Fatec
Araras

Tópicos Especiais em Sistemas para Internet III

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Agenda

- Orientação a objetos
  - Frameworks
  - Padrões de Projetos
    - Criação, Estruturais e Comportamentais
    - Singleton
    - Factory
- Ambientes Virtuais (virtualenv)

Framework

Fatec Araras

Um framework é um conjunto de classes cooperantes que constroem um projeto reutilizável para uma específica classe de software. [Deu89, JF88]

Você customiza um framework, para uma aplicação específica, atraves da criação de subclasses específicas para a aplicação, sendo essas subclasses específicas das classes abstratas do framework.

Os frameworks sempre tem um particular domínio de aplicação.

Design Patterns

Fatec Araras

Projetar software orientado a objetos é difícil, mas projetar software reutilizável orientado a objetos é ainda mais difícil. O seu projeto deve ser específico para resolver um problema, mas genérico o suficiente para atender futuros requisitos.

Leva um longo tempo para os novatos aprenderem o que é um bom projeto orientado a objetos.

Projetistas experientes entendem que devem reutilizar soluções que funcionaram no passado.

Design Patterns

Como registrar a experiência adquirida em projetos orientado a objetos ? Por meio de padrões de projeto.

O objetivo dos padrões de projeto é tornar mais fácil a reutilização em novos projetos e arquiteturas bem-sucedidas.

Nenhum dos padrões de projeto descreve projetos novos ou não-testados.

Design Patterns

Em geral, um padrão tem quatro elementos essenciais:

Nome do padrão: uma referência para descrever um problema de projeto.

O **problema** descreve quando aplicar o padrão.

A **solução** descreve os elementos que compõem o projeto.

As **consequências** são os resultados e análises das vantagens e desvantagens (trade-offs) de aplicação do padrão

Frameworks e Design Patterns

Fatec Araras

Ambos possuem algumas similaridades, e as pessoas frequentemente se perguntam no que eles diferem:

- Os padrões de projeto são mais abstratos que frameworks.Frameworks podem ser materializados em código, mas somente exemplos de padrões podem ser materializados em código.
- Padrões de projetos são elementos de arquiteturas menores que frameworks. Um framework típico contém vários padrões de projeto, mas a recíproca nunca é verdadeira.

Frameworks e Design Patterns

Fatec Araras

- Padrões de projetos são menos especializados que framework. Os frameworks sempre tem um particular domínio de aplicação. Em contraste, os padrões de projeto podem ser usados em quase qualquer tipo de aplicação.

Padrões do livro da GoF

Fatec Araras

O livro da GoF sobre padrões de projeto discute 23 padrões e os classifica em três categorias principais:

- Padrões de Criação
- Padrões Estruturais
- Padrões Comportamentais

A classificação dos padrões é feita principalmente com base na forma como os objetos são criados, as classes e os objetivos são estruturados em uma aplicação de software e inclui também a forma como os objetos interagem entre si.

Padrões de Criação

- Funcionam com base no modo como os objetos podem ser criados
- Isolam detalhes da criação dos objetos
- O código é independente do tipo de objeto a ser criado.

Padrões Estruturais

- Eles determinam o design da estrutura de objetos e classes para que estes possam ser compostos e o resultado mais aplos sejam alcançados.
- O foco está em simplificar a estrutura e identificar o relacionamento entre classes e objetos
- Estão centrados em herança e composição de classes

Padrões Comportamentais

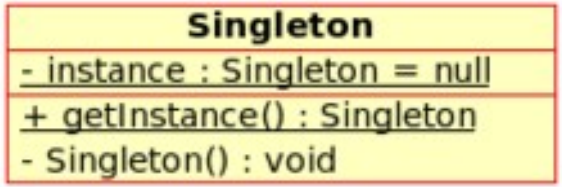
- Estão preocupados com a interalçao entre os objetos e suas responsabilidades
- Os objetos devem ser capazes de interagir e-mesmo assim ter baixo acoplamento.

Singleton

Singleton

- O Singleton proporciona uma forma de ter um e somente um objeto de determinado tipo, além de disponibilizar um ponto de acesso global.
- Por isso, os singleton são geralmente usados como logging ou operadores de banco de dados, spoolers de impressão e muitos outros cenários em que seja necessário que haja apenas uma instância disponível.
- **código: singleton.py e singleton2.py**

Singleton



Fonte: https://upload.wikimedia.org/wikipedia/commons/b/bd/Singleton.png

Monostate

- Segundo GoF, deve haver somente um e somente um objeto por classe. Entretanto, de acordo com Alex Martelli, o que o programador geralmente precisa é de instâncias que compartilhem o mesmo estado.

  código: monostate.py

Singleton com metaclasses

Fatec Araras

- Uma metaclasse é uma classe de outra classe, o que significa que a classe é uma instância de sua metaclasse. Com as metaclasses, os programadores têm a oportunidade de criar seus próprios tipos a partir de classes Python predefinidas.
- A definição de classe é determinada por sua metaclasse, portanto, quando criamos uma classe com class A, Python a cria com
- A = type(name, base, dict)

código: singleton3.py

Singleton Desvantagens

Fatec Araras

- Variáveis globais podem ser alteradas por engano em um lugar e, como o desenvolvedor pode achar que elas permanecem inalteradas, as variáveis poderão acabar sendo usadas em outro lugar da aplicação.
- Várias referências podem ser criadas para o mesmo objeto. Como o Singleton cria apenas um objeto, várias referências podem ser criadas nesse ponto para o mesmo objeto.
- Todas as classes que são dependentes de variáveis globais acabam se tornando altamente acopladas, pois uma mudança feita por uma classe no dado global.

Factory

Factory

- A primeira vantagem é o baixo acoplamento – A criação de um objeto pode ser independente da implementação da classe
- O cliente não precisa conhecer a classe que cria o objeto, que, por sua vez, é utilizado pelo cliente. É necessário conhecer apenas a interface, os métodos e os parâmetros que devem ser passados para criar os objetos do tipo desejado.
- A fábrica pode reutilizar objetos existentes.

Simple Factory

- Para alguns, o SimpleFactory não é, por si só, um padrão. É mais um conceito que os desenvolvedores devem conhecer antes de saberem mais sobre o Factory Method e o Abstract Factory.
- O Factory ajuda a criar objetos de tipos diferentes em vez de os objetos serem diretamente instanciados.

código: animal.py

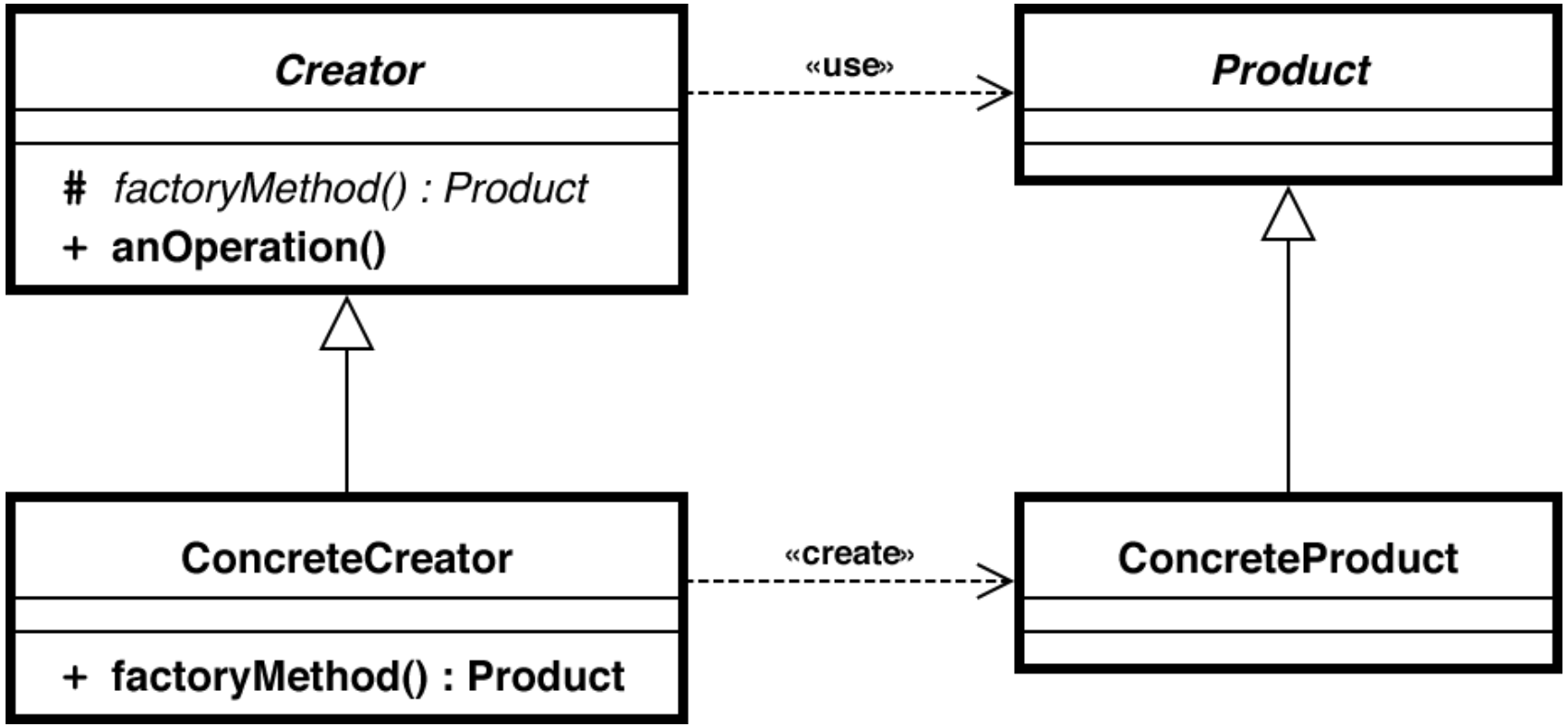
Factory Method

Fatec Araras

- Definimos uma interface para criar objetos, mas, em vez de a fábrica ser responsável pela criação dos objetos, a responsabilidade é passada para a subclasse, que decidirá a classe a ser instanciadas.
- A criação do Factory Method não é feita por instanciação, mas por herança
- O Factory Method deixa o design mais personalizável. Ele pode devolver a mesma instância ou subclasse no lugar de um objeto de determinado tipo.

código: factory_method.py

Factory Method



Fonte: https://pt.wikipedia.org/wiki/Factory_Method#/media/Ficheiro:Factory_Method_UML_class_diagram.png

Abstract Factory

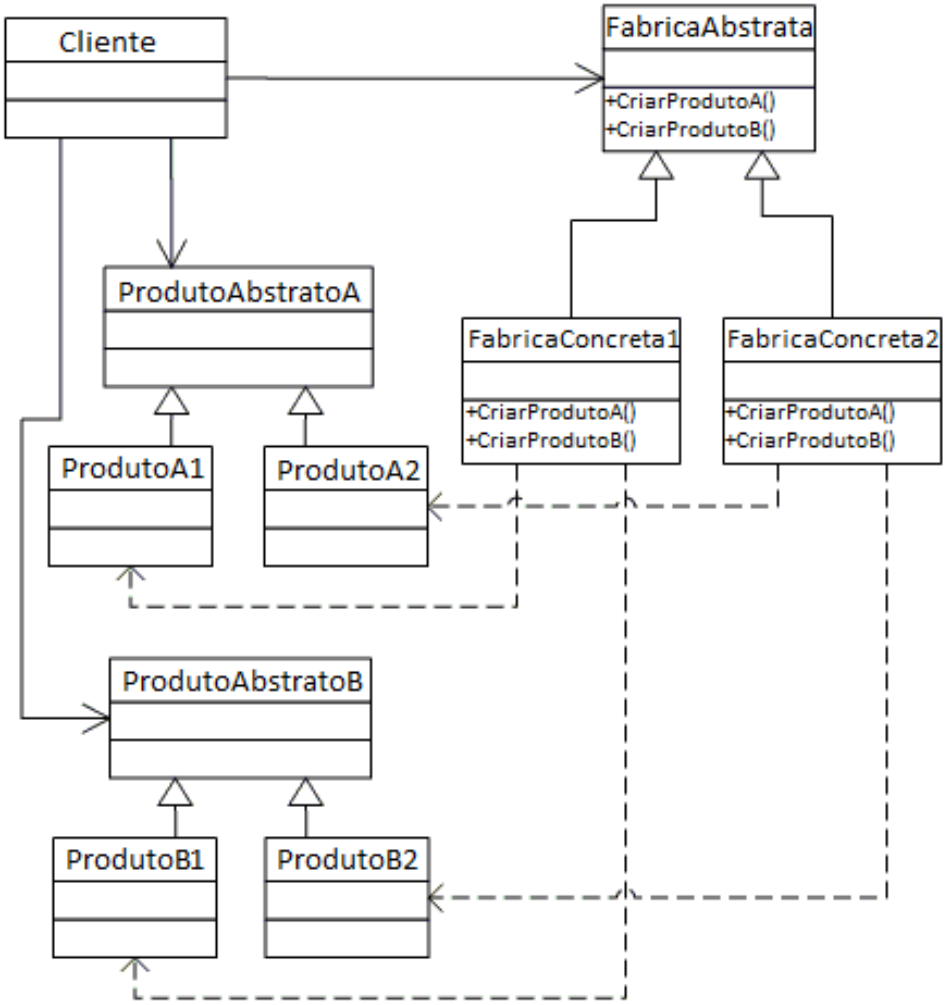
Fatec Araras

O objetivo principal do padrão Abstract Factory é fornecer uma interface para criar famílias de objetos relacionados sem especificar a classe concreta. Enquanto o Factory Method adia a criação da instância para as subclasses, o objetivo do método Abstract Factory é criar famílias de objetos relacionados.

código: abstract_factory.py

Abstract Factory

Fatec Araras



Fonte: https://upload.wikimedia.org/wikipedia/commons/0/0f/F%C3%A1brica_Abstrata.png

Factory Method e Abstract Factory

Fatec Araras

Factory Method	Abstract Factory
Expõe um método ao cliente para criar objetos.	O método Abstract Factory contém um ou mais métodos de fábrica para criar uma família de objetos relacionados.
Usa herança e subclasses para definir o objeto a ser criado.	Usa composição para delegar a responsabilidade de criar objetos de outra classe.
O Factory Method é usado para criar um produto	O método Abstract Factory diz respeito a criar famílias de produtos relacionados.

Virtual Env

VirtualEnv

Fatec Araras

Uma ferramenta para criar um ambiente Python isolado.

Acessar: https://virtualenv.pypa.io/en/latest/

Fatec
Araras

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br