

Exercício Ponto Prova

Exercícios – Padrões de Criação

1. Abstract Factory

1.1.Exercício:

Crie um “Hello, World” que utilize o padrão Abstract Factory para escolher dentre duas formas de impressão: (a) na tela ou (b) num arquivo chamado output.txt. Seu programa deve escolher dentre as duas fábricas aleatoriamente.

1.2.Exercício:

Considere os seguintes conceitos do mundo real: pizzeria, pizzaiolo, pizza, consumidor. Considere ainda que em uma determinada pizzeria, dois pizzaiolos se alternam. Um deles trabalha segundas, quartas e sextas e só sabe fazer pizza de calabresa (queijo + calabresa + tomate), o outro trabalha terças, quintas e sábados e só sabe fazer pizza de presunto (queijo + presunto + tomate). A pizzeria fecha aos domingos.

Tente mapear os conceitos acima para o padrão Abstract Factory (hierarquia de fábricas, hierarquia de produtos, cliente) e implemente um programa que receba uma data como parâmetro (formato dd/mm/yyyy) e imprima os ingredientes da pizza que é feita no dia ou, se a pizzeria estiver fechada, informe isso na tela.

Agora imagine que a pizzeria agora faz também calzones (novamente, de calabresa ou presunto). Complemente a solução com mais este componente.

2. Factory Method

2.1.Exercício:

Construa um programa que receba como parâmetro um ou mais nomes, cada um podendo estar em um dos seguintes formatos:

- "nome sobrenome";
- "sobrenome, nome".

Escreva duas aplicações de construção de nomes, uma para cada formato. Cada uma deve ser responsável por armazenar os nomes criados e imprimi-los quando requisitado. Implemente o padrão Factory Method de forma que somente a criação do nome seja delegada às aplicações concretas (subclasses). Seu programa deve criar as duas aplicações e construir objetos da classe Nome, que deve ter propriedades nome e sobrenome para armazenar as informações em separado. Os nomes não precisam ser impressos em ordem.

Ex.:

```
Entrada: "McNealy, Scott" "James Gosling" "Naughton, Patrick"
James Gosling
Scott McNealy
Patrick Naughton
```

2.2.Exercício:

Crie dois arquivos texto em um diretório qualquer:

publico.txt

"Estas são informações públicas sobre qualquer coisa. Todo mundo pode ver este arquivo."

confidencial.txt

"Estas são informações confidenciais, o que significa que você provavelmente sabe a palavra secreta!"

Usando o padrão Factory Method, crie duas provedoras de informação: uma que retorna informações públicas e outra que retorna informações confidenciais. Utilize o provedor confidencial se o usuário informar a senha "*designpatterns*" como parâmetro para o programa, que deve recuperar a informação e exibi-la na tela.

2.3.Exercício:

Escreva um programa que conte até 10 e envie os números para uma ferramenta de *log*. Esta ferramenta de *log* deve ser construída por uma fábrica. Utilize *Factory Method* para permitir a escolha entre dois tipos de log: em arquivo (log.txt) ou diretamente no console. A escolha deve ser por um parâmetro passado ao programa ("arquivo" ou "console").