

Resumo de arquitetura de software - pt VII

Padrão de projeto - BUILDER

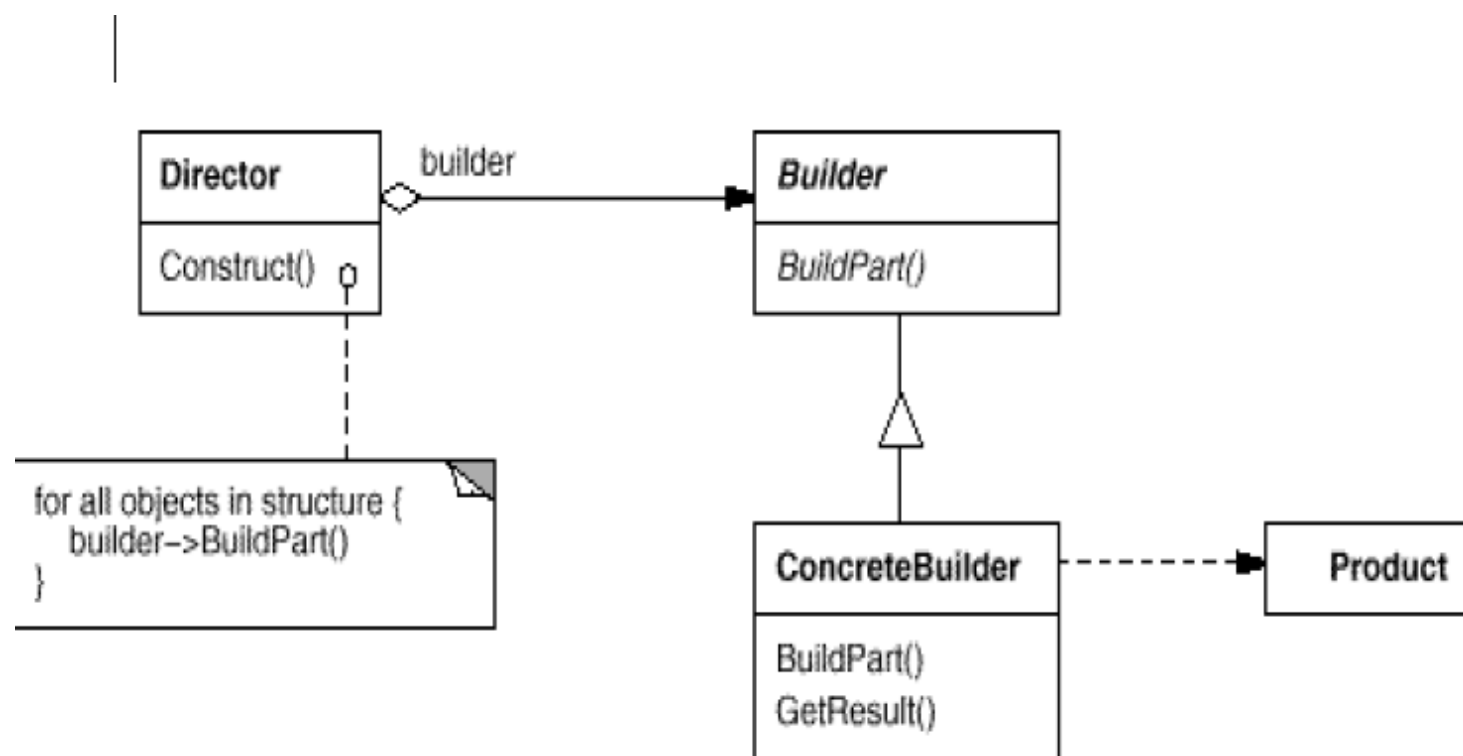
Projetos criacionais?

- Abstraem o processo de instanciação
 - Como é criado , compostos, representados
 - Importante a medida que o sistema evolue
 - Permite configurar um sistema com objetos que variam em estrutura e funcionalidade
-

Builder?

- **Intenção:**
 - Separar a construção de um objeto complexo da sua representação de modo que o processo de construção possa criar diferentes representações
- Aplicabilidade :

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



Participantes

- **Builder** (TextConverter)
 - Interface abstrata para criação de partes de um produto
- **ConcreteBuilder** (ASCIIConverter, TeXConverter, TextWidgetConverter)
 - Constrói e monta partes de um produto
- **Director** (RTFReader)
 - Constrói um objeto usando a interface de Builder
- **Product** (ASCIIText, TeXText, TextWidget)
 - Objeto complexo em construção e partes constituintes

Consequências

1. Permite variar a representação interna de um produto
2. Isola o código para construção e representação
3. Oferece um controle mais fino sobre o processo de construção

Padrões Relacionados

- **Abstract Factory** é semelhante ao Builder no sentido de que também constrói objetos complexos
 - A diferença principal é que o padrão Builder focaliza a **construção** de um objeto complexo **passo a passo**
 - A ênfase do Abstract Factory é sobre **famílias de objetos**
- Um **Composite** é o que frequentemente o Builder constrói

O que posso melhorar?

-