

Curso: Sistemas de Informação	
Disciplina: Padrões de projeto	Período: 8
Professor: Luiz Gustavo Dias	Tipo: Design Patterns
Objetivo: Factory Method	

DESIGN PATTERNS: FACTORY METHOD

Intenção

Definir uma interface para criar um objeto, mas deixas as subclasses decidirem que classe instanciar. O Factory Method permite adiar a instanciação para as subclasses.

Em outras palavras, tem-se uma classe com um método que cria um objeto, que pode (ou não) ser abstrato, e, sendo abstrato, força-se as subclasses a implementá-lo.

Dentro da programação, tudo que possui o nome Factory (fábrica) utiliza operações para a criação de objetos.

Sobre o padrão:

- É um padrão de projeto de criação (lida com a criação de objetos);
- Oculta a lógica de instanciação do código do cliente. O método fábrica será responsável por instanciar as classes desejadas;
- É obtido através de herança, podendo ser criado ou sobrescrito por subclasses;
- Dá flexibilidade ao código cliente permitindo a criação de novas factories sem alterar as já existentes, o que garante a aplicação do Open/Closed Principle (OCP) – aberto para extensão/fechado para alteração;
- Pode utilizar parâmetros para determinar o tipo dos objetos a serem criados ou os parâmetros a serem enviados aos objetos sendo criados.

Motivação

Criar um objeto geralmente requer processos complexos não apropriados para incluir dentro da composição do objeto. A criação do objeto talvez necessite de uma duplicação de código significativa, talvez necessite informações não acessíveis para a composição do objeto, talvez não providencie um grau de abstração suficiente, ou então não faça parte da composição das preocupações do objeto. O Design Pattern

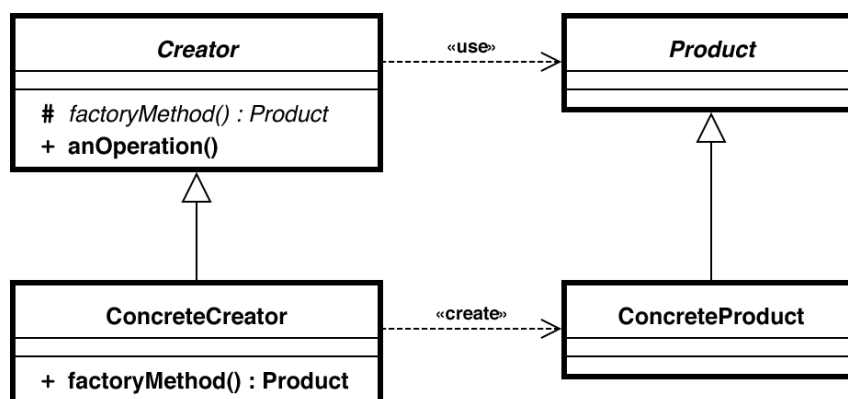


Factory Method maneja/trata esses problemas definindo um método separado para criação dos objetos, no qual as subclasses possam sobrescrever para especificar o "tipo derivado" do produto que vai ser criado.

Aplicabilidade

- Use o Factory Method quando não souber com certeza quais os tipos de objetos o seu código vai precisar;
- Use o Factory Method para permitir a extensão de suas factories para a criação de novos objetos (OCP);
- Use o Factory Method para separar o código que cria do código que utiliza as classes (Single Responsibility Principle);
- Use o Factory Method para dar um hook (gancho) às subclasses para permitir que elas decidam a lógica de criação de objetos;
- Use o Factory Method para eliminar duplicação de código na criação de objetos.

Estrutura



Product: produto abstrato, define uma interface para os objetos criados pelo factory method.

ConcreteProduct: produto concreto, uma implementação para a interface produto.

Creator: criado abstrado, declara o Factory Method que retorna o objeto da classe Product. Este elemento também pode definir uma implementação básica que retorna o objeto de uma classe ConcretProduct básica;

Concret Creator: criador concreto, sobrescreve o Factory Method e retorna um objeto da classe ConcretProduct.

Consequências

Positivas:

1. Ajuda a aplicar o Open/Closed Principle, mantendo seu código aberto para extensão;
2. Ajuda a aplicar o Single Responsibility Principle, pois separa o código que cria do código que usa o objeto;
3. Ajuda no desacoplamento do código.

Negativas:

1. Se usado sem necessidade, haverá uma explosão de subclasses. Será necessário uma classe ConcreteCreator para cada ConcreteProduct. Neste cenário, avalie a utilização de outros padrões em substituição ao Factory Method.