

Padrões de Projeto GoF

TCE/TO (CESPE 2008)

48 Acerca de padrões de projeto, assinale a opção correta.

A) Padrões de projeto são aplicados no tratamento de códigos específicos de linguagem de programação orientada a objetos. Essa característica permite redução de custos, mas prejudica a qualidade de uma aplicação.

B) Padrões de projeto são tratados com o enfoque na melhoria da qualidade de um projeto de software. O seu uso racional aumenta a produtividade de determinado ponto de desenvolvimento e aumenta os custos em função da qualidade de uma aplicação.

C) Embora não sejam aplicados em técnicas de desenvolvimento de aplicações, padrões de projeto fornecem uma visão orientada a objetos para solução de problemas, com um baixo custo.

D) Padrões de projeto são aplicados ao desenvolvimento de aplicações e, de maneira geral, fornecem meios de se descrever soluções para problema. Essa característica permite redução do tempo gasto com o desenvolvimento e melhoria da qualidade da aplicação.

E) Padrões de projeto permitem redução do tempo gasto com um desenvolvedor e geralmente melhoram a qualidade de uma aplicação. Essa característica incentiva o reuso de software para resolver problemas, mas tem como desvantagem o aumento do custo de uma aplicação.

TRT/RN – Técnico (CESPE 2010)

Acerca dos padrões de projeto, julgue o próximo item.

78 Os padrões de projeto podem ser definidos como soluções já testadas para problemas que ocorrem frequentemente durante o projeto de software.

INMETRO (CESPE 2010)

48 Acerca dos padrões de criação e de padrões estruturais, assinale a opção correta.

A) O padrão adaptador de classe (class adapter) facilita acomodar a adaptação de subclasses de uma classe previamente adaptada.

B) Uma das consequências do uso do padrão Prototype é o aumento no uso de subclasses.

C) O uso do padrão singleton apresenta como consequências uma redução do uso de variáveis globais e uma melhor organização do espaço de nomes de um software orientado a objetos.

D) São exemplos de padrões estruturais: Adapter, Façade, Proxy, Composite, Decorator e Chain of Responsibility.

E) São participantes do padrão Façade: o cliente, o alvo, o adaptador e o adaptado.

SERPRO (CESPE 2010)

87 Padrões de criação, como singleton e prototype, envolvem necessariamente a instanciação de um ou mais objetos, sendo que no padrão singleton, o controle sobre a instanciação é cedido ao cliente do padrão, o que não ocorre no padrão prototype

88 No padrão builder, a responsabilidade pela criação de instâncias é compartilhada por um diretor e um construtor, sendo o vínculo entre eles estabelecido pelo cliente do padrão

89 No emprego do padrão fábrica abstrata, a criação de novos produtos dentro de uma mesma família torna-se dificultada, quando comparada à criação de novas famílias de

produto que aderem a uma mesma estrutura de produtos abstratos

90 Uma das conseqüências do uso adaptador de classes é que a adaptação de uma classe concreta não se estende automaticamente às suas subclasses

91 Uma das conseqüências desejáveis do uso de uma façade é a promoção do acoplamento fraco entre um sistema e seus clientes, sobretudo porque as dependências de compilação entre módulos podem ser reduzidas. Uma das conseqüências indesejáveis é a redução de portabilidade dos sistemas

INMETRO (CESPE 2009)

96 A programação de um padrão Singleton na linguagem java usualmente demanda que o método getInstance seja declarado como de instância.

98 Uma das vantagens do padrão Prototype é o reúso de memória possível por meio do compartilhamento de vários objetos por meio de múltiplas interfaces.

99 Devido à inexistência de múltipla herança de classes na linguagem Java, o padrão Adapter de classes é pouco utilizado por programadores dessa linguagem.

101 O uso do padrão Command apresenta conseqüências como um objetoCommand é usualmente refratário ao enfileiramento; um objeto Command é usualmente transiente, isto é, não é passível de serialização e o uso disseminado de Commands dificulta a estruturação de um sistema em operações de alto nível.

TCU (CESPE 2009)

118 Se, no desenvolvimento de uma aplicação que leia documentos do tipo txt e seja capaz de converter o documento em vários formatos distintos, houver a

necessidade de facilitar acréscimos de novos tipos de conversão, será mais indicado o uso do padrão de estrutura Adapter que o uso do padrão de estrutura Bridge, pois o padrão Adapter separa a construção de um objeto complexo de sua representação para criar representações diferentes com o mesmo processo.

119 No desenvolvimento de um sistema estruturado em subsistemas para facilitar o acesso e minimizar a comunicação e dependências entre os subsistemas, o padrão de criação Factory Method, que fornece uma interface para a criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas, é mais indicado que o padrão de criação Prototype.

TCE/RN (CESPE 2009)

59 O template method se aplica primariamente às classes, sendo um padrão de projeto com finalidade comportamental, ou seja, caracterizado pela maneira como as classes interagem e distribuem responsabilidades.

STJ (CESPE 2008)

70 Os padrões de projeto podem ser usados no projeto orientado a objetos para apoiar o reúso de software. Esses padrões freqüentemente empregam a herança e o polimorfismo para prover generalidade. Abstract factory, strategy e template method são padrões de projeto que podem ser empregados nos frameworks orientados a objetos para facilitar a adaptação dos frameworks.

SERPRO (CESPE 2008)

Com relação aos padrões de projeto, julgue os itens seguintes.

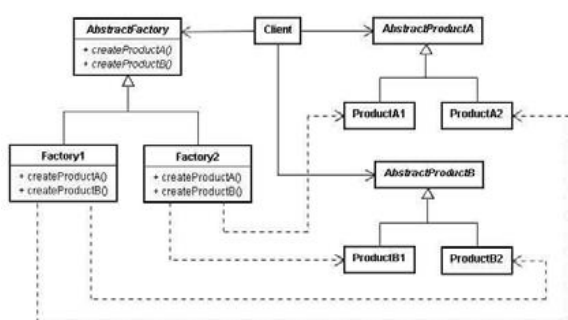
111 O Singleton é um padrão que garante que uma classe tenha apenas uma instância.

112 Alguns padrões de criação como o Prototype, o Proxy e o Façade não são adequados para a programação orientada a objetos.

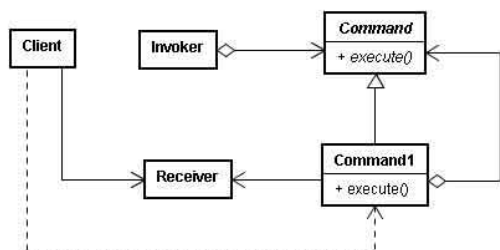
115 Adapter é um padrão estrutural utilizado para compatibilizar interfaces de modo que elas possam interagir.

Min. Comunicações (CESPE 2008)

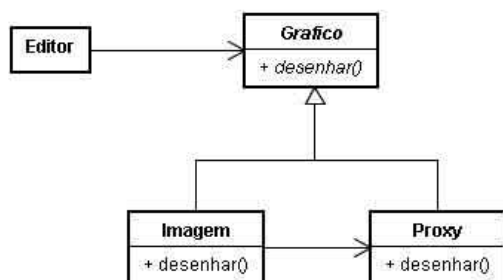
97 O seguinte diagrama UML documenta corretamente o padrão abstract factory.



98 O padrão command está corretamente documentado no seguinte diagrama UML.



99 O padrão proxy está corretamente documentado no seguinte diagrama UML.



BASA (CESPE 2007)

65 O padrão adapter define uma família de algoritmos, permite o encapsulamento de algoritmos e possibilita a substituição desses algoritmos. Os algoritmos podem variar independentemente dos seus clientes. Esse padrão pode ser usado quando várias classes relacionadas diferirem apenas nos seus comportamentos.

TJDFT (CESPE 2008)

85 O padrão de projeto orientado a objetos denominado singleton exprime o fenômeno recorrente na análise que é a existência de muitas aplicações nas quais há um objeto que é a única instância de sua classe.

PETROBRAS (CESPE 2007)

90 O padrão de projeto (design pattern) Proxy é usado quando se quer usar uma classe existente, mas a interface da classe não é a de que se precisa. O Proxy pode ser implementado usando-se herança múltipla para converter uma interface em outra.

91 Ao se estruturar um sistema em subsistemas deve-se minimizar as comunicações e dependências entre subsistemas no projeto empregando-se o padrão de projeto Decorator, no qual os clientes se comunicam com o subsistema enviando mensagens para uma instância de uma classe Decorator, que repassa as mensagens para objetos no subsistema.

92 O padrão Abstract Factory visa possibilitar que os passos para se criar um objeto complexo independam das partes que compõem esse objeto. Ao encapsular a construção de objetos complexos, esse padrão contribui na modularização do sistema.

DATAPREV (CESPE 2006)

Com relação a padrões de projeto (design patterns), julgue os itens que se seguem.

67 As seguintes situações justificam o uso do padrão Abstract Factory: o sistema deve ser independente de como os objetos são criados; o sistema deve poder ser configurado com diferentes famílias de classes; é necessário garantir que certas classes sejam usadas em conjunto.

68 As seguintes situações justificam o uso do padrão Adapter: é necessário um objeto local que se faça passar por um objeto localizado em outro espaço de endereçamento; é necessário controlar o acesso a um objeto; um objeto persistente deve ser carregado em memória somente quando for referenciado.

70 As seguintes situações justificam o uso do padrão Strategy: é necessário configurar uma classe com uma variedade de comportamentos; uma classe usa diferentes variações de um algoritmo; o método de uma classe tem muitos enunciados condicionais, pois a classe tem comportamentos variados.

TSE (CESPE 2006)

49 Quanto aos padrões de projeto orientados a objetos, assinale a opção correta.

A) O Facade pode ser usado quando se deseja prover uma interface simples para um subsistema complexo; existem muitas dependências entre clientes e as classes que implementam uma abstração.

B) Pode-se usar o Decorator quando um sistema deve ser configurado com uma entre várias famílias de produtos; uma família de produtos relacionados foi projetada para ser usada em conjunto.

C) O Adapter pode ser usado quando objetos se comunicam de forma definida, mas complexa; as interdependências entre

os objetos são difíceis de entender; o reuso está sendo dificultado, pois um objeto se comunica com vários outros.

D) Pode-se usar o Builder quando o comportamento de um objeto muda em tempo de execução e depende do seu estado; as operações têm múltiplos enunciados condicionais que dependem do estado do objeto.

Gabaritos:

TCE/TO (CESPE 2008)
48 D

TRT/RN – Técnico (CESPE 2010)
78 C

INMETRO (CESPE 2010)
48 C

SERPRO (CESPE 2010)
87 E, 88 C, 89 C, 90 C, 91 E

INMETRO (CESPE 2009)
96 E, 98 E, 99 E, 101 E

TCU (CESPE 2009)
118 E, 119 E

TCE/RN (CESPE 2009)
59 C

STJ (CESPE 2008)
70 C

SERPRO (CESPE 2008)
111 C, 112 E, 115 C

Min. Comunicações (CESPE 2008)
97 C, 98 E, 99 E

BASA (CESPE 2007)
65 E

TJDFT (CESPE 2008)
85 C

PETROBRAS (CESPE 2007)
90 E, 91 E, 92 E

DATAPREV (CESPE 2006)
67 C, 68 E, 70 C

TSE (CESPE 2006)
49 A