PROGRAMAÇÃO ORIENTADA A OBJETOS [POO]

Material 01 – POO_01

Prof. Mestre Marcos Roberto de Moraes [Maromo]

Orientação a Objetos
Alguns conceitos importantes

- Programação Estruturada X Programação OO
- Orientação a Objetos
- Classes e Objetos
- Atributos e Métodos
- Visibilidade
- Herança / Herança Múltipla
- Polimorfismo
- Abstração / Classes Abstratas
- Exercícios Propostos

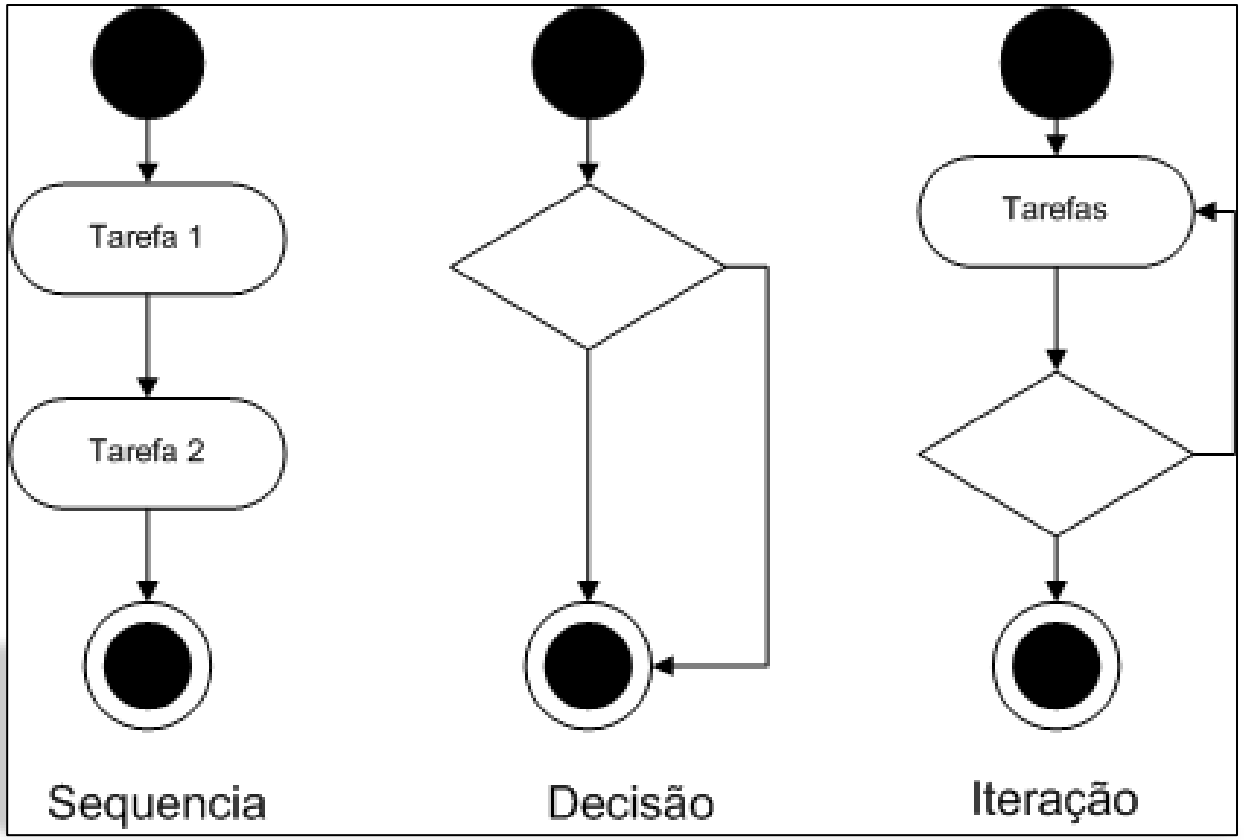
Nota sobre a aula

- Apresenta-se de forma resumida os principais conceitos de Orientação a Objetos que será alvo de estudo ao longo do curso.
- Mostra-se um comparativo entre os Paradigmas de Programação Estruturada versus Orientado a Objetos

Programação Estruturada

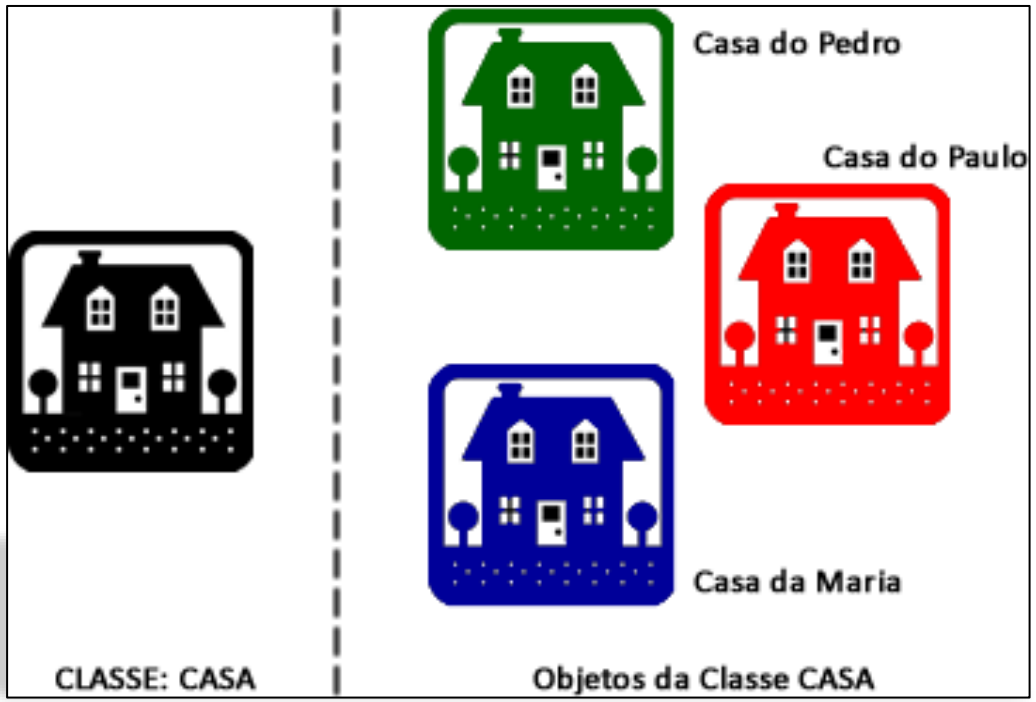
- Fortemente baseada em modularização (dividir para conquistar):
  - Unidades menores são construídas para desempenhar uma tarefa específica.
  - Pode ser executada várias vezes.
  - Funções podem receber parâmetros e retornar valor.

Programação Estruturada: Mecanismos de Interligação



Orientação a Objetos

- Representação por meio de abstrações.
  - Ex: As crianças aprendem a reconhecer coisas simples como pessoa, carro e casa. Cada objeto é um exemplo de um determinado grupo.



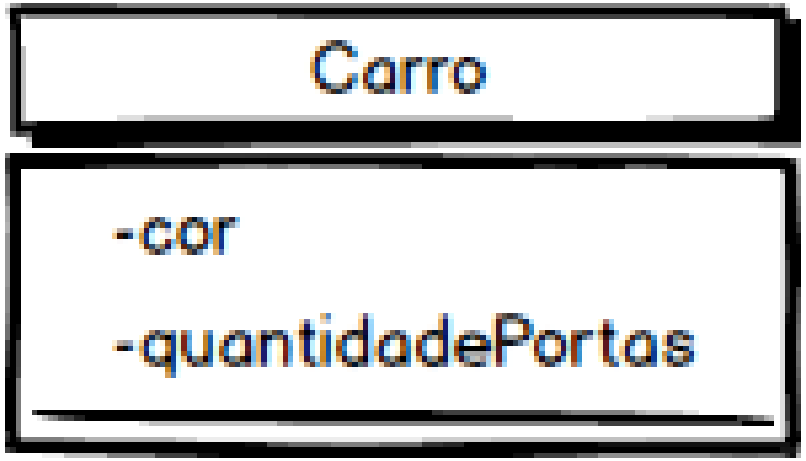
Classes

- Classe é uma estrutura que define um tipo de dados, podendo conter:
  - Atributos ou propriedades e;
  - Métodos ou comportamentos.

Nota: na Linguagem UML (Linguagem de Modelagem Unificada) uma Classe é identificada por um retângulo. A figura acima apresenta uma divisão. Entretanto, normalmente é representada com três divisões.
Por padrão na linguagem JAVA a classe é definida com a primeira letra maiúscula.

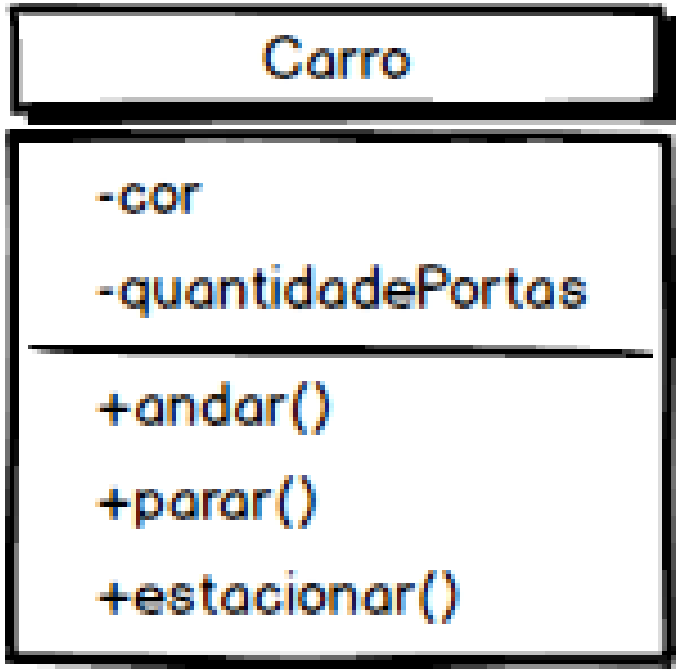
Atributos

- Representam características de uma classe, ou seja, as pecularidades que variam de objeto para objeto.
  - Ex: Classe Carros. Atributos: cor, quantidade de portas, etc...



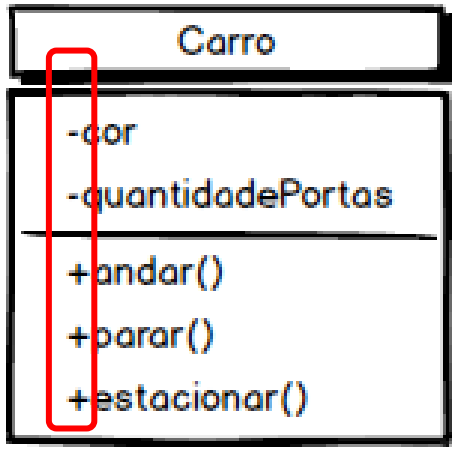
Métodos
ou comportamentos

- Representam uma atividade que um objeto de uma classe pode realizar.
  - Ex: um objeto da classe **Carro**: anda, para, estaciona, etc.



Visibilidade

- Indica o nível de acessibilidade de um determinado atributo ou método.
  - Pública [+]
    - Atributo ou método pode ser usado por qualquer classe;
  - Protegida [#]
    - O modificador protected torna o membro acessível às classes do mesmo pacote ou através de herança;
  - Privado [-]
    - Somente a classe que possui o atributo ou método poderá utilizá-lo.

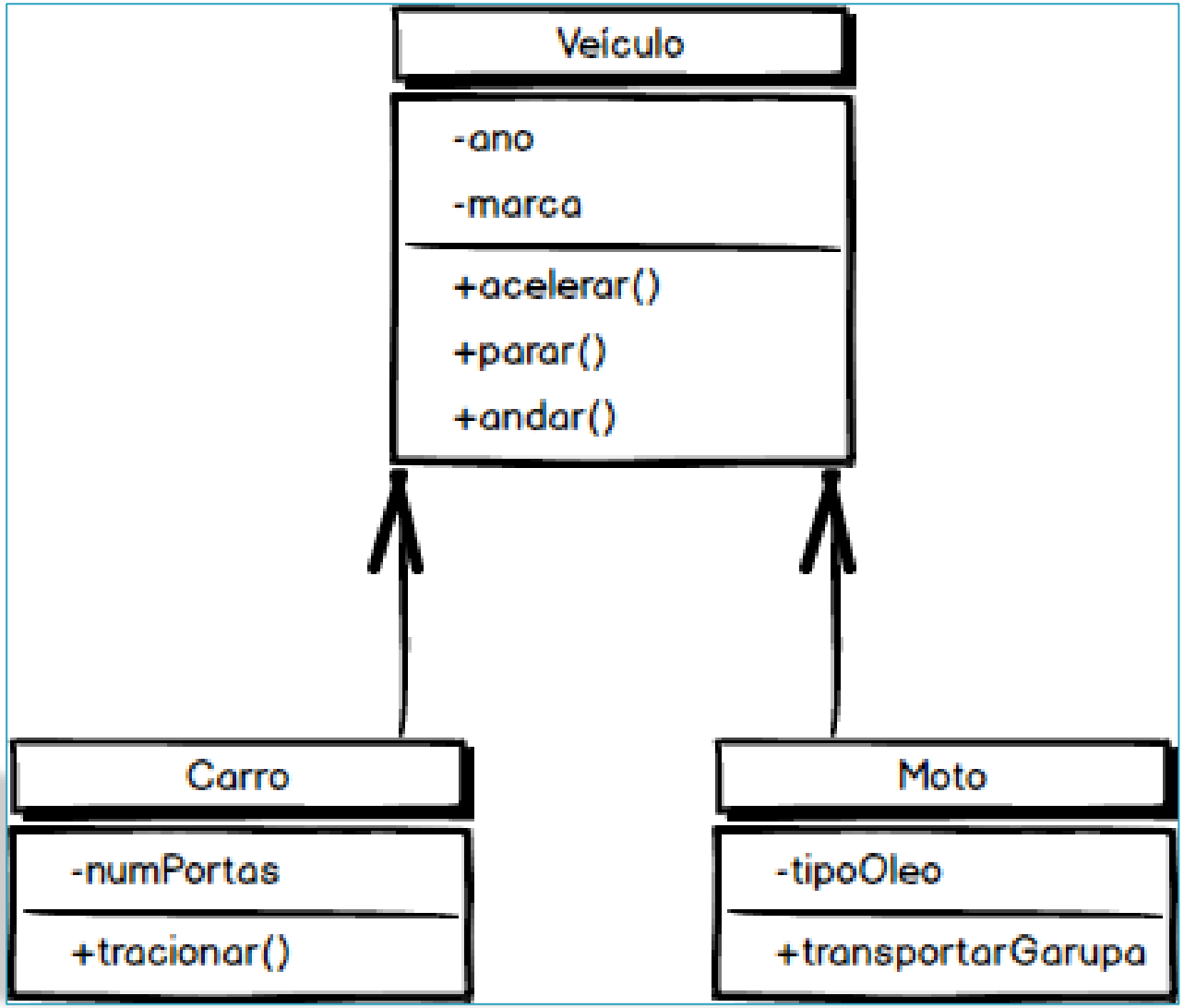


Herança

- Para entender o conceito de herança, primeiro precisa-se saber o que são: superclasses e subclasses.
  - **Superclasse:** ou classe-mãe, é uma classe que possui classes derivadas a partir dela.
  - **Subclasse:** ou classe-filha, é uma classe derivada de uma superclasse.
- As classes-filhas (subclasses) herdam características de uma classe-mãe (atributos e métodos).

Herança

Atente às explicações do professor

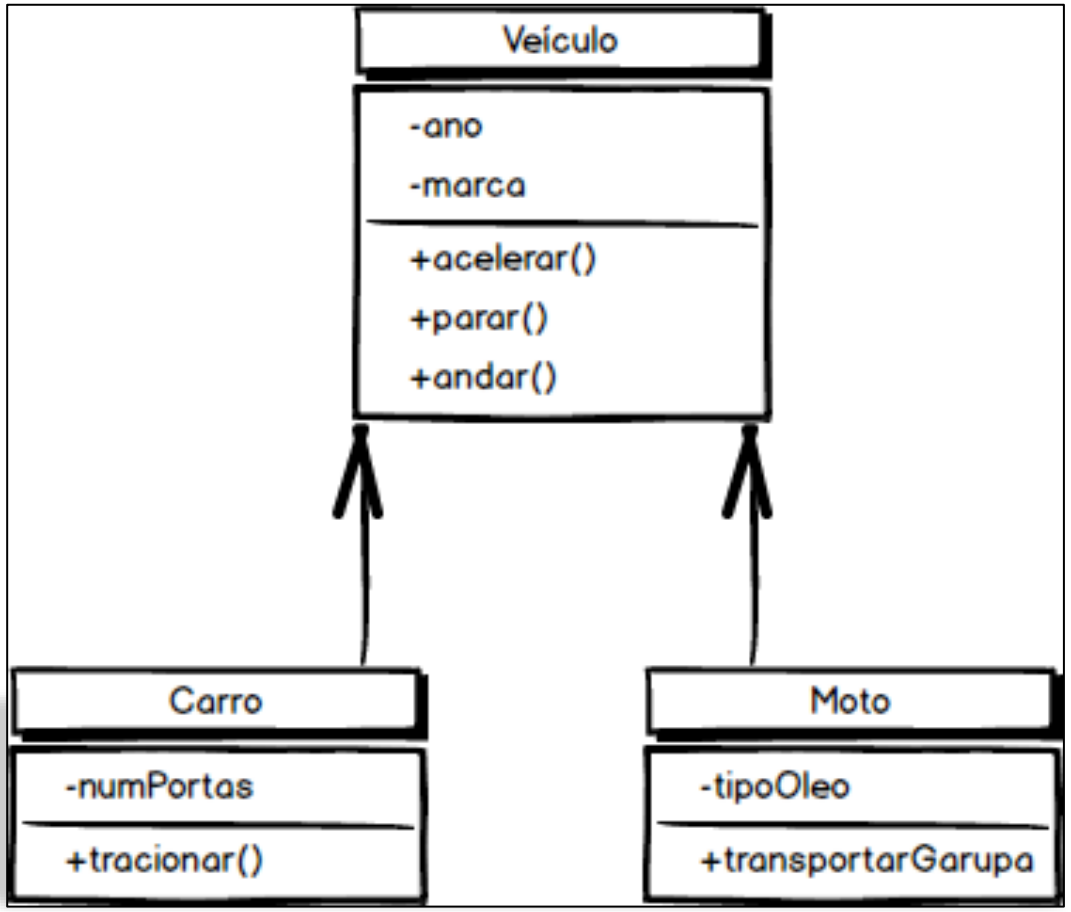


Herança - Vantagem

- A principal vantagem é o compartilhamento de atributos e comportamentos entre classes de uma mesma hierarquia.
- Podemos reaproveitar uma estrutura já existente que nos forneça uma base abstrata para o desenvolvimento de software.

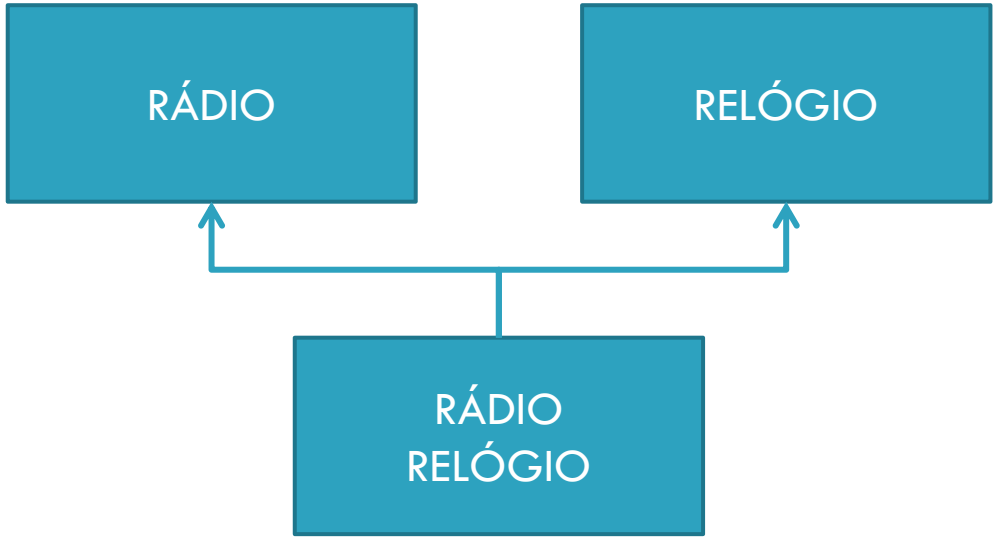
Herança: Generalização e Especialização

- Podemos criar classes **gerais**, com características compartilhadas por muitas classes (Veiculo), mas que possuem diferenças pequenas entre si (Carros possuem portas, motos não) – consideramos carro e moto como especializações de veículos.



Herança Múltipla

- Ocorre quando uma subclasse herda características de duas ou mais superclasses.

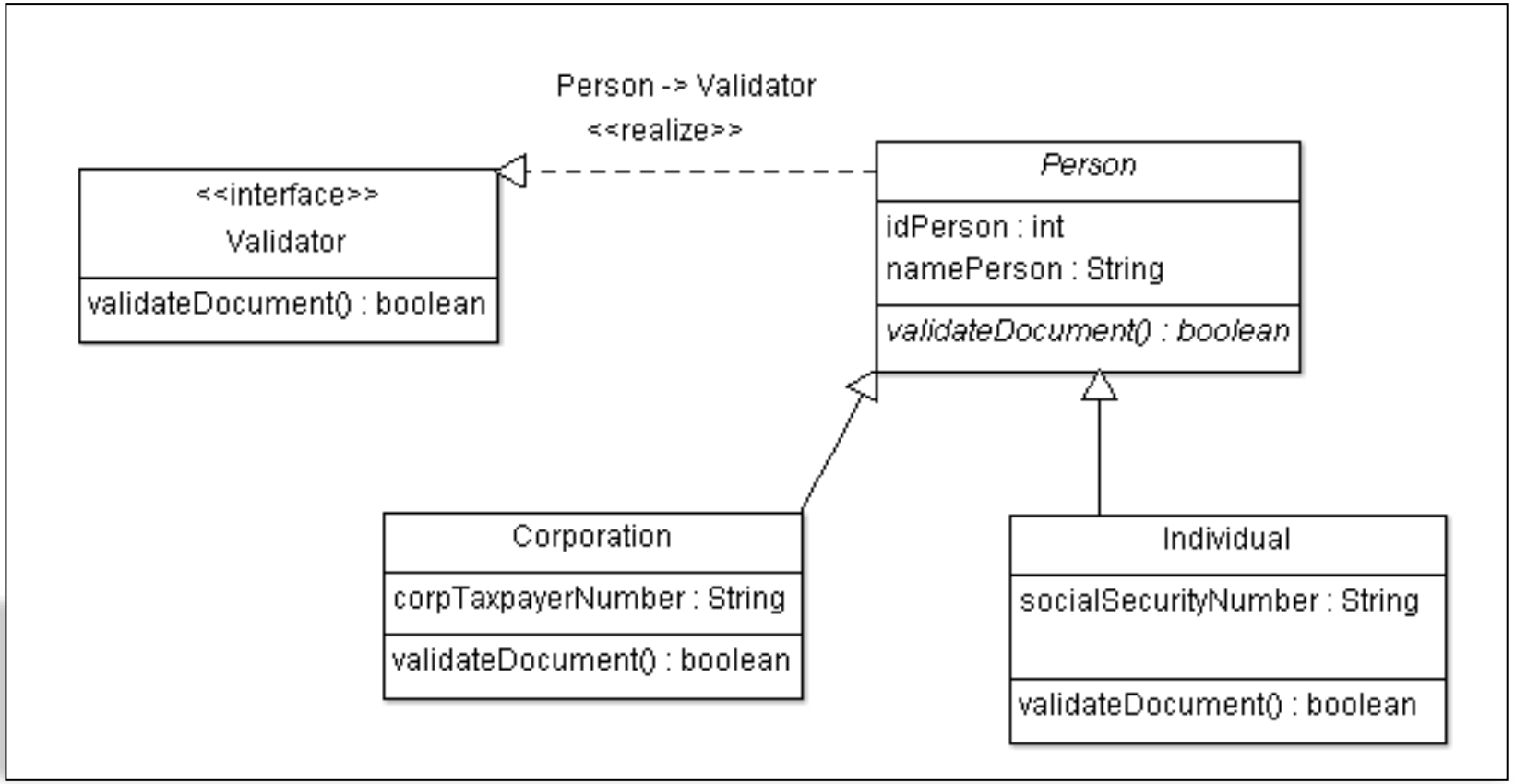


Java abandona a ideia de herança múltipla.
Em seu lugar usa-se interfaces.

Polimorfismo

- O termo polimorfismo é originário do grego e significa "muitas formas" (*poli* = muitas, *morphos* = formas).
- Numa linguagem de programação, isso significa que pode haver várias formas de fazer uma "certa coisa".
- Dois formas: polimorfismo com hierarquia de classes e sem hierarquia.

Polimorfismo



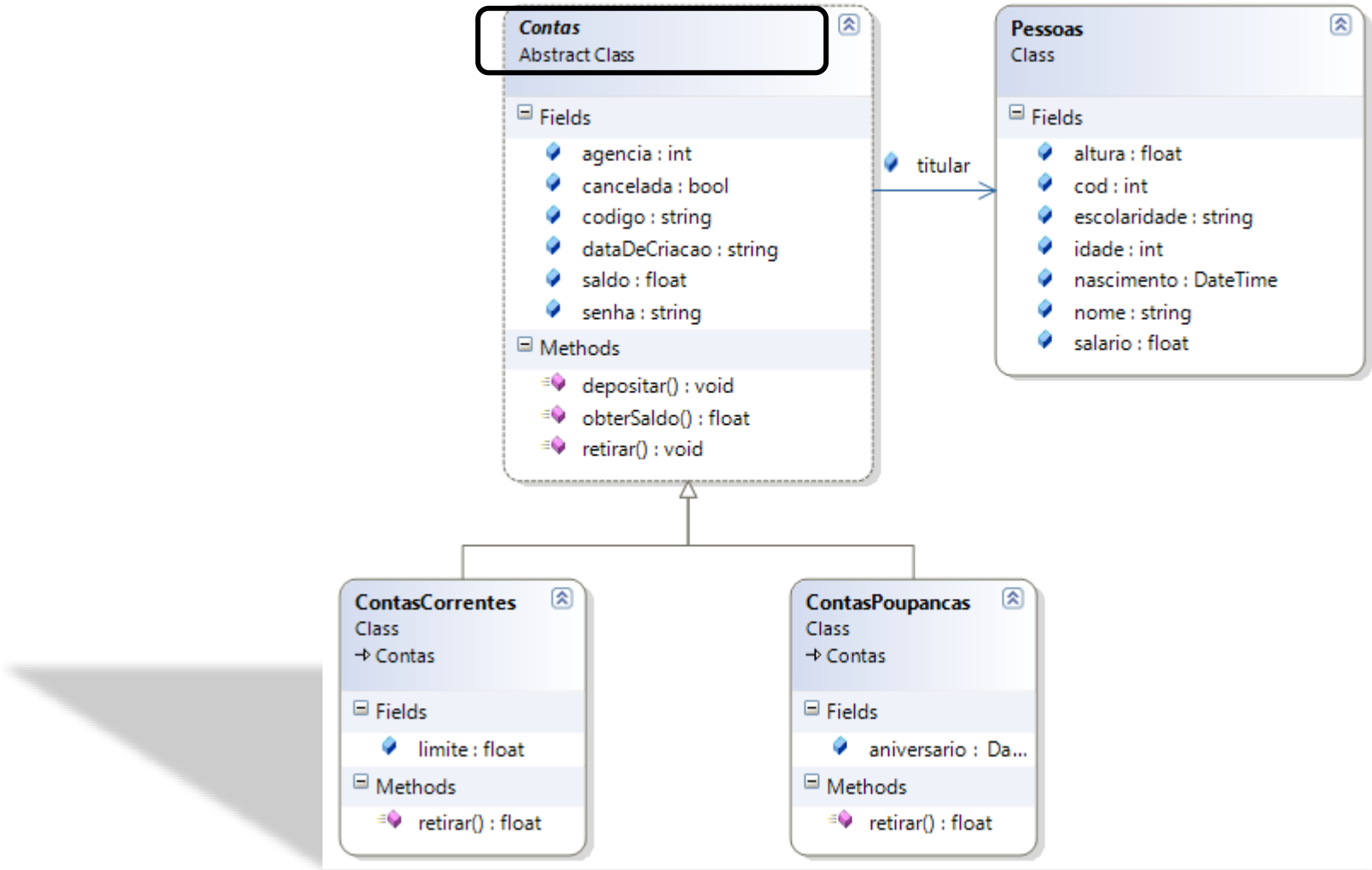
Abstração

- Capacidade de considerar de forma isolada, simplificar, determinar o problema genericamente, dando importância aos aspectos mais relevantes.
- Na prática, construir “peças” (classes) bem definidas que possam ser reaproveitas, formando uma estrutura hierárquica.

Classes Abstratas

- Classes estruturais, que servem de base para outras. São classes que nunca serão instanciadas na forma de objetos; somente suas derivadas serão.
  - Ex: Num domínio de uma agência bancária uma determinada pessoa pode ter uma Conta Corrente ou uma Conta Poupança, mas jamais poderá ter uma Conta (Genérica).
  - Neste exemplo a Classe Conta tem todo sentido ser considerada como uma classe Abstrata.

Exemplo: Classe Abstrata - Conta



Exercícios

- A) Pesquise e comente sobre a importância do desenvolvimento de software usando o paradigma de programação orientado a objetos. Procure por outros paradigmas e trace uma comparação entre eles.

- B) Procure saber sobre: “o que são *design patterns*”. Tente descobrir a razão pela qual as empresas estão utilizando cada vez mais de padrões de projeto.

Referências Bibliográficas

- Mendes; **Java com Ênfase em Orientação a Objetos**, Novatec.
- Deitel; **Java, como programar** – 10° edição. Java SE 7 e 8
- Arnold, Gosling, Holmes; **A linguagem de programação Java** – 4° edição.
- Apostilas da Caelum.

FIM

Maromo