

ITE-004 - TÓPICOS ESPECIAIS

PROF. ALEXANDRE GARCIA

1. PROVA 3

Exercício 1. *(Operações)* Deseja-se implementar uma pequena calculadora (que efetua soma apenas) usando o pattern visitor. A *ConcreteElement DoisArgs* deve carregar dois ints, um construtor e o método *accept*, já a *TresArgs* deve carregar tres ints, construtor e o método *accept* (ambas devem ser subtipos de *NumeroElement*). O *Concrete Visitor SomaVisitor* deve somar adequadamente os números de acordo com os argumentos e mostrar na tela o valor desta soma (esta deve ser subtipo de *OperVisitor*). A classe *CalcStructure* deve percorrer a estrutura. Implemente todas as classes.

Exercício 2. A interface *Cloneable* provê o método *clone()* usado no padrão *Prototype*. Uma classe *Foo* possui um atributo *x* inteiro e um atributo *bar* do tipo *Bar*. Sabe-se que *Foo* é um *Cloneable*. A classe *Bar* possui o atributo *y* do tipo *String*. Implemente o método *clone()* da classe *Foo* de modo a se obter um deep clone de *this*.

Exercício 3. Em um sistema jurídico, um processo civil é composto por uma descrição, uma número de processo e uma instância (inteiro onde indica qual instância o processo está. Instância aqui não é sobre POO). Sabe-se que um processo civil pode ser julgado em quatro instâncias no Brasil. Primeiramente é julgado na vara cível, se houver recurso (o atributo instância que controla), o processo poderá ser julgado no tribunal de justiça, em recursos específicos poderá ser julgado em terceira instância ou Superior tribunal de justiça ou em última instância no Supremo Tribunal Federal (STF). Porém há casos que exigem rapidez no julgamento e os processos podem ir para a arbitragem (que não é um tribunal!) a qualquer momento no fluxo das 4 instâncias. Dado a situação descrita

- (1) Escolha um ou mais patterns convenientes para a solução do problema. Justifique sua(s) escolha(s);

- (2) *Justifique o motivo de todos os outros patterns não ter(em) sido(s) escolhido(s);*
- (3) *Implemente todas as classes com clareza;*

Exercício 4. *Deseja-se construir um aplicativo para conversão de unidades monetárias. O sistema deve processar uma String contendo o valor e a moeda a ser convertida (por exemplo, "200 reais para euros") e retornar um valor double representando a o valor convertido. O conversor para cada corréncia (Euro, Dólar, Libra e Peso), deve variar no mesmo objeto de conversão.*

Exercício 5. *Em um sistema de produtos bancários sabe-se que um Investimento possui nome e valor aplicado. Os investimentos podem ser Fundos, Poupança e Renda Fixa. O rendimento é um valor que será acrescido ao valor aplicado, este pode ser aleatório (`Math.random()`), 0.5% ou 1.1% para Fundos, Poupança e Renda Fixa respectivamente. São passíveis de imposto apenas Fundos e Renda Fixa com taxas de 10% e 3% respectivamente. Implemente este problema utilizando um padrão conveniente. Use enums em alguma parte da implementação.*

Exercício 6. *Usando o sistema acima os produtos Seguro de Vida e Seguro Casa, serão oferecidos pelo banco como um adicional. Ambos cobram taxas adicionais de 1% e 1.3% sobre o valor aplicado. Esses adicionais, devem compor o produto bancário em tempo de execução.*

Exercício 7. (5.0) *Um novo APP é lançado para Android! O FootStar App é um aplicativo para Android que permite avisar a todos os Fãs de futebol cadastrados de um determinado astro do futebol informações como, o local onde ele está no momento, as últimas fofocas e até potenciais trocas de time em primeira mão! Este app é perfeito para a Imprensa e Fãs de plantão! Há mensagens personalizadas para estes dois grupos distintos.*

- (1) *Qual o padrão adequado para o problema? Justifique.*
- (2) *O código fonte*

Exercício 8. *Sabe-se que bolas de futebol são feitas por diferentes fabricantes de material esportivo. Cada fabricante(Nike, Adidas e Puma) pode dar cores, materiais e nomes diferentes para as bolas de futebol.*

BolaNike, BolaPuma e BolaAdidas são tipos de Bola e cada uma possui um método de informações do fabricante (quem fabricou, cor, nome e material). O fabricante deverá ser obrigatoriamente uma enum. O fabricante deve retornar uma instância correspondente de bola.

Exercício 9. *Sabe-se que Quadrados e Retângulos são Quadriláteros. Quadrados possuem um lado e o Retângulo dois. É possível calcular áreas e perímetros para ambos. Porém, só é possível calcular essas medidas se os lado(s) forem maiores que zero. Dado o problema de validação dos dados, duas novas entidades são necessárias. QuadrilateroProxy e FiguraInexistente. Usando o padrão Proxy codifique todas as classes envolvidas. Na classe QuadrilateroProxy em seu construtor, haverá a validação (se é Quadrado, Retângulo ou Inexistente).*

Exercício 10. *Deseja-se construir uma nova linguagem de programação chamada Z. Para isso sabe-se que a linguagem possui dois tipos de comando, comando fim e um comando de continuação. A linguagem deve ter uma abstração dos comandos pelo método void execute(). O comando fim, chamado End, deve encerrar o programa e a especificação de execute deve ser apenas a mensagem programa encerrado com sucesso. Os comandos de continuação são dois: Bing que toca um som (syso de um som na tela), Output que imprime uma string (essa string deverá ser atributo da classe) e Calculation que representa um cálculo (está deverá ter dois atributos e 4 subclasses representando as quatro operações). Os comandos de continuação podem ser executados dentro de uma sequência, até que fim (End) os encerre. Ou seja, pode haver uma lista de comandos mesclados entre Bing, Output e Calculation até que End encerre esta lista. Usando a situação descrita e o princípio do aberto/fechado. Faça um programa que calcule a área de um quadrado usando essa nova linguagem como teste.*