Aulas 6

Interface –Padrões de Projetos – Strategy

RESUMO

Interface

-Um padrão é definido através de especificações ou contratos.
-Nas aplicações orientadas a objetos, podemos criar um "contrato" para definir um determinado conjunto de métodos que deve ser implementado pelas classes que "assinarem" este contrato. Em orientação a objetos, um contrato é chamado de interface.
-Uma interface é composta basicamente por métodos abstratos.
-Todos os métodos de uma interface devem ser públicos e abstratos.
As classes concretas que implementam uma interface são obrigadas a possuir uma implementação para cada método declarado na interface.

Exemplo:

```
1  // Esta classe não compila porque ela não implementou o método saca()
2  class ContaCorrente implements Conta {
3    public void deposita(double valor) {
4      // implementacao
5    }
6  }
```

Código Java 11.4: ContaCorrente.java

As vantagens são:
-Utilizar uma interface é a padronização das assinaturas dos métodos oferecidos por um determinado conjunto de classes.
-Garante que determinadas classes implementem certos métodos.

***Anotações*:** *contrato = interface.*

Polimorfismo com interface

-Se uma classe implementa uma interface, podemos aplicar a ideia do polimorfismo assim como quando aplicamos herança.
-Dessa forma, outra vantagem da utilização de interfaces é o ganho do polimorfismo.
Exemplo:
-Suponha que a classe ContaCorrente implemente a interface Conta.
-Podemos guardar a referência de um objeto do tipo ContaCorrente em uma variável do tipo Conta.

```
1  Conta c = new ContaCorrente();
```

Código Java 11.5: Polimorfismo

Interface e Herança

-A grosso modo, priorizar a utilização de interfaces permite que alterações pontuais em determinados trechos do código fonte sejam feitas mais facilmente pois diminui as ocorrências de efeitos colaterais indesejados no resto da aplicação.

-Por outro lado, priorizar a utilização de herança pode diminuir a quantidade de código escrito no início do desenvolvimento de um projeto.

***Anotações**: Interface faz a manutenção do código, herança diminui o código e utiliza sem duplicar o código*

Padrões de Projetos

Padrões GoF são classificados em três categorias:
-Padrões de Criação
-Padrões Estruturais
-Padrões Comportamentais.

Padrões Comportamentais

Estes padrões facilitam a comunicação entre os objetos, distribuindo as responsabilidades e definindo a comunicação interna.
Veja abaixo um resumo do objetivo de cada padrão comportamental.
-Command Controla chamadas a um determinado componente, modelando as requisições como objeto. Permite que operações sejam desfeitas, enfileiradas ou registradas.
-Iterator Fornecer um modo eficiente para percorrer sequencialmente elementos de uma coleção, sem expor a estrutura interna da coleção.
-Mediator Diminuir a quantidade de "ligações" entre objetos introduzindo mediador, através do qual toda comunicação será realizada.
-Observer Definir um mecanismo eficiente para reagir às alterações realizadas em determinados objetos.
-State Alterar o comportamento de um determinado objeto de acordo com o estado no qual ele se encontra.
-Strategy Permitir de maneira simples a variação dos algoritmos utilizados na resolução de um determinado problema
-TemplateMethod Definir a ordem na qual determinados passos devem ser realizados na resolução de um problema e permitir que esses passos possam ser realizados de formas diferentes de acordo com a situação.
-Visitor Permitir atualizações específicas em uma coleção de objetos de acordo como tipo particular de cada objeto atualizado

Strategy

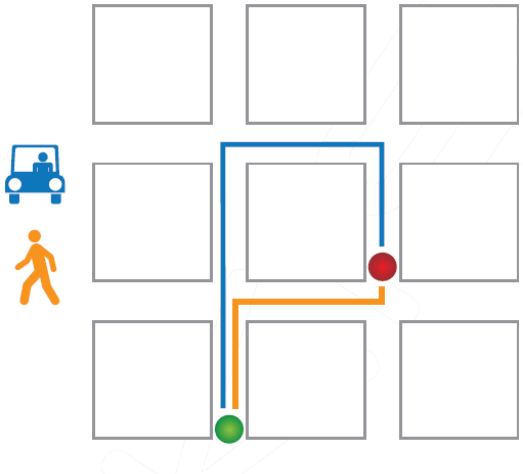

Figura 4.10: Diferentes modos de se chegar a um determinado destino

-O padrão Strategy propõe uma solução que pode ser adotada nesse cenário.
-A ideia fundamental desse padrão é possibilitar facilmente a variação do algoritmo a ser utilizado na resolução de um problema
-Em nosso exemplo, diferentes algoritmos são usados para se encontrar uma rota entre dois pontos, dependendo do modo como o usuário deseja percorrer o caminho.

Anotações: *o resultado sempre dever ser o mesmo, não importa o tipo de algoritmo foi aplicada.*