



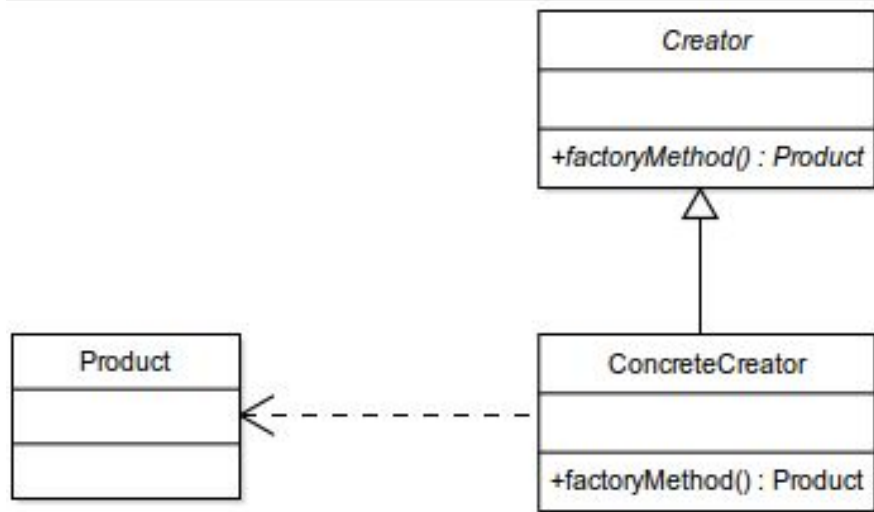
# Padrão Factory Method

Padrão de criação



# FactoryMethod

- *"Definir uma interface para criar um objeto mas deixar que subclasses decidam que classe instanciar."* (GoF)
- Separação na criação de objetos
- Objetos são criados sem saber qual sub-tipo será instanciado (uso de polimorfismo)
- Princípio
  - Criador concreto é chamado apenas quando necessário
  - Objeto atribuído para variável com tipo da superclasse (ou interface)
- Exemplos
  - Instanciação de editores de IDEs



Fonte: wikipedia

# Definição e uso

```
interface IEditor {
    public void load();
    public void save();
}

classe EditorTexto implements IEditor {
    public void load(){ /*implementar*/ }
    public void save(){ /*implementar*/ }
}

classe EditorJava implements IEditor {
    public void load(){ /*implementar*/ }
    public void save(){ /*implementar*/ }
}

interface ICriador {
    public Editor createEditor();
}

classe CriadorTexto implements ICriador {
    static public Editor createEditor() {
        new EditorTexto();
    }
}

classe CriadorJava implements ICriador {
    static public Editor createEditor() {
        new EditorJava()
    }
}
```

```
IEditor editorJava = CriadorJava.createEditor();
IEditor editorTexto = CriadorTexto.createEditor();
```

# Factory Method

- Vantagem
  - Uso do polimorfismo para instanciação dos criadores concretos
  - Possibilidade de utilização de API reflexiva
- Desvantagens
  - Extensão da classe
  - Construtor privado

Figura 1.1 do livro

