

Padrões de Projeto

Prof. Paulo Henrique Pisani

<http://professor.ufabc.edu.br/~paulo.pisani/>

novembro/2020

Tópicos

- Introdução
- Alguns padrões:
 - Singleton
 - Factory Method
 - Strategy

Introdução

- Padrões de projetos são soluções para problemas recorrentes envolvendo orientação a objetos.
- São apresentados 23 padrões no livro *Design Patterns: Elements of Reusable Object-Oriented Software*. O livro foi publicado por Erich Gamma, John Vlissides, Ralph Johnson, e Richard Helm - GoF (*gang of four*).

Organização dos 23 padrões GoF

- Propósito:
 - Padrões de Criação
 - Padrões Estruturais
 - Padrões Comportamentais
- Escopo:
 - Classe
 - Objeto

Alguns padrões

- Veremos alguns padrões sendo aplicados na prática:
 - Singleton (criação, objeto)
 - Factory method (criação, classe)
 - Strategy (comportamental, objeto)

Singleton

- **Padrão de criação**
- Garante que haverá apenas um instância da classe e fornece um método para acesso a essa instância.

Singleton

| Singleton |
|--|
| - <u>instanciaUnica: Singleton</u> |
| - Singleton() + <u>getInstanciaUnica(): Singleton</u> |

Exemplo

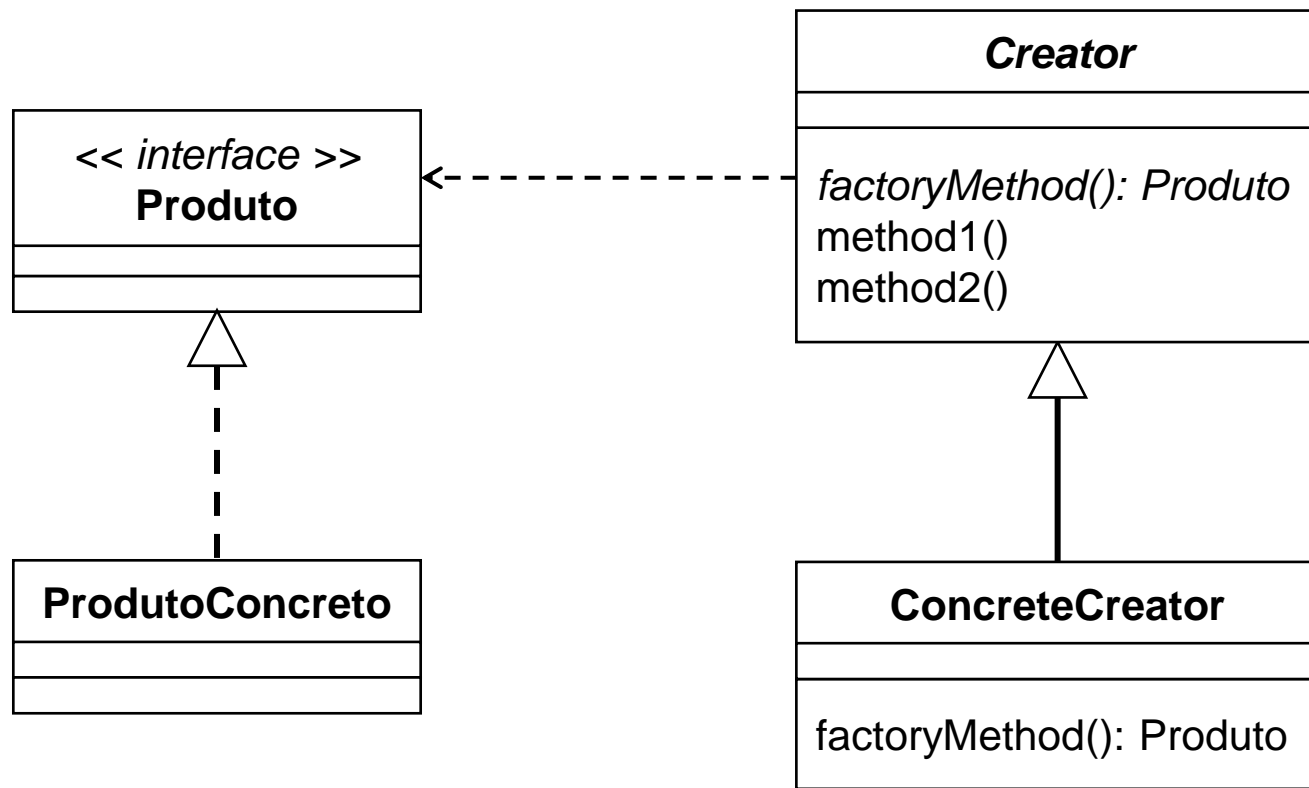
Vamos implementar a classe a seguir.

| LogSistema |
|--|
| <ul style="list-style-type: none">- <u>instanciaUnica: LogSistema</u>- registros: ArrayList<String> |
| <ul style="list-style-type: none">- LogSistema()+ <u>getInstanciaUnica(): LogSistema</u>+ registrar(evento: String)+ imprimir() |

Factory Method

- **Padrão de criação**
- Define um método para criar objetos concretos, mas as subclasses podem decidir qual classe instanciar (o método é implementado na subclasse).
- Separa o código que usa a classe daquele que instancia a classe.

Factory Method

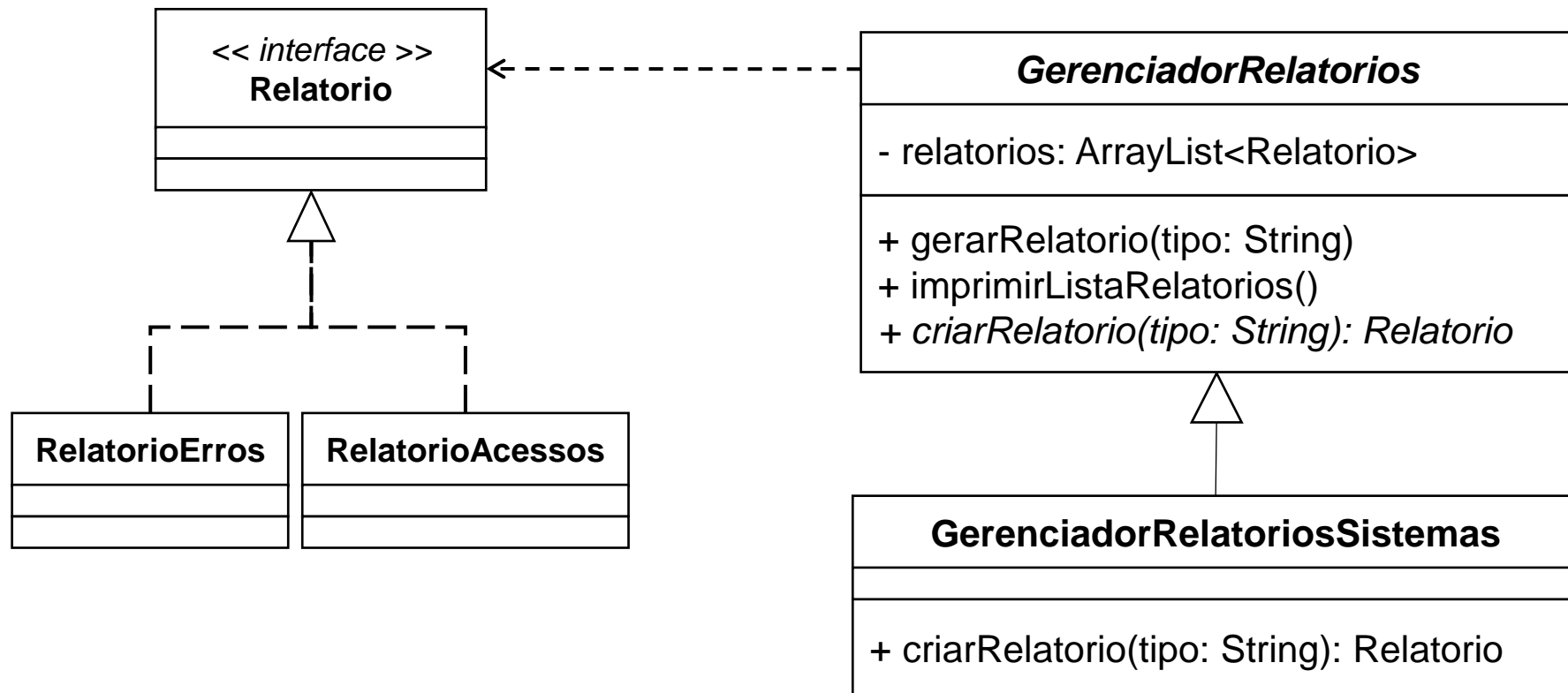


Método pode ter um parâmetro

Exemplo

- Vamos escrever um programa para gerenciar relatórios;
- Haverá dois tipos de relatórios:
 - Relatório de erros;
 - Relatório de acessos.
- A seguir, é apresentado como **Factory method** será aplicado e depois implementaremos em Java...

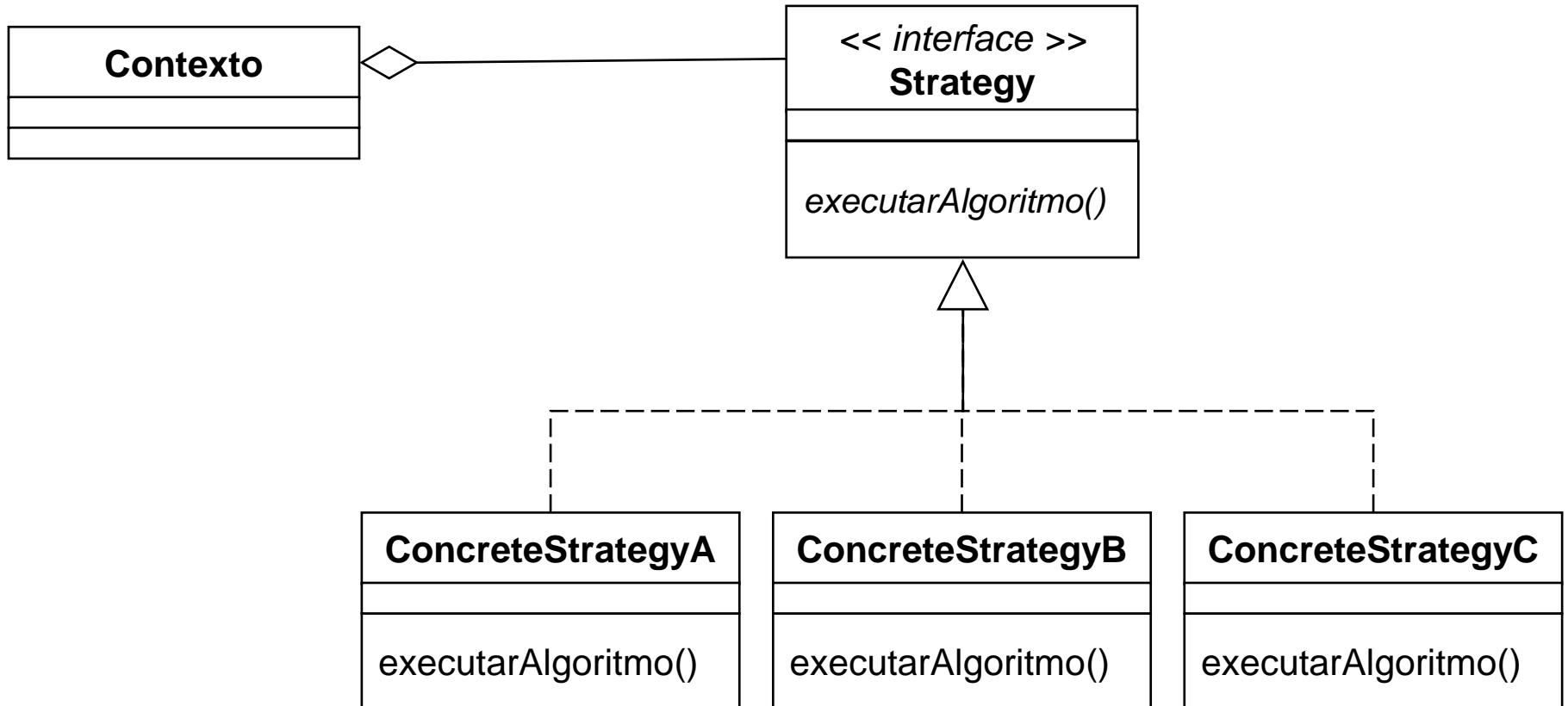
Exemplo



Strategy (Policy)

- **Padrão comportamental**
- Define uma família de algoritmos relacionados, cada um fica em uma classe que implementa uma interface *strategy* comum;
- A classe que usa a família de algoritmos referencia uma classe concreta que implementa a interface da *strategy*;
- Usa composição, permite mudar a implementação do algoritmo em tempo de execução.

Strategy

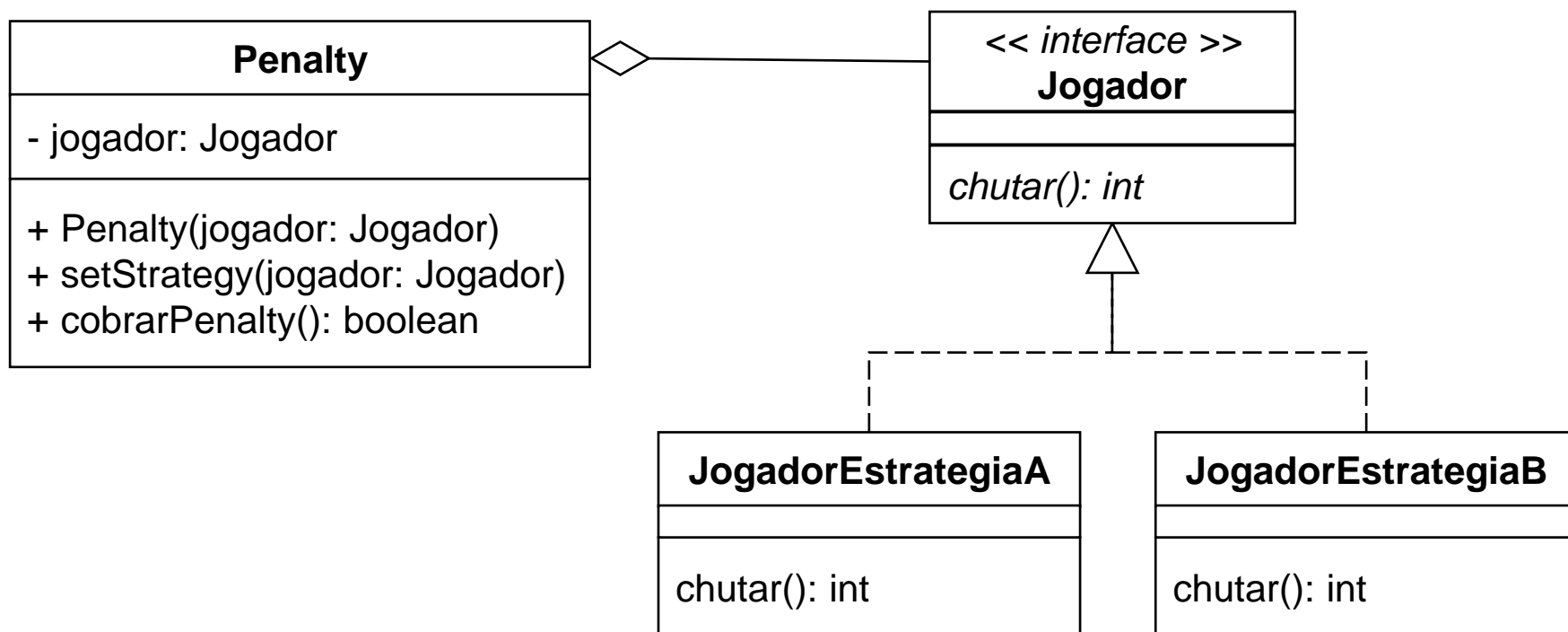


Exemplo

- Vamos escrever um programa para simular cobranças de penalty;
- Será considerado um cenário que o cobrador pode chutar em três lugares: esquerda, meio, direita. O goleiro também poderá pular para um desses três lugares.
- Se o goleiro escolher o mesmo lugar que o cobrador escolher, ocorre uma defesa. Caso contrário, é marcado um gol.

Exemplo

- Nesse exemplo, o comportamento do jogador cobrador pode mudar: chute aleatório ou sempre chutar no meio do gol.



Referências

- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos. Bookman, 2000.
- Refactoring guru:
<https://refactoring.guru/design-patterns/>

Bibliografia básica

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro, RJ: Campus, 2005.
- DEITEL, H. M.; DEITEL, P. J. Java: como programar. 6. ed. Porto Alegre, RS: Bookman, 2005.
- GUEDES, G. T. A. UML 2: uma abordagem prática. São Paulo, SP: Novatec, 2009.

Bibliografia complementar

- BARNES, D. J.; KOLLING, M. Programação orientada a objetos com Java. 4. ed. São Paulo, SP: Pearson Prentice Hall, 2009.
- BRUEGGE, B.; DUTOIT, A. H. Object-oriented software engineering: using UML, patterns, and Java. 2. ed. Upper Saddle River, NJ: Prentice Hall, 2003.
- FLANAGAN, D. Java: o guia essencial. 5. ed. Porto Alegre, RS: Bookman, 2006.
- FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre, RS: Bookman, 2005.
- LARMAN, C. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento interativo. 3. ed. Porto Alegre, RS: Bookman, 2007.