

Padrões de Projetos

Allyson Oliveira de Abreu
Patrick Anderson de Souza



Atividade Colaborativa

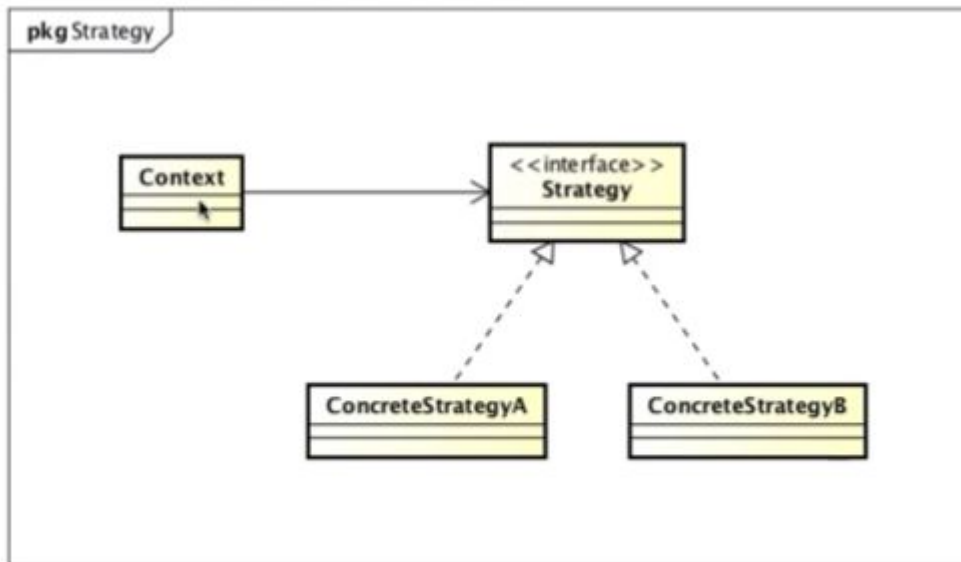
Padrão escolhido: Pattern Strategy





Conceito

- O Pattern Strategy define uma família de algoritmos, encapsula cada um deles e os torna intercambiáveis. O strategy deixa o algoritmo varia independente dos clientes que o utilizam.



Atividade Individual

Padrão escolhido: Composite
Patrick Anderson de Souza





Conceito

- Padrão de projeto estrutural;
- Compor classes utilizando outras pré-existentes;
- Criar estruturas semelhantes a uma árvore.



Objetivo

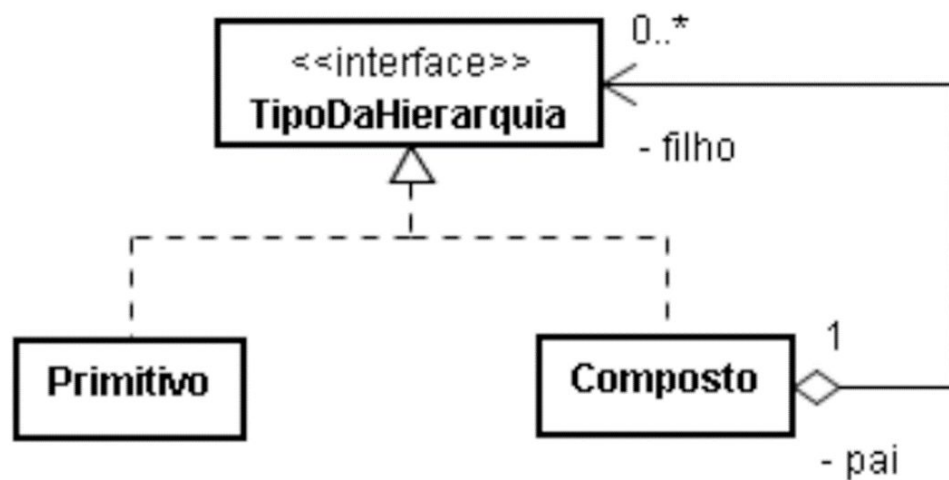
- Construir objetos com base em outros objetos semelhantes, para formar uma hierarquia;
- Referenciar facilmente objetos que fazem parte da mesma árvore.



Objetivo

- Garantir tratamento igualitário a qualquer objeto, seja ele individual ou composto;
- Permite construir elementos mais complexos a partir de elementos simples;
- Não há preocupação dentro do programa se o objeto é primitivo ou não.

Exemplo



Atividade Individual

Padrão escolhido: Singleton
Allyson Oliveira de Abreu





Conceito

- Neste padrão, garantimos que apenas um objeto será instanciado;
- Geramos apenas um objeto que estará disponível de forma única para todo o escopo da aplicação.



Objetivo

- Utilizar quando se quer apenas uma única instância de uma classe para toda a aplicação;
- Ter uma maneira de fácil acesso a esta instância;



Formas de instanciação

- Quando a classe é carregada;
- Tardia (lazy instantiation), com o primeiro uso da classe.



Exemplo

```
abstract public class Singleton
{
    protected Singleton() {}
    abstract protected Singleton makeInstance();
    private static Singleton soleInstance = null;
    public static Singleton Instance() {
        if (soleInstance == null)
            soleInstance = makeInstance();
        return soleInstance;
    };
};
```