

PRÁTICA LABORATORIAL FACTORY METHOD

Objetivos:

- Implementar um exemplo de Factory Method

EXERCÍCIOS

Parte 1

1. Vamos considerar um programa de processamento de pedidos de uma loja online. Imagine que temos diferentes tipos de produtos que podem ser vendidos na loja, como livros, equipamentos eletrônicos e roupas. Cada tipo de produto possui a sua própria classe correspondente. Agora, vamos criar um Factory Method para criar objetos de produtos com base no tipo especificado.

Vamos começar por definir a classe abstrata Product, que será a classe base para todos os tipos de produtos:

```
public abstract class Product {  
    public abstract void display();  
}
```

Agora, vamos criar as classes concretas para diferentes tipos de produtos, estendendo a classe Product:

```
public class Book extends Product {  
    @Override  
    public void display() {  
        System.out.println("Este é um livro.");  
    }  
}  
  
public class Electronics extends Product {  
    @Override  
    public void display() {  
        System.out.println("Este é um produto eletrônico.");  
    }  
}  
  
public class Clothing extends Product {  
    @Override  
    public void display() {  
        System.out.println("Este é uma peça de roupa.");  
    }  
}
```

A seguir, vamos criar a classe ProductFactory que implementará o Factory Method para criar objetos de produtos com base no tipo fornecido:

```
public class ProductFactory {  
    public static Product createProduct(String type) {  
        switch (type) {  
            case "book":  
                return new Book();  
            case "electronics":  
                return new Electronics();  
            case "clothing":  
                return new Clothing();  
            default:  
                throw new IllegalArgumentException("Tipo de produto inválido: " + type);  
        }  
    }  
}
```

Agora, podemos usar o Factory Method para criar objetos de produtos no nosso programa principal:

```
public class Main {  
    public static void main(String[] args) {  
        Product book = ProductFactory.createProduct("book");  
        Product electronics = ProductFactory.createProduct("electronics");  
        Product clothing = ProductFactory.createProduct("clothing");  
  
        book.display();  
        electronics.display();  
        clothing.display();  
    }  
}
```

Neste exemplo, o Factory Method createProduct da classe ProductFactory recebe um tipo de produto como argumento e retorna uma instância apropriada do objeto de produto correspondente. Em seguida, podemos invocar o método display em cada objeto de produto para exibir uma mensagem correspondente ao tipo de produto.

2. Vamos considerar um sistema de gestão de pedidos de um restaurante Mexicano. Imagine que temos diferentes tipos de tacos que podem ser pedidos, como taco de carne vaca, taco de carne de frango e taco vegetariano. Cada tipo de taco possui uma própria classe correspondente. Crie um Factory Method para criar objetos de tacos com base no tipo especificado.

Deve ter uma classe abstrata Taco com os métodos abstratos prepare(), bake(), e box():

- Prepare(): Imprime na consola “A preparar taco [...]”.
- Bake(): Imprime na consola “A cozinhar taco [...]”.
- Box(): Imprime na consola “A embalar taco [...]”.~

Cada implementação deve ser específica da subclasse.

No main deve ser possível instanciar, de acordo com uma escolha do utilizador um taco de determinado tipo e invocar os seus métodos (prepare, bake e box).

3. Vamos considerar um sistema de criação de documentos num processador de texto. Imagine que temos diferentes tipos de documentos que podem ser criados, tendo todos eles um autor, um nome e extensão, como documentos de texto simples, folhas de cálculo e apresentações. Cada tipo de documento possui a própria classe correspondente. Crie um Factory Method para criar objetos de documentos com base no tipo especificado pelo utilizador.

Deve ter uma classe abstrata Document que terá autor, nome e extensão (deve ser automaticamente atribuída de acordo com o tipo de ficheiro docx. para texto, .xlsx para folha de calculo e .pptx para apresentação) como atributos, assim como três métodos: open(), save(),close:

- open(): Imprime na consola “A abrir o documento de [tipo de documento] [nome].[extensão]”. Por exemplo “A abrir documento de texto fichaPratica.docx ”
- save(): Imprime na consola “Alterações do [autor] gravadas”. Por exemplo “Alterações do Vitor gravadas”.
- close(): Imprime na consola “A fechar o documento de [tipo de documento]”.

No main, o utilizador deve dizer qual o tipo de documento, nome e autor e, de seguida, escolher qual dos métodos quer invocar (open(), save(), close()).

Bom trabalho! 😊