

The background features abstract green geometric shapes. On the left, a thin, elongated green triangle points downwards. On the right, a larger, more complex shape composed of several overlapping triangles in various shades of green (from light lime to dark forest green) is positioned. A thin, light gray line extends from the bottom left towards the right, passing behind the green shapes.

# Polimorfismo

Prof. Nelson Bellincanta Filho

# Revisão

- ▶ Herança
- ▶ Capacidade das classes expandirem-se a partir das classes existentes;
- ▶ Classe herdeira (subclasse)
  - ▶ Possui os mesmos atributos da superclasse;
  - ▶ Herda acesso aos métodos desta superclasse;
  - ▶ Pode acrescentar novos atributos e métodos (especialização).

# Revisão - Sobrecarga de Métodos

- ▶ Assinatura de um método é formada por:
  - ▶ Nome do método
  - ▶ Número de parâmetros
  - ▶ Tipos dos parâmetros (considerando a ordem)
- ▶ Métodos tenha o mesmo nome, mesmo número de parâmetros com os mesmos tipos (na ordem) têm a mesma assinatura

# Revisão - Sobrecarga de Métodos

- ▶ Tipos de sobrecarga de Métodos
  - ▶ Sobrecarga na mesma classe
    - ▶ Assinaturas diferentes;
  - ▶ Sobrecarga em classes herdeiras (sobrescrever método)
    - ▶ Assinaturas diferentes;
    - ▶ Assinaturas iguais.

# Revisão - Exemplo

Escreva um módulo para calcular a próxima parcela de um financiamento.

- ▶ Dados disponíveis:
  - ▶ S - Valor da primeira parcela
  - ▶ N - Número de parcelas
  - ▶ J - Percentual de juros mensal
- ▶ Cada nova parcela é sempre calculada em relação à anterior:

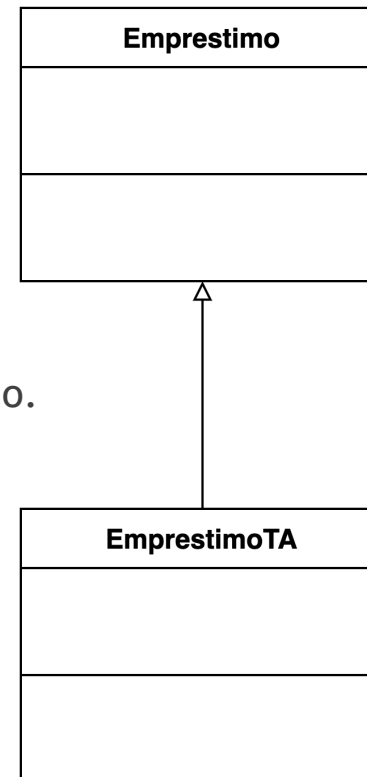
$$P_{\text{atual}} = P_{\text{anterior}} + \text{Juros}$$

## Exemplo: Empréstimo com taxa de administração.

Emprestimo
- n : int - j : float - corrente: int - p : float
+ <<constructor>> Emprestimo (s: float, n : int, j : float) + proximaParcela() : float

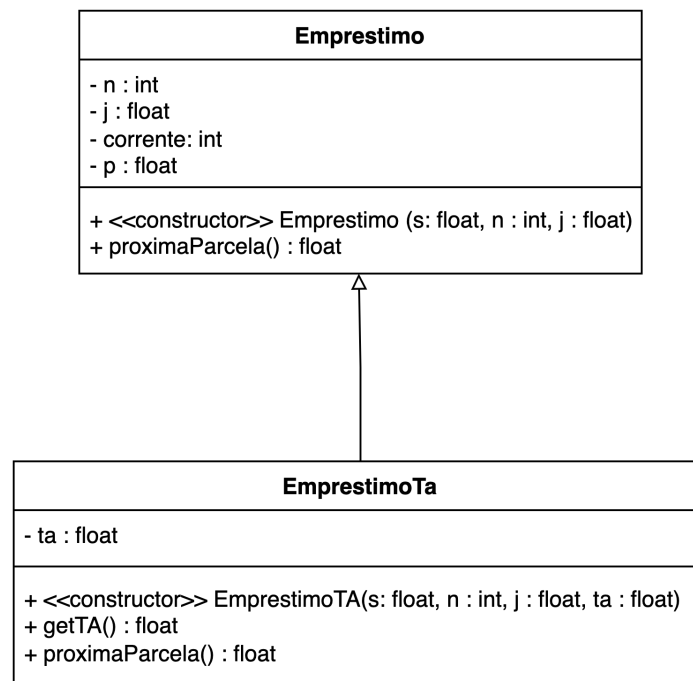
## Exemplo: Empréstimo com taxa de administração.

- Taxa de valor fixo adicionada sobre o valor do empréstimo.



# Exemplo: Empréstimo com taxa de administração.

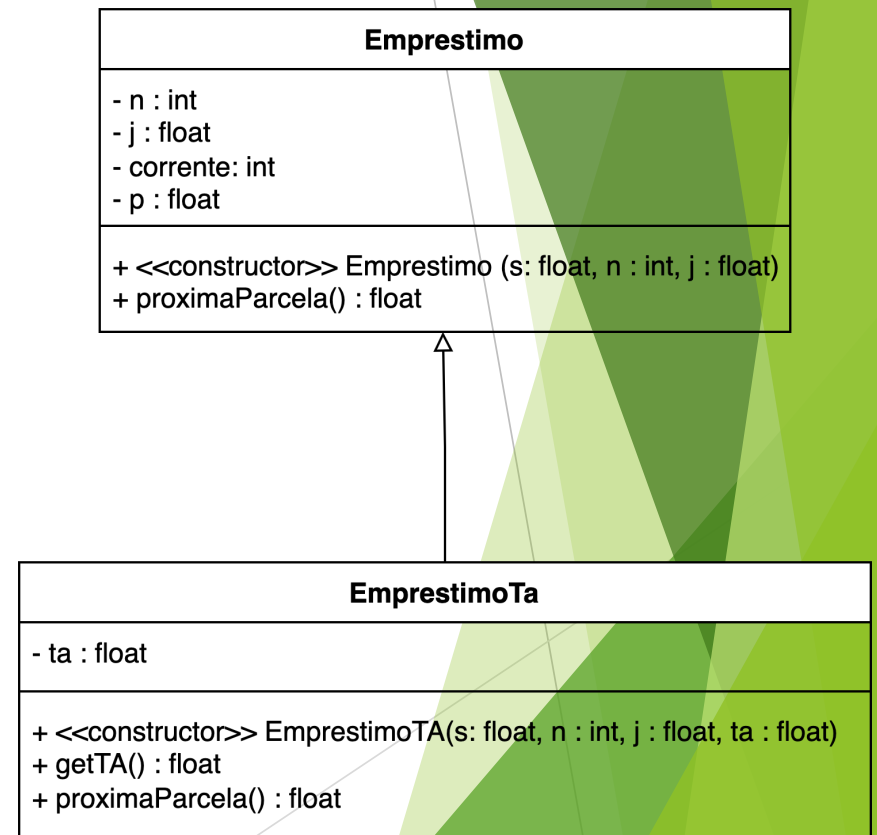
- Sobrecarga de proximaParcela() em EmpréstimoTA





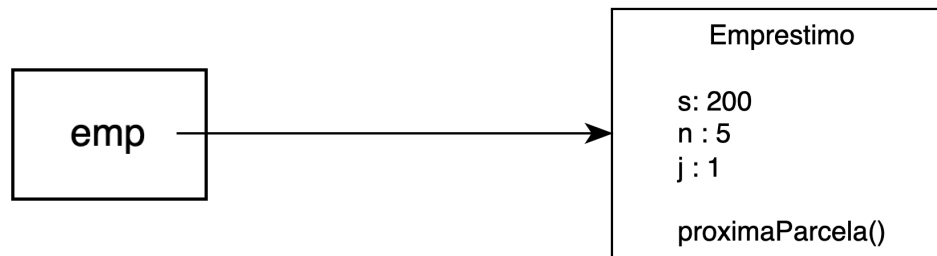
# Exemplo: Empréstimo com taxa de administração.

- ▶ Sobrecarga de proximaParcela() em EmpréstimoTA;
- ▶ Para um objeto da subclasse B, um método sobrescrito em B tem prioridade sobre o mesmo método da superclasse;
- ▶ Nesse caso, só estarão "visíveis" (respeitadas as regras de encapsulamento):
  - ▶ Atributos e métodos da superclasse A;
  - ▶ Métodos sobrescritos da subclasse B.
- ▶ Uma variável declarada em subclasse B não pode ser instanciada em sua superclasse



# Exemplo: Empréstimo com taxa de administração.

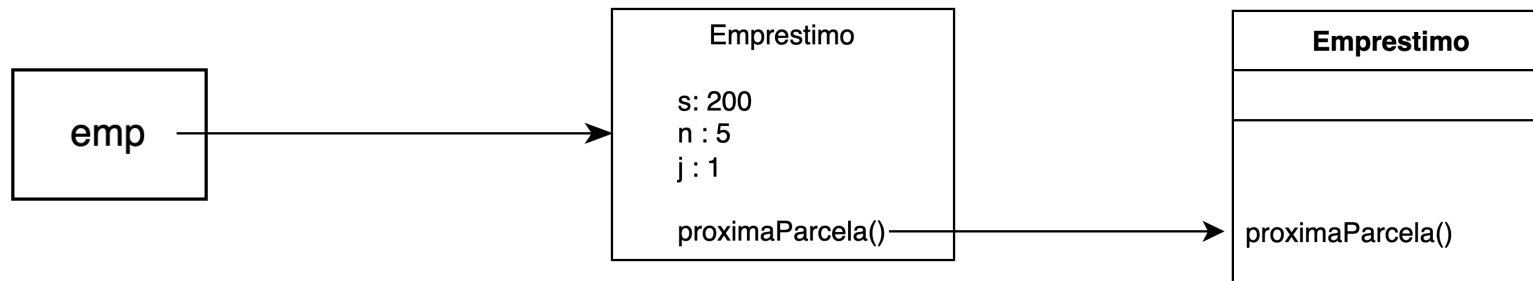
Objeto empréstimo



```
Emprestimo emp = new Emprestimo(200,5,1);
```

# Exemplo: Empréstimo com taxa de administração.

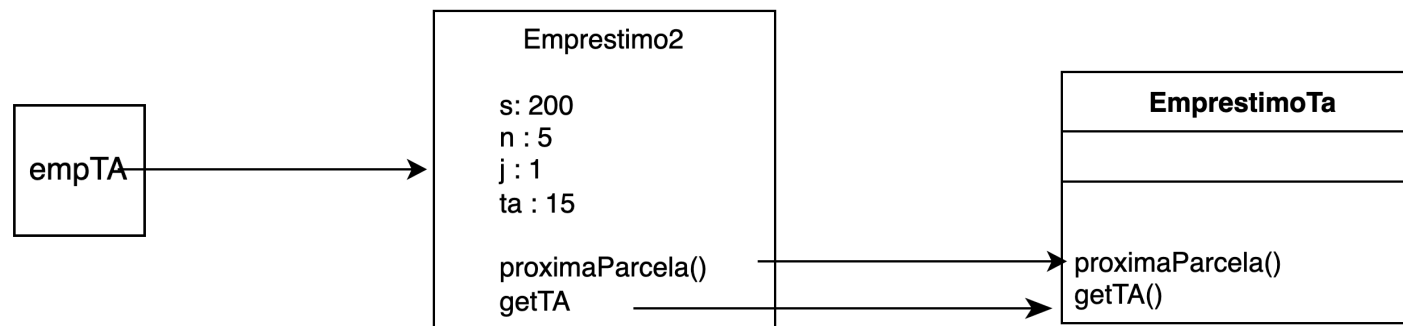
## Objeto empréstimo



```
Emprestimo emp = new Emprestimo(200,5,1);
```

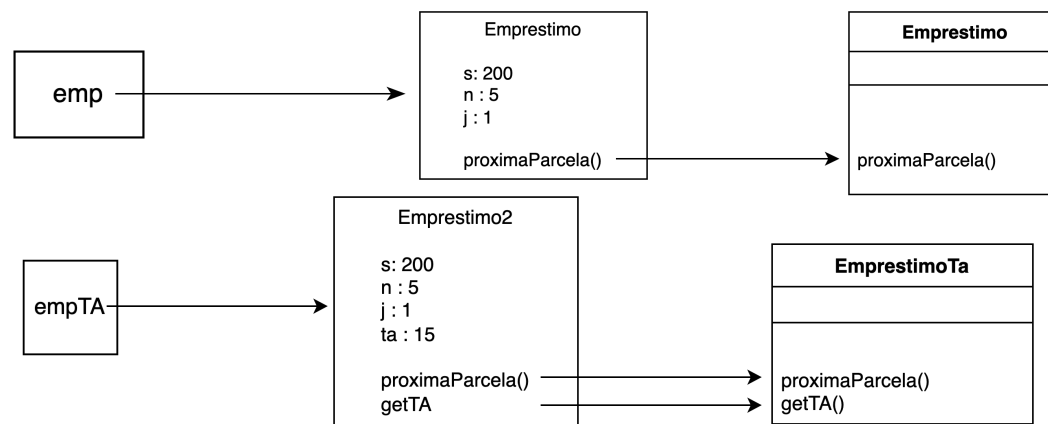
# Exemplo: Empréstimo com taxa de administração.

## Objeto EmpréstimoTA



```
EmpréstimoTA empTA = new EmpréstimoTA(200,5,1,15);
```

# Exemplo: Empréstimo com taxa de administração.



- ▶ `float pst = emp.proximaParcela();`
- ▶ `float pst = empTA.proximaParcela();`
- ▶ `float adm = empTA.getTA();`

Polimorfismo



# Polimorfismo

- ▶ O polimorfismo representa uma técnica avançada de programação, e seu uso pode gerar economia de recursos computacionais;
- ▶ A ideia geral do polimorfismo é que um objeto de uma determinada classe mais genérica (a superclasse) possa assumir diferentes comportamentos, gerando objetos distintos, dependendo de certas condições;
- ▶ Na prática quer dizer que um mesmo objeto pode executar métodos diferentes, dependendo do momento de sua criação.

# Polimorfismo

- ▶ O uso do polimorfismo pressupõe duas condições:
  - ▶ Existência de herança entre as classes;
  - ▶ Redefinição de métodos em todas as classes.
- ▶ Todas as classes devem possuir métodos com a mesma assinatura (nome e parâmetros), porém com funcionalidades diferentes;



# Princípio para o Polimorfismo

- ▶ Uma variável declarada em uma classe pode ser instanciada em qualquer classe;
- ▶ A declaração:

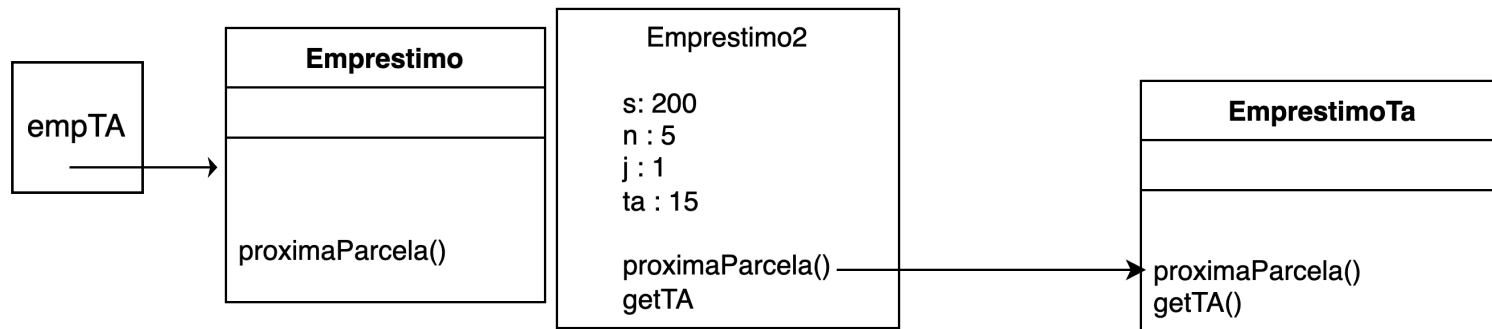
```
EmprestimoTA empTA = new EmprestimoTA(200,5,1,15);
```

- ▶ Também pode ser declarada como:

```
Emprestimo empTA = new EmprestimoTA(200,5,1,15);
```

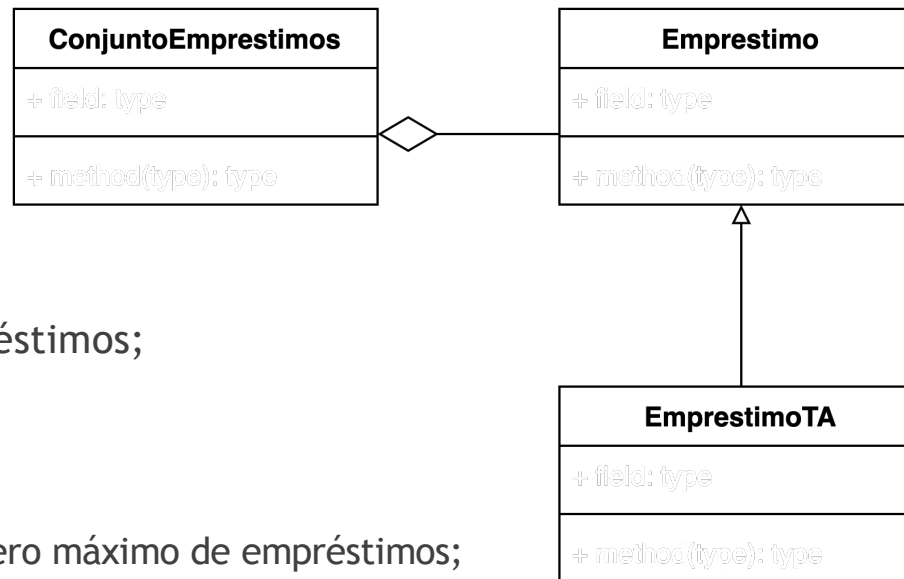
- ▶ Pergunta: Ao ser chamado o método proximaParcela(), qual dos dois será chamado, o da superclasse ou do herdeiro?

# Princípio para o Polimorfismo



```
Emprestimo empTA = new EmprestimoTa(200,5,1,15);
```

# Classe ConjuntoEmprestimos



- ▶ Controla um conjunto de empréstimos;
- ▶ Métodos:
  - ▶ construtor -> parâmetro: número máximo de empréstimos;
  - ▶ adicionaEmprestimo-> parâmetro: um objeto da classe empréstimo;
  - ▶ proximasParcelas -> mostra a próxima parcela de Todos os empréstimos cadastrados.

# Referências

FURGERI, Sérgio. Java 8: Ensino Didático. 1ª ed., São Paulo: Érica, 2015.

DEITEL, Harvey M.; DEITEL, Paul J. Java: como programar. 10ª ed. Editora Pearson Education do Brasil, 2017.

JUNIOR, Peter Jandl. **Java Guia do Programador**. 4a Edição: Atualizado para Java 16. Novatec Editora, 2021.