

Default methods

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Default methods (defender methods)

- A partir do Java 8, interfaces podem conter métodos concretos.
- A intenção básica é prover implementação padrão para métodos, de modo a evitar:
 - 1) repetição de implementação em toda classe que implemente a interface
 - 2) a necessidade de se criar classes abstratas para prover reuso da implementação
- Outras vantagens:
 - Manter a retrocompatibilidade com sistemas existentes
 - Permitir que "interfaces funcionais" (que devem conter apenas um método) possam prover outras operações padrão reutilizáveis

Problema exemplo

Fazer um programa para ler uma quantia e a duração em meses de um empréstimo. Informar o valor a ser pago depois de decorrido o prazo do empréstimo, conforme regras de juros do Brasil. A regra de cálculo de juros do Brasil é juro composto padrão de 2% ao mês.

Veja o exemplo.

Example

Amount: **200.00**
Months: **3**
Payment after 3 months:
212.24

| BrazilInterestService |
|---|
| - interestRate : double |
| + payment(amount : double, months : int) : double |

Calculations: $\text{Payment} = 200 * 1.02 * 1.02 * 1.02 = 200 * 1.02^3 = 212.2416$

$\text{Payment} = \text{amount} * (1 + \text{interestRate} / 100)^N$

<https://github.com/acenelio/interfaces5-java>

What if there was another interest service from another country?

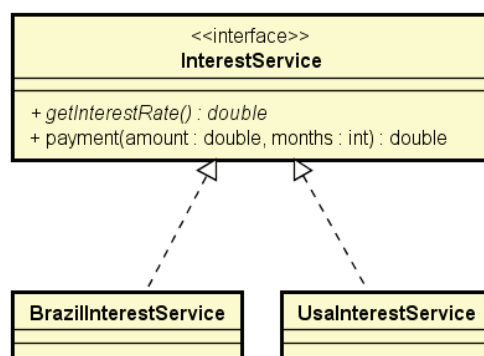
Amount: **200.00**
 Months: **3**
 Payment after 3 months:
 206.06

| UsaInterestService |
|---|
| - interestRate : double |
| + payment(amount : double, months : int) : double |

Calculations: $\text{Payment} = 200 * 1.01 * 1.01 * 1.01 = 200 * 1.01^3 = 206.0602$

$\text{Payment} = \text{amount} * (1 + \text{interestRate} / 100)^N$

<https://github.com/acenelio/interfaces5-java>



Considerações importantes

- Sim: agora as interfaces podem prover reuso
- Sim: agora temos uma forma de herança múltipla
 - Mas o compilador reclama se houver mais de um método com a mesma assinatura, obrigando a sobrescreve-lo
- Interfaces ainda são bem diferentes de classes abstratas. Interfaces não possuem recursos tais como construtores e atributos.