



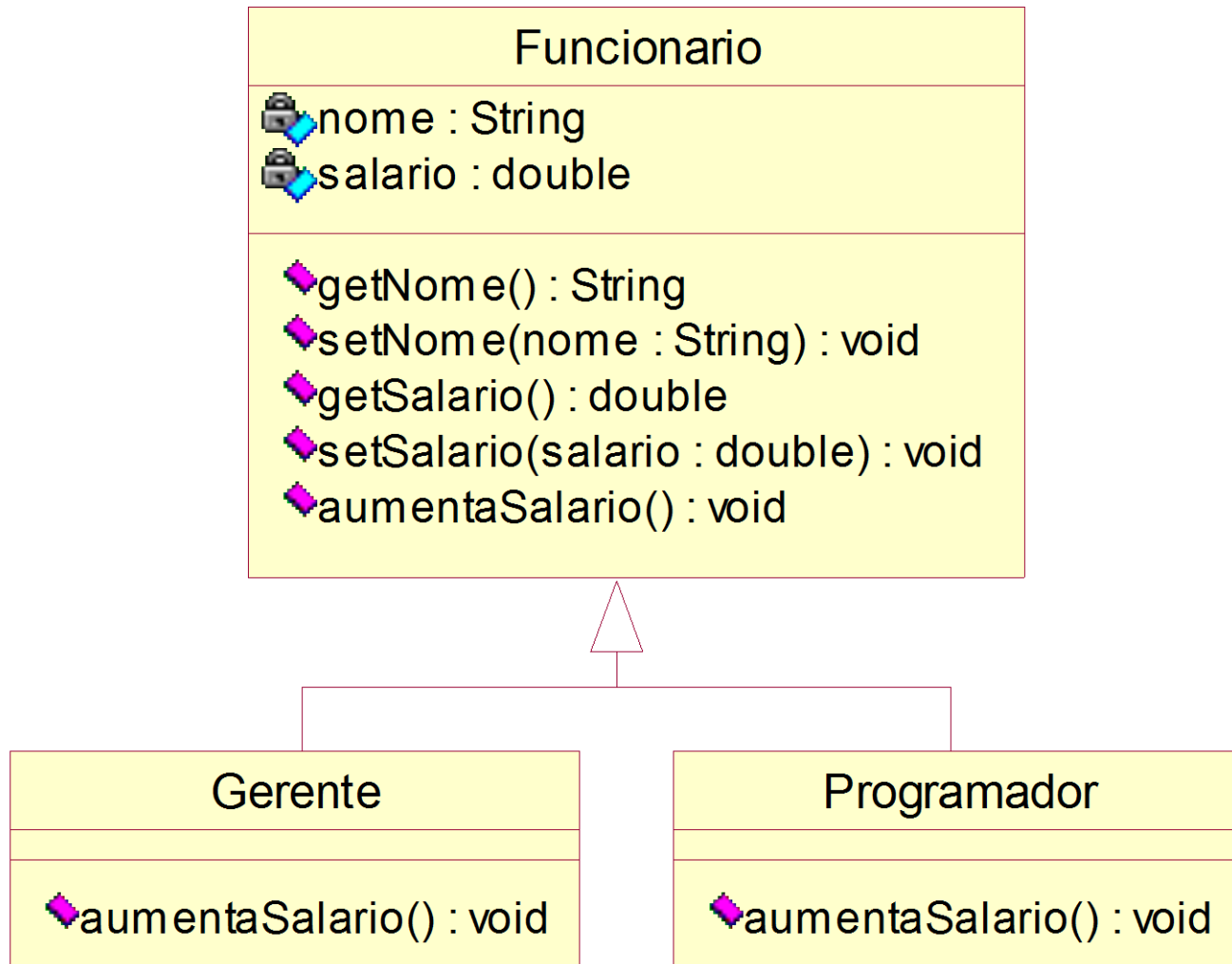
Sobreposição de Métodos

Universidade Católica de Pernambuco
Ciência da Computação

Prof. Márcio Bueno
pooite@marciobueno.com

Fonte: Material da Prof^a Karina Oliveira

Sobreposição de Métodos





Sobreposição de Métodos

■ Contextualizando

- Vamos supor agora que o método *aumentaSalario()* da classe *Funcionario* no exercício anterior possui uma implementação, ou seja, o método não é mais abstrato.
- E que as subclasses *Gerente* e *Programador* mantêm suas implementações para o método *aumentaSalario()*.



Sobreposição de Métodos

■ Definição

- Quando uma subclasse declara um método com o mesmo nome, mesmo tipo de retorno e mesma lista de parâmetros de um método da sua superclasse, dizemos que ocorreu uma **sobreposição de método** ou **redefinição de método** (*overriding*).
- Um método redefinido em uma subclasse oculta o método da classe ancestral a partir da subclasse.
- **Métodos privados não podem ser sobrepostos.**



Sobreposição de Métodos

■ Observação

- O nome e a lista de parâmetros de um método é chamado de **assinatura do método**. Contudo, o tipo de retorno não faz parte da assinatura.
- Em Java, ter métodos em uma superclasse e uma subclasse com a mesma assinatura, mas tipos diferentes de retorno vai resultar em um **erro em tempo de compilação**.

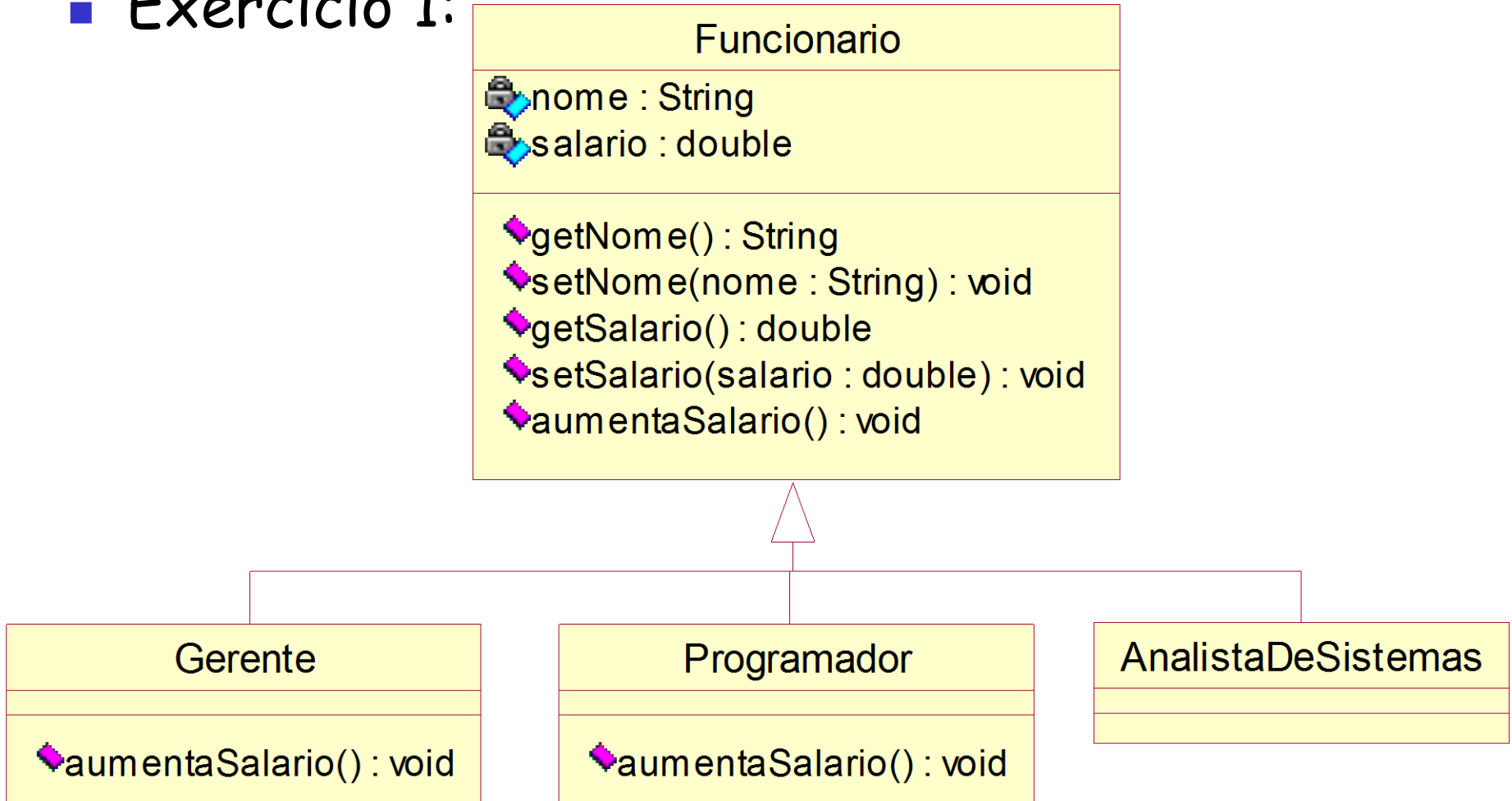


Sobreposição de Métodos

- **Sobreposição X Sobrecarga**
 - Sobrecarga (*overloading*) significa que pode-se ter métodos de mesmo nome, mas que difiram na lista de parâmetros.
 - Ou seja, métodos sobrecarregados não possuem mesma assinatura.

Sobreposição de Métodos

Exercício 1:





Sobreposição de Métodos

- Exercício 1 - Considerações:
 - A classe `Funcionario` é abstrata.
 - As classes `Gerente` e `Programador` são concretas.
 - Uma chamada ao `umentaSalario()` do `Funcionario` aumenta seu salário em 5%.
 - Uma chamada ao `umentaSalario()` do `Gerente` aumenta seu salário em 10%.
 - Uma chamada ao `umentaSalario()` do `Programador` aumenta seu salário em 20%.



Sobreposição de Métodos

- Exercício 1 - Aplicação:
 - Implemente uma aplicação que declara três variáveis do tipo `Funcionario` e cria três objetos um do tipo `Gerente`, outro do tipo `Programador` e o terceiro do tipo `AnalistaDeSistemas`. Em seguida, o programa deve oferecer um menu para o usuário com as seguintes opções:
 - Imprimir dados - O usuário deverá informar se ele deseja imprimir os dados do `Gerente`, do `Programador` ou do `AnalistaDeSistemas`.
 - Aumentar salário - O usuário deverá informar se ele deseja aumentar o salário do `Gerente`, do `Programador` ou do `AnalistaDeSistemas`.



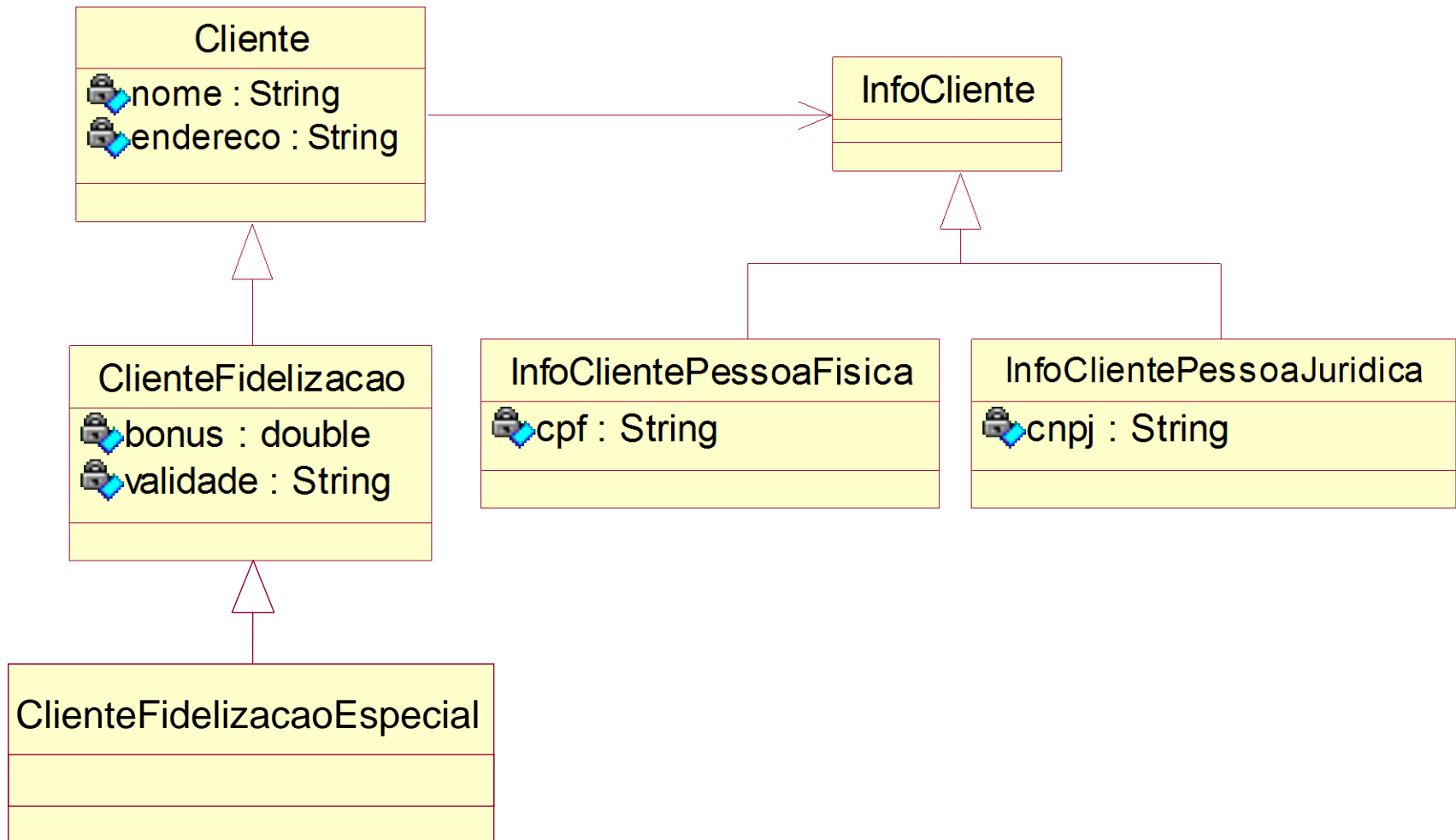
Sobreposição de Métodos

■ Observação

- A ligação entre a assinatura de um método e o método por ela designado efetua-se em tempo de execução. Este mecanismo é conhecido como ligação dinâmica - *dynamic binding*.

Sobreposição de Métodos

Exercício 2:





Sobreposição de Métodos

- Exercício 2:
 - Acrescentar a classe `ClienteFidelizacao` o método `adicionaBonus` que recebe o valor da compra realizada pelo cliente e soma ao bônus do cliente 5% do valor desta compra.
 - Suponha que agora, no sistema, existe mais um tipo de `Cliente`, o `ClienteFidelizacaoEspecial`, que apresenta as mesmas características do `ClienteFidelizacao` porém, ganha bônus de 10% em cima das compras realizadas.



Sobreposição de Métodos

- Exercício 3:

- Implemente uma classe *Conta* que contenha os atributos nome do cliente, número da conta, saldo e limite. Estes valores deverão ser informados no construtor, sendo que o limite não poderá ser maior que o valor do salário mensal do cliente. Implemente também um método depósito e outro método saque. O método saque retorna um *booleano* indicando se o saque pôde ser efetuado ou não.



Sobreposição de Métodos

- Exercício 3:
 - Implemente uma classe `ContaEspecial` que funciona da mesma forma que a classe `Conta`, mas que aceita um limite de até 3 vezes o valor do salário do cliente.