

Builder Pattern

Padrões GoF (Gang of Four)

- Padrões de criação
- Padrões estruturais
- Padrões comportamentais

<u>Criação</u>	<u>Estrutural</u>	<u>Comportamental</u>	
Abstract Factory	Adapter	Chain of Responsibility	Observer
Builder	Facade	Command	State
Factory Method	Bridge	Interpreter	Strategy
Prototype	Decorator	Iterator	Template Method
Singleton	Flyweight	Mediator	Visitor
	Composite	Memento	
	Proxy		

Definição do padrão builder

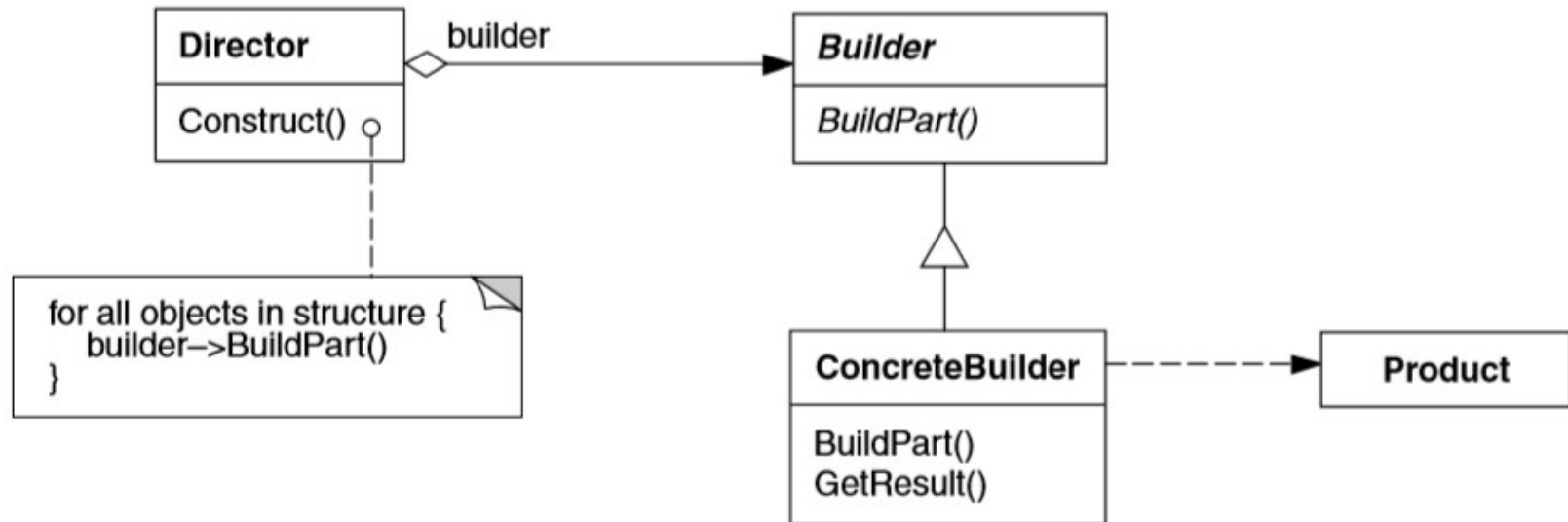
- O padrão Builder tem por intenção, segundo o livro do GoF, “separar a construção de um objeto complexo da sua representação de modo que o mesmo processo de construção possa criar diferentes representações”.
- Traduzindo: a responsabilidade de se criar um objeto é delegada para outro objeto (o Builder), que cria o objeto-produto parte por parte. Aliás, construir um objeto parte por parte é o foco do padrão e o que o difere dos outros padrões de criação.

Aplicabilidade

Use o padrão Builder quando :

- o algoritmo para criação de um objeto complexo deve ser independente das partes que compõem o objeto e de como elas são montadas.
- o processo de construção deve permitir diferentes representações para o objeto que é construído.

Estrutura do Builder



- **Director** utiliza a interface de Builder para construir um determinado objeto. Em outras palavras, criamos um Director e dizemos a ele qual Builder concreto iremos utilizar para construir o produto.
- **Builder** especifica uma interface para criação das partes do produto. Pode ser implementado como interface ou como classe-abstrata. No diagrama, temos apenas “BuildPart”, mas na prática temos vários métodos, cada um criando uma parte do produto.
- **ConcreteBuilder** implementa uma forma de construir o produto, parte por parte, seguindo a abstração Builder. ConcreteBuilder guarda a representação do objeto-produto e fornece um método para recuperá-lo, após sua construção (no diagrama, este método é o “getResult”).
- **Product** é o objeto-produto da construção “orquestrada” pelo Builder.



Exemplo 3

Participantes

As classes e objetos que participam desse padrão são:

- **Construtor (VehicleBuilder)**
 - especifica uma interface abstrata para criar partes de um objeto Produto
- **ConcreteBuilder (MotorCycleBuilder, CarBuilder, ScooterBuilder)**
 - constrói e monta partes do produto implementando a interface do Builder
 - define e acompanha a representação que cria
 - fornece uma interface para recuperar o produto
- **Diretor (Loja)**
 - constrói um objeto usando a interface do Construtor
- **Produto (veículo)**
 - representa o objeto complexo em construção. ConcreteBuilder constrói a representação interna do produto e define o processo pelo qual é montado
 - inclui classes que definem as peças constituintes, incluindo interfaces para montagem das peças no resultado final



Exemplo 4