

Nome: João Kennedy Souza Soares

Curso: Ciência da Computação

Disciplina: POO

Seminário

Factory method

1. Introdução

No cenário dinâmico do desenvolvimento de software, a aplicação de padrões de projeto desempenha um papel crucial na busca por soluções elegantes e reutilizáveis para desafios comuns. Este relatório explora a implementação do padrão Factory Method em Java, focando na criação de computadores desktop e laptops.

2. Categorias e Objetivos dos Padrões de Projeto

2.1 Contextualização e Motivação

A importância dos padrões de projeto reside na capacidade de fornecer estruturas organizadas para o desenvolvimento de software. Eles não apenas simplificam a complexidade do código, mas também promovem a manutenibilidade e a escalabilidade.

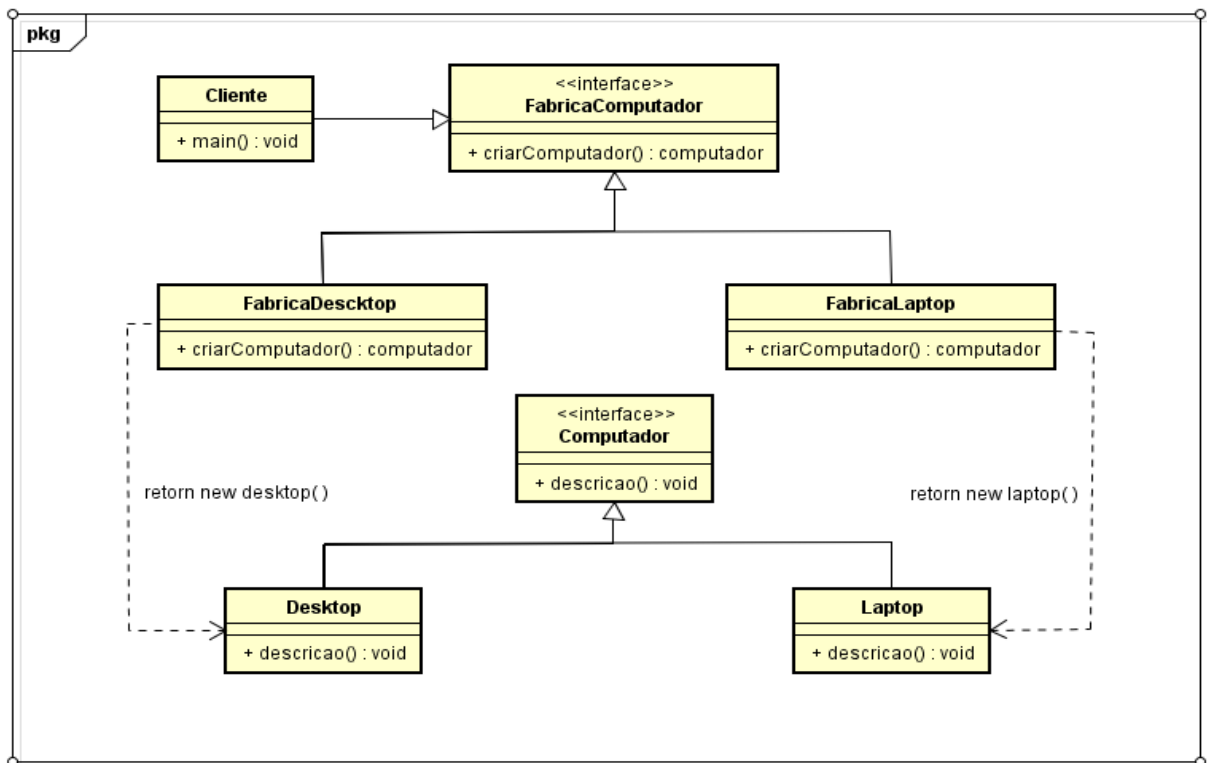
2.2 Categorias de Padrões de Projeto

O padrão Factory Method pertence à categoria Criacional, que lida com a criação de objetos. Ele fornece uma interface para criar objetos em uma superclasse, permitindo que subclasses alterem o tipo de objetos a serem criados.

3. Objetivo, Problema Resolvido, Conceitos, Funcionamento e Aplicação

O Factory Method é um padrão de projeto que visa fornecer uma interface em uma superclasse para a criação de objetos, permitindo que subclasses determinem o tipo específico de objetos a serem criados. Isso promove flexibilidade na instanciação de objetos, evitando acoplamento direto entre cliente e classes concretas. Os conceitos fundamentais incluem a interface para objetos criados (Produto), a declaração do Factory Method na superclasse, e a implementação desse método nas fábricas concretas. O cliente utiliza a interface do produto sem se preocupar com a classe concreta, facilitando a extensão do sistema. O Factory Method é especialmente útil em cenários de criação complexa de objetos ou sujeitos a variações frequentes, proporcionando uma estrutura flexível para sistemas.

4. Diagrama de Classe UML



Fonte: astah community

5. Códigos-fonte

Link github: https://github.com/JoaoKSS/Poo_FactoryMethod.git

6. Conclusão

Em conclusão, a implementação do padrão Factory Method neste contexto específico oferece uma visão clara de suas vantagens na criação de objetos. A flexibilidade proporcionada pela delegação de responsabilidade para subclasses destaca-se como uma abordagem eficaz para lidar com a incerteza na criação de objetos.

Referências

DigitalOcean Community. Factory Design Pattern in Java. DigitalOcean Community, 2022-08-10. Disponível em: <https://www.digitalocean.com/community/tutorials/factory-design-pattern-in-java>. Acesso em: 21 nov. 2023.

Dragon Programmer. Design Patterns Java Factory Method Pattern. Dragon Programmer, 2023-07-20. Disponível em: <https://dragonprogrammer.com/design-patterns-java-factory-method-pattern/>. Acesso em: 21 nov. 2023.

Spring Framework Guru. Factory Method Design Pattern. Spring Framework Guru, 2023-05-03. Disponível em: <https://springframework.guru/gang-of-four-design-patterns/factory-method-design-pattern/>. Acesso em: 21 nov. 2023.