

Programação II

Prof. Alcemir Rodrigues Santos

Introdução aos Padrões de Projeto



Agenda

- Introdução
- Definição
- Exemplos e Implementações
 - Singleton
 - Observer
 - Command
 - State



11/19/19

Programação II | Alcemir Santos

2



Padrões de Projeto

Introdução

19-Nov-19

Engenharia de Software | Alcemir Santos

3



Padrões de Projeto

- Um padrão de projeto é uma solução reutilizável geral para um problema comum
- Eles representam as melhores práticas
 - Padrões permitem reuso de conhecimento de experts
- Eles não descrevem uma solução completa, vez que eles tem o propósito de servir diversas aplicações



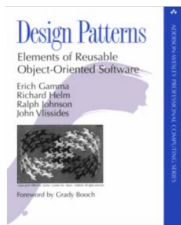
19-Nov-19

Engenharia de Software | Alcemir Santos

4



Os 23 Padrões do GoF



- Os mais conhecidos padrões de projetos estão neste livro
 - Erich Gamma
 - Richard Helm
 - Ralph Johnson
 - John Vlissides
- Estes autores ficaram conhecidos como "Gang of Four" (GoF)
- O livro trouxe 23 padrões de projeto

19-Nov-19

Engenharia de Software | Alcemir Santos

5



Classificações dos Padrões de Projeto do GoF

- **Propósito**
 - **Criacional:** o processo de criação de objetos
 - **Estrutural:** a composição das classes
 - **Comportamental:** as formas como classes e objetos interagem e distribuem responsabilidades
- **Escopo**
 - **Classes:** relacionamentos entre classes e suas subclasses
 - **Objetos:** relacionamentos entre objetos, os qual pode ser mudado em tempo de execução (dinâmico)

19-Nov-19

Engenharia de Software | Alcemir Santos

6



Padrões de Projeto do GoF

- Criacionais
 - Abstract Factory
 - Builder
 - Factory Method
 - Prototype
 - Singleton
- Estruturais
 - Adapter
 - Bridge
 - Composite
 - Decorator
 - Facade
 - Flyweight
- Proxy
- Comportamentais
 - Chain of Responsibility (CoR)
 - Command
 - Interpreter
 - Iterator
 - Mediator
 - Memento
 - Observer
 - State
 - Strategy
 - Template Method
 - Visitor

19-Nov-19

Engenharia de Software | Alomar Santos

7

Singleton

Padrões Criacionais

19-Nov-19

Engenharia de Software | Alomar Santos

8

Contexto

- Algumas classes tem exatamente uma instância
- Singleton faz a classe ser responsável por manter registro de sua única instância

19-Nov-19

Engenharia de Software | Alomar Santos

9

Descrição do Padrão

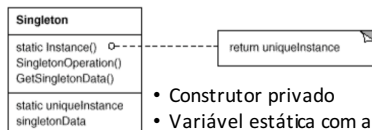
- Nome
 - Singleton
- Descrição do Problema
 - Garantir que a classe tem somente uma instância
- Solução (à seguir)
- Consequências
 - Acesso controlado à única instância
 - Ponto de acesso à instância compartilhado

19-Nov-19

Engenharia de Software | Alomar Santos

10

Solução (Singleton)



- Construtor privado
- Variável estática com a única instância
- O método **instance()** é estático e público
 - Retorna a a instância única

19-Nov-19

Engenharia de Software | Alomar Santos

11

Exemplo

```

public class ClassSingleton {
    private static ClassSingleton instance;

    private ClassSingleton(){}

    public static ClassSingleton getInstance() {
        if (instance == null)
            instance = new ClassSingleton();
        return instance;
    }
}
  
```

Um atributo estático é usado para armazenar a instância.

19-Nov-19

Engenharia de Software | Alomar Santos

12

Exemplo

```
public class ClassSingleton {
    private static ClassSingleton instance;

    private ClassSingleton(){}

    public static ClassSingleton getInstance() {
        if (instance == null)
            instance = new ClassSingleton();
        return instance;
    }
}
```

Construtor privado para evitar instâncias inesperadas.

19-Nov-19

Engenharia de Software | Alimir Santos

13

Exemplo

```
public class ClassSingleton {
    private static ClassSingleton instance;

    private ClassSingleton(){}

    public static ClassSingleton getInstance() {
        if (instance == null)
            instance = new ClassSingleton();
        return instance;
    }
}
```

Forma única de acesso à instância da classe..

19-Nov-19

Engenharia de Software | Alimir Santos

14

Observer

Padrões comportamentais

19-Nov-19

Engenharia de Software | Alimir Santos

15

Definição

- Habilita uma funcionalidade publicação/assinante
- Esta funcionalidade é possível de ser alcançada por um objeto autônomo, publicante que permite que outros objetos assinem/cancem o recebimento de suas atualizações
- O padrão não impõe quaisquer limite ao número de assinantes
 - Todos receberam a notificação de atualizações futuras do publicante
- Onde usar?
 - Onde um objeto quer publicar informação e muitos objetos querem receber essa informação

Programação 2 | Alimir Santos

16

Vantagens? Desvantagens?

Vantagens

- Minimiza o acoplamento entre publicante e assinante, vez que o publicante não precisa saber quem ou como muitos assinantes haverão

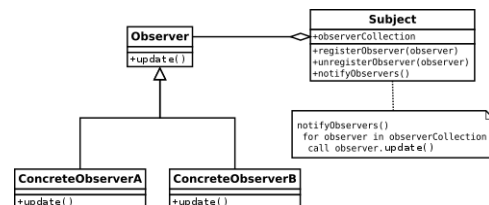
Desvantagens

- Cenários complexos podem haver problemas para determinar se a publicação é relevante para todos os assinantes ou só parte deles
- Enviar um sinal de atualização para todos os assinantes pode gerar sobrecarga de comunicação de informação não necessária

Programação 2 | Alimir Santos

17

Solução (Observer)



11/19/19

Programação 2 | Alimir Santos

18

Command

Padrões comportamentais

19-Nov-19

Engenharia de Software | Alcemir Santos

19

Definição

- O Command é usado para criar objetos que representam ações e eventos em uma aplicação
- Ele encapsula uma ação ou evento e contém toda informação necessária para entender exatamente o que aconteceu
- Ao passar um objeto Command como parâmetro, pode-se a qualquer momento extrair informação sobre as ações e eventos que aconteceram

Onde usar?

- Onde você precisa que uma ação possa ser representada de diversas formas, como em menus drop-down, botões e menus pop-up
- Para criar uma funcionalidade de "desfazer" (CTRL + Z)

Programação 2 | Alcemir Santos

20

Vantagens? Desvantagens?

Vantagens

- Um objeto Command pode armazenar parâmetros de procedimentos. Pode ser usado para recuperar parâmetros de chamadas de função e deixá-los de lado para usar depois
- Permite a reação a um comando que deve ser executado algum tempo depois de ter acontecido
- Habilitam estruturas contendo múltiplos comandos
- Objetos Command permitem desfazer as ações, vez que eles são armazenados

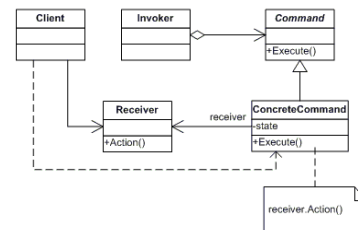
Desvantagens

- Proliferação de pequenas classes que podem poluir o código

Programação 2 | Alcemir Santos

21

Solução (Command)



11/19/19

Programação 2 | Alcemir Santos

22

State

Padrões comportamentais

19-Nov-19

Engenharia de Software | Alcemir Santos

23

Motivação

- Considere um sistema de gerenciamento de tarefas
 - Uma tarefa pode ter diferentes estados (e.g., "em progresso", "concluído", ou "cancelado")
- Uma tarefa tem comportamento diferente dependendo de seu estado corrente
 - Ex. Se uma tarefa está cancelada, ela não pode ser alocada à ninguém

19-Nov-19

Engenharia de Software | Alcemir Santos

24

Descrição do Padrão

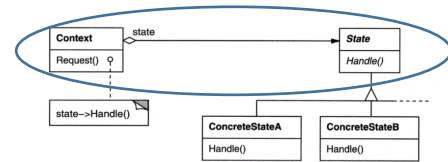
- Nome
 - State
- Descrição do Problema
 - Permitir a um objeto a mudar o seu comportamento quando o seu estado interno muda
- Solução (à seguir)
- Consequências
 - Torna a transição de estados explícita
 - Os objetos Estados podem ser compartilhados

19-Nov-19

Engenharia de Software | Alomar Santos

25

Solução (State)



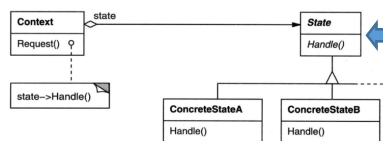
A classe contexto tem 1 estado ativo e tem acesso a ele via associação.

19-Nov-19

Engenharia de Software | Alomar Santos

26

A classe Estado



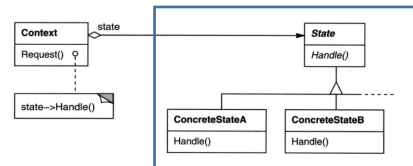
A classe abstrata comum define métodos comuns a todos os estados.

19-Nov-19

Engenharia de Software | Alomar Santos

27

Estados concretos



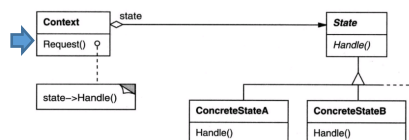
Os Estados concretos implementam os estados possíveis e sobrescrevem os métodos de comportamento específico de cada objeto.

19-Nov-19

Engenharia de Software | Alomar Santos

28

A classe Estado



O comportamento de um contexto depende dos métodos do estado corrente.

19-Nov-19

Engenharia de Software | Alomar Santos

29