

Técnico em Desenvolvimento de Sistemas

Projeto de Software I

Juliano Lucas Gonçalves

juliano.goncalves@ifsc.edu.br / julianolg@gmail.com

Projeto de Software I

Agenda

Padrões de Projetos

- Conceitos
- Tipos

Conceitos Básicos

O que é um padrão de projetos?

Descrevem soluções para problemas recorrentes no desenvolvimento de sistemas orientado a objetos.

- possui um nome
- define o problema
- define a solução
- quando aplicar esta solução
- consequências

Conceitos Básicos

GoF (Gang of Four) (Total de 23 padrões)

Em engenharia de software, a Gangue dos quatro (Gang of Four, ou GoF) eram Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, autores do livro-charneira Design Patterns.

- Padrões de criação (“criacionais”)
- Padrões estruturais
- Padrões comportamentais

Padrões de Criação

PADRÕES DE CRIAÇÃO (“criacionais”) - 5 padrões

- Aplicam-se em situações que envolvem a criação de objetos.
- Ajudam a fazer um sistema independente de como seus objetos são criados, compostos e representados (Erich Gamma, et al.).

Padrões estruturais

PADRÕES ESTRUTURAIS – 7 padrões

- De que maneira as classes e objetos são compostos para a formação de grandes estruturas (Erich Gamma, et al.).

Padrões comportamentais

PADRÕES COMPORTAMENTAIS – 11 padrões

- Preocupam-se com algoritmos e atribuição de responsabilidades entre objetos. Descrevem, também, um padrão de comunicação entre classes ou objetos (Erich Gamma, et al.).

Padrões de Criação

- Eles disponibilizam uma maneira para criar objetos ocultando os detalhes da sua criação, ao invés de utilizar o operador **new** diretamente. Isto dá mais flexibilidade na hora de decidir quais objetos precisam ser criados para um determinado caso.

Padrões de Criação

Quais são eles:

- **Singleton**
- **Factory Method**
- Abstrat Factory
- Builder
- Prototype

Padrões de Criação

Singleton

Intenção: garantir que uma determinada classe tenha uma, e somente uma instância.

Padrões de Criação

Singleton (Exemplo)

```
1  class Usuario{
2      static private $instance = null;
3      private $nome;
4      private $senha;
5      private function __construct(){
6
7      }
8
9      static function getInstance(){
10         if (self::$instance == null){
11             self::$instance = new Usuario;
12         }
13         return self::$instance;
14     }
```

Padrões de Criação

Factory Method

- **Intenção:** concentrar a maneira como os nossos objetos são criados em um mesmo lugar e com isso se uma nova regra surgir em relação a criação desses objetos alteramos em apenas um local. Para isso criamos uma classe central que fica responsável por criar nossos objetos.

Padrões de Criação

Factory Method (Exemplo)

```
1 interface ICarro{
2     function Ligar();
3 }
4
5 class Gol implements ICarro{
6     function Ligar(){echo "Gol Ligado !!!!"; }
7 }
8
9 class Palio implements ICarro{
10     function Ligar(){echo "Palio Ligado !!!!"; }
11 }
```

Padrões de Criação

Factory Method (Exemplo)

```
1 class FactoryCar{  
2     static function CriarCarro($marca){  
3         switch ($marca){  
4             case "Gol": $obj = new Gol;  
5                         $obj->Ligar();  
6             break;  
7             case "Palio": $obj = new Palio;  
8                          $obj->Ligar();  
9         }  
10    }  
11 }
```

```
1 | FactoryCar::CriarCarro("Palio"); ou FactoryCar::CriarCarro("Gol");
```

Padrões de Criação

Abstract Factory

- **Intenção:** – Intenção: fornecer uma interface para criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas.
- Cria uma instância de várias famílias de classes.

Padrões de Criação

Abstract Factory (exemplo)



Padrões de Criação

Abstract Factory (exemplo)

Imagine que você está criando um simulador de loja de móveis. Seu código consiste de classes que representam:

1. Uma família de produtos relacionados, como: `Cadeira` + `Sofá` + `MesaDeCentro`.
2. Várias variantes dessa família. Por exemplo, produtos `Cadeira` + `Sofá` + `MesaDeCentro` estão disponíveis nessas variantes: `Moderno`, `Vitoriano`, `ArtDeco`.

	Cadeira	Sofá	Mesa de Centro
Art Deco			
Vitoriano			
Moderno			

Padrões de Criação

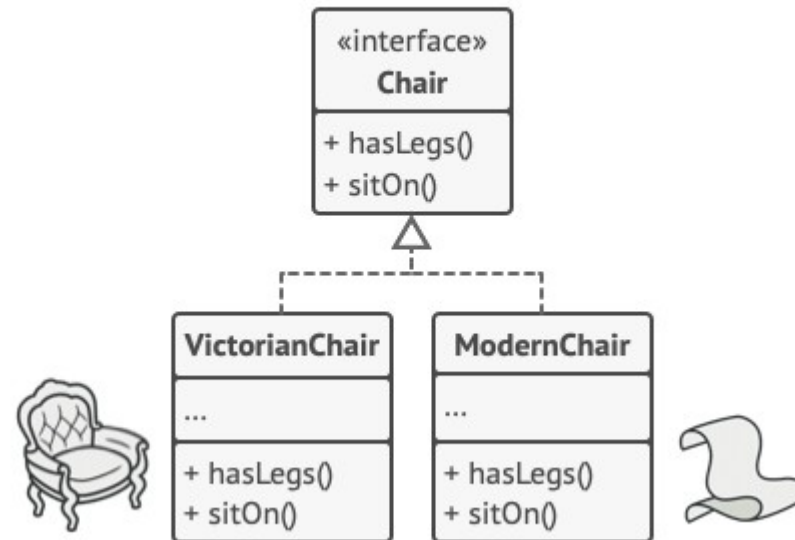
Abstract Factory (exemplo)



Um sofá no estilo Moderno não combina com cadeiras de estilo Vitoriano

Padrões de Criação

Abstract Factory (exemplo)



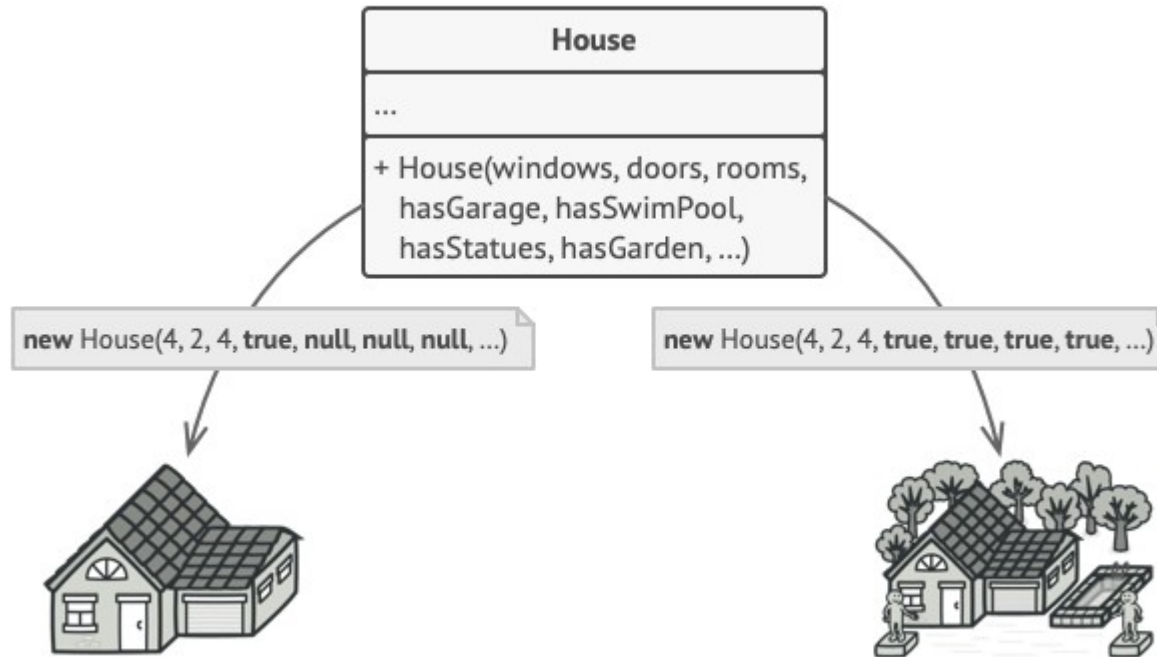
Padrões de Criação

Builder

- Intenção: separar a construção de um objeto complexo da sua representação de modo que o mesmo processo de construção possa criar diferentes representações.
- Separa a construção do objeto de sua representação

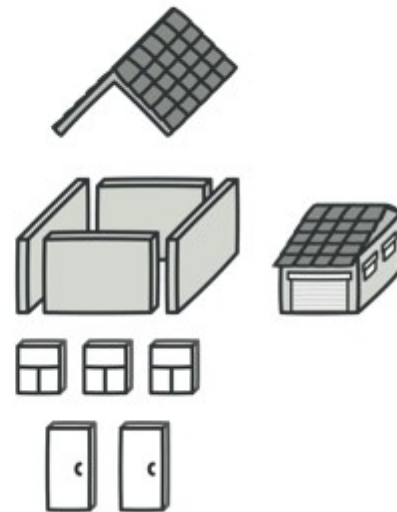
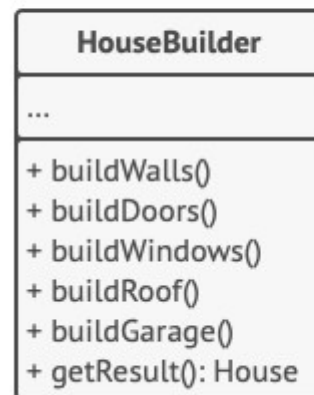
Padrões de Criação

Builder (exemplo)



Padrões de Criação

Builder (exemplo)



Padrões de Criação

Builder (exemplo)



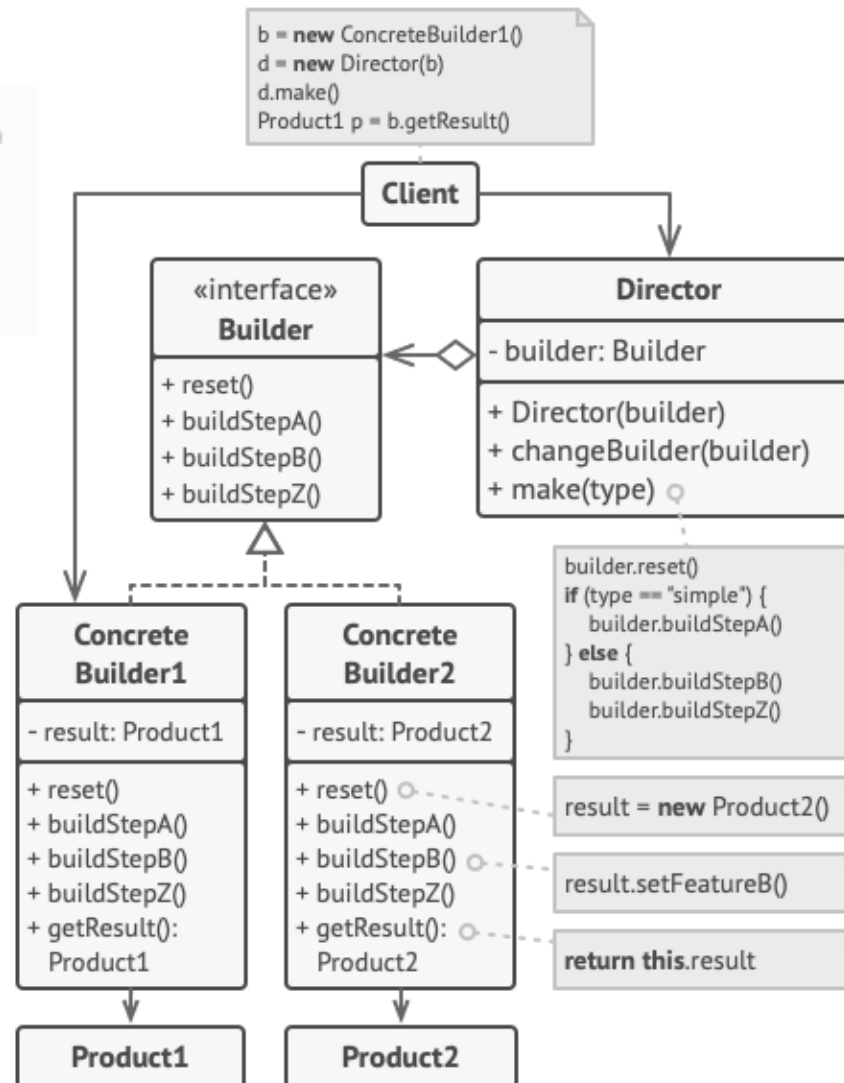
Padrões de Criação

Builder (exemplo)

1 A interface **Construtor** declara etapas de construção do produto que são comuns a todos os tipos de construtoras.

2 **Construtores Concretos** provêm diferentes implementações das etapas de construção. Construtores concretos podem produzir produtos que não seguem a interface comum.

3 **Produtos** são os objetos resultantes. Produtos construídos por diferentes construtores não precisam pertencer a mesma interface ou hierarquia da classe.



4 A classe **Diretor** define a ordem na qual as etapas de construção são chamadas, então você pode criar e reutilizar configurações específicas de produtos.

5 O **Cliente** deve associar um dos objetos construtores com o diretor. Usualmente isso é feito apenas uma vez, através de parâmetros do construtor do diretor. O diretor então usa aquele objeto construtor para todas as futuras construções. Contudo, há uma abordagem alternativa para quando o cliente passa o objeto construtor ao método de produção do diretor. Nesse caso, você pode usar um construtor diferente a cada vez que você produzir alguma

Padrões de Criação

Prototype

- Intenção: especificar os tipos de objetos a serem criados usando uma instância protótipo e criar novos objetos pela cópia deste protótipo.
- Uma instância inicializada a ser copiada ou clonada.

Padrões de Criação

Prototype (exemplo)



Problema: atributos privados

Padrões de Criação

Prototype (exemplo)

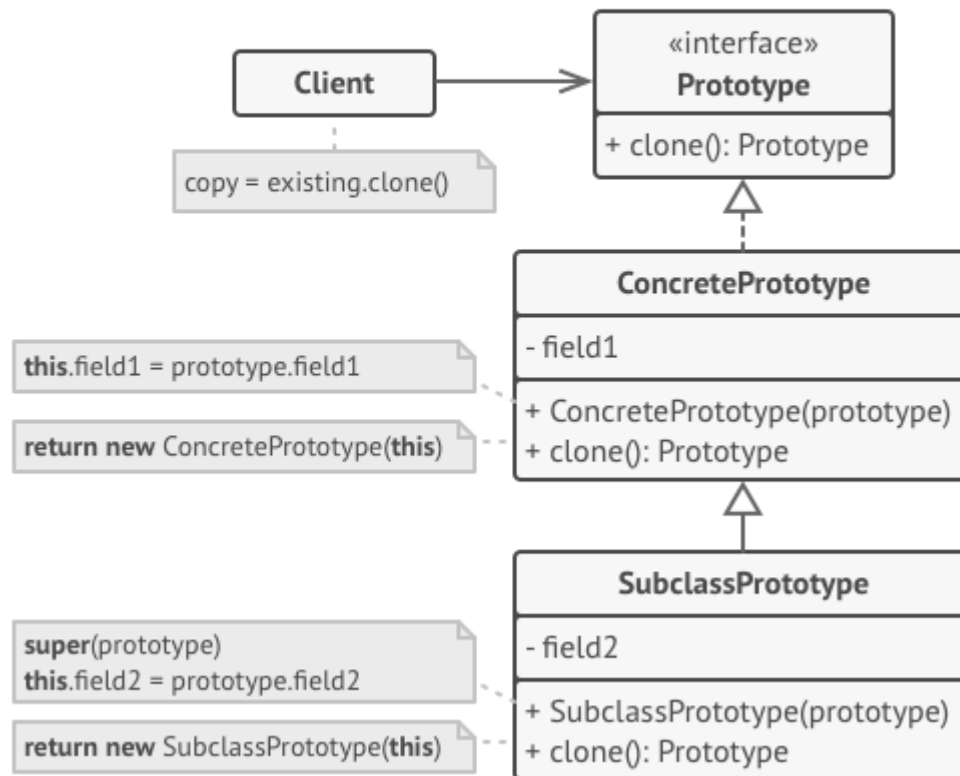


Padrões de Criação

Prototype (exemplo)

3 O **Cliente** pode produzir uma cópia de qualquer objeto que segue a interface do protótipo.

1 A interface **Protótipo** declara os métodos de clonagem. Na maioria dos casos é apenas um método `clonar`.



2 A classe **Protótipo Concreta** implementa o método de clonagem. Além de copiar os dados do objeto original para o clone, esse método também pode lidar com alguns casos específicos do processo de clonagem relacionados a clonar objetos ligados, desfazendo dependências recursivas, etc.