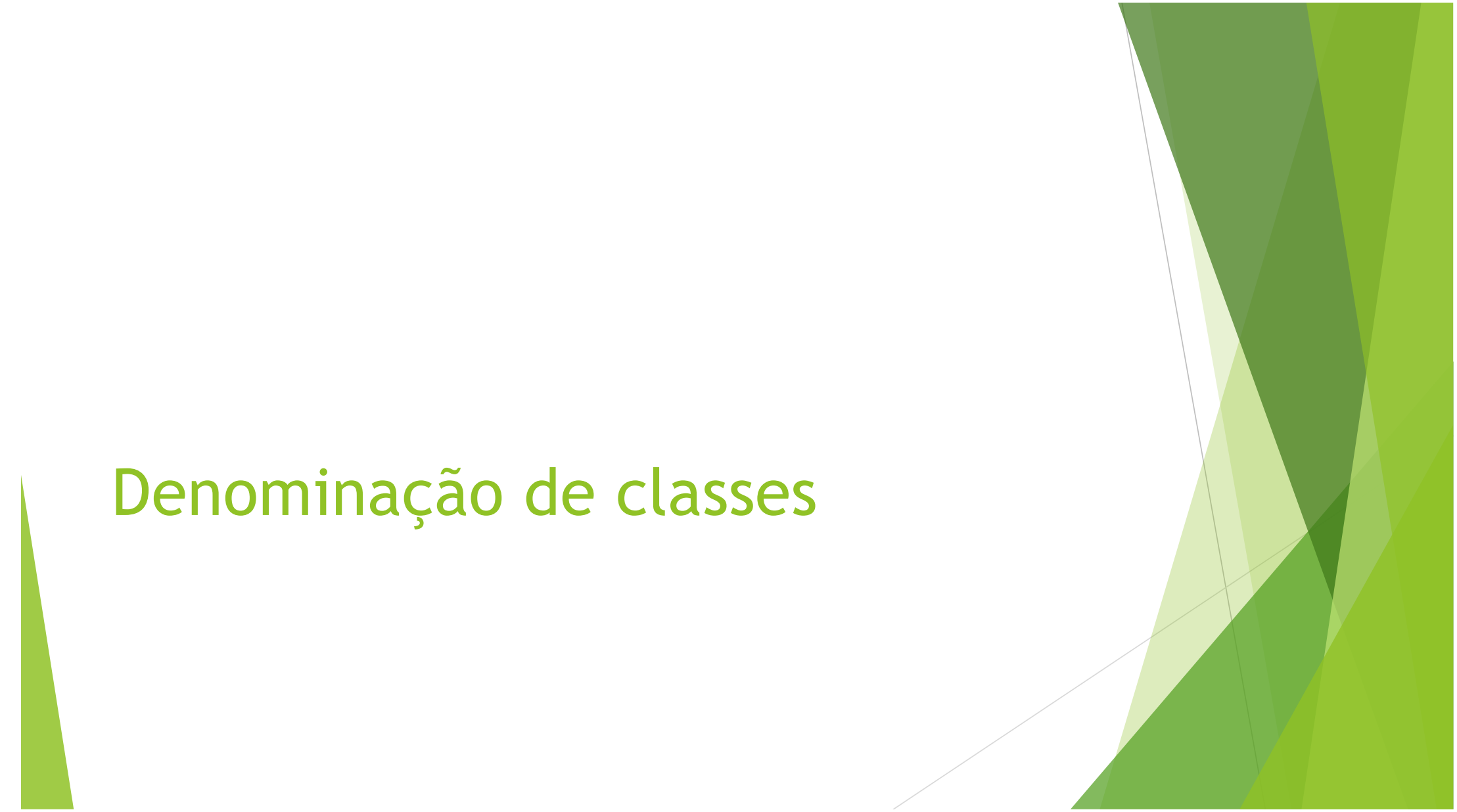
```
qualificador class NomeDaClasse { // declaração de uma classe
    // corpo da classe
}
```

Declaração de classes e variáveis de instância

- A partir de uma classe concreta é possível fabricar ou criar objetos desse tipo com uma operação chamada de instanciação, a qual envolve o operador new e uma chamada especial utilizando o nome da classe (construtor):

```
// declaração de variável de instância
<Classe> variável;
// instanciação de objeto
variável = new <Classe>([parâmetros]);
```

Denominação de classes



Denominação de classes

- Nomes de classes devem ser iniciados com letras maiúsculas, sendo as demais letras em minúsculas;
- No caso de nomes compostos de várias palavras, cada inicial deve ser maiúscula para melhorar a legibilidade.
- Embora permitido, não se recomenda o uso de dígitos, cifrão $ ou sublinhado _.

Visibilidade

Visibilidade

- A visibilidade ou acessibilidade é um aspecto de extrema importância na programação OO, pois permite que o programador restrinja o uso de certos elementos da classe;
- É com a restrição de acesso que se implementa, em grande parte, o encapsulamento de informações e das implementações dentro das classes;
- O Java possui três especificadores de acesso explícitos (public, protected e private) e um especificador implícito.

Visibilidade

Especificador	Nível	Indica que o campo ou método:
public	público	Pode ser usado livremente pelas instâncias da classe.
	pacote	Só pode ser usado por classes e instâncias dentro do mesmo pacote.
protected	protegido	Só pode ser usado na implementação de subclasses e instâncias dentro do mesmo pacote.
private	privado	Não pode ser usado fora da implementação da própria classe.

Especificadores de acesso.

Visibilidade

- O uso dos especificadores possibilita que o programador decida a visibilidade dos elementos da classe, ou seja, se campos ou métodos podem ser utilizados livremente (public);
- Se devem ficar ocultos (private), evitando seu uso;
- Ou ainda se poderão ser empregados na construção de novas subclasses (protected) por meio do mecanismo da herança.

Campos(fields)

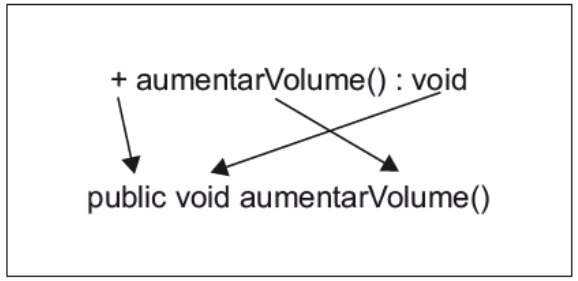
Campos(fields)

- Os campos (fields), atributos ou as variáveis-membro de uma classe são variáveis destinadas a armazenar dados que caracterizam o objeto, seu estado ou suas partes.

- Sua declaração, dentro do corpo da classe, usa a sintaxe:

<especificadorDeAcesso> <tipo> nomeCampo [= expressãoInicialização];

Tv
+ volume : int + canal : int
+ aumentarVolume() : void + reduzirVolume(): void +trocarCanal(int canal) : void +mostrar(): String



Correspondência UML e Java

Referências

- DEITEL, Harvey M.; DEITEL, Paul J. **Java: como programar.** 10ª ed. Editora Pearson Education do Brasil, 2017.
- FURGERI, Sérgio. **Java 8: Ensino Didático**. 1ª ed., São Paulo: Érica, 2015.
- JUNIOR, Peter Jandl. **Java Guia do Programador.** 4ª Edição: Atualizado para Java 16. Novatec Editora, 2021.