

Processing

Objetivo

Criar um ponto único de acesso para as chamadas dos serviços que o aplicativo irá utilizar.

Motivação

Na programação, em geral, é importante manter o código legível e modulado para facilitar a organização das funções que o aplicativo utilizará. Com a privação dos recursos que os dispositivos vestíveis têm em relação aos computadores convencionais, a organização do desenvolvimento se torna fundamental para ter o controle dos recursos do aplicativo.

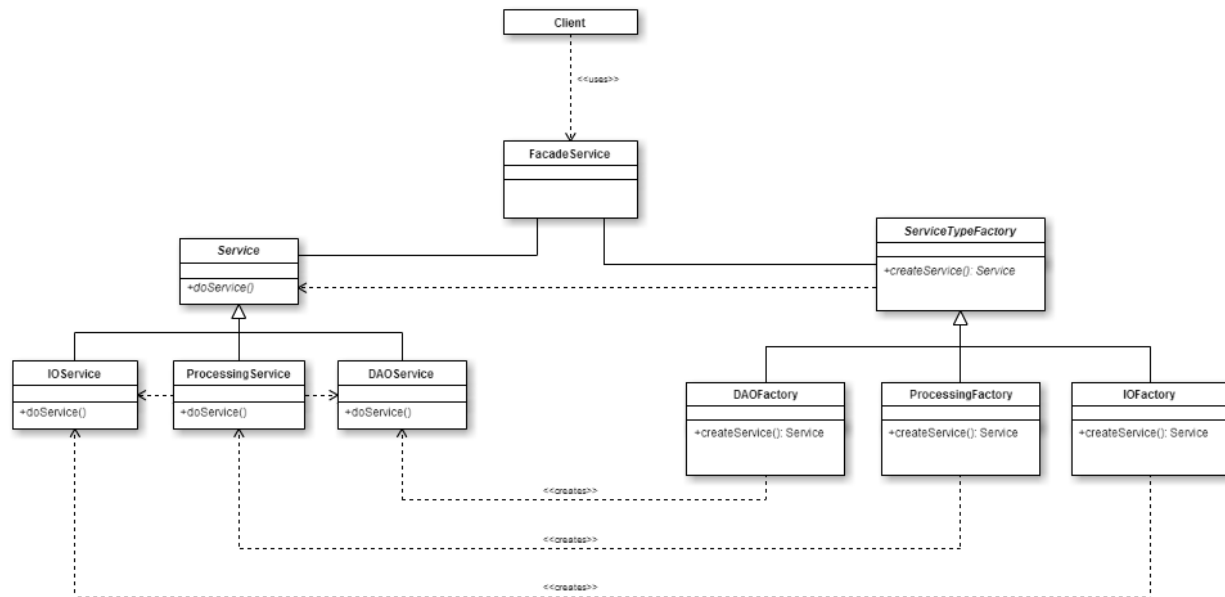
Desta forma, este padrão recomenda o uso dos padrões do catálogo GoF Façade, para o uso de todos os serviços através de um único ponto, e Abstract Factory, para a criação dos serviços de forma escalável.

Aplicabilidade

Usar este padrão para:

- Organizar o código fonte;
- Escalabilidade das classes de serviços;
- Controle das funções de serviço através de uma única classe;
- Modularização das tarefas do aplicativo.

Estrutura



Participantes

- Client: Classe que pede interação com o aplicativo.
- FacadeClass: Controla os acessos às classes do aplicativo.
- ServiceTypeFactory: Classe responsável pela criação das fábricas de serviços.
- DAOFactory/IOFactory/ProcessingFactory: Classes responsáveis por criar os serviços.
- Service: Classe responsável pela implementação dos serviços.
- DAOService/IOService/ProcessingService: Classes de serviço responsáveis pelo controle dos serviços.

Colaborações

Mantém-se maior controle do desenvolvimento e melhor organização.

Consequência

- Melhor controle dos processos do aplicativo, essencial para dispositivos com recursos limitados.
- Refatorável, ajudando a refatorar o código para utilização em outros aplicativos.

Implementação

Exemplo de Código

Usos conhecidos

Patterns Relacionados

Façade, Abstract Factory, Storage, Communication.