



PADRÕES DE PROJETO DE SOFTWARE

CCT0763_A4_201309069484_V1



Vídeo



PPT



MP3

Aluno: **ANA CRISTINA FERREIRA LEITE LOURENÇO DE SOUSA**Matr.: **201309069484**Disc.: **PAD. PROJ. SOFTWARE****2020.1 - F (G) / EX**

Prezado (a) Aluno(a),

Você fará agora seu **TESTE DE CONHECIMENTO**! Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responde cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

Saiba mais

1. O objetivo do Flyweight é diminuir o consumo de memória, porém deve ser utilizado com cautela. Gamma et al (2000) recomendam sua utilização somente quando todas as condições abaixo forem verdadeiras, EXCETO uma. Qual delas abaixo é essa situação?

- ☐ Uma aplicação utiliza um grande número de objetos.
- ☐ Muitos grupos de objetos podem ser substituídos por relativamente poucos objetos compartilhados, uma vez que estados extrínsecos são removidos.
- ☒ Quando for necessário elaborar um algoritmo ou algo complicado para separar objetos mutáveis de imutáveis;
- ☐ Os custos de armazenamento são altos por causa da grande quantidade de objetos.
- ☐ A aplicação não depende da identidade do objeto.

Explicação:

Essa afirmativa está **incorreta** para o padrão **Flyweight**, porque ela é uma situação em que não deve ser usada, assim como também quando o estado dos objetos não for imutável (é preciso passar o estado mutável como parâmetro e isto pode ser impraticável se o estado consistir de vários objetos)..

Saiba mais

2. Estruturar um sistema em subsistemas ajuda a reduzir a complexidade do problema (Gamma et al., 2000). No entanto, um objetivo comum de todos os projetos é minimizar a comunicação e as dependências entre os diferentes subsistemas. Uma das maneiras de contornar esse problema é utilizar o padrão de projeto que qual forneça uma interface única e simplificada para os recursos e facilidades mais gerais de um subsistema. Este padrão de projeto é conhecido como:

- ☐ Flyweight
- ☐ Composite;
- ☐ Decorator
- ☐ Proxy;
- ☒ Façade;

**Gabarito
Coment.**

< Tópico Anterior

Próximo Tópico >



3. (FCC - 2011) No contexto dos padrões de projeto:

I. Oferecer uma interface simples para uma coleção de classes.

II. Desacoplar uma abstração de sua implementação para que ambas possam variar independentemente.

Correspondem respectivamente a

- ☐ Adapter e Façade.
- ☐ Bridge e Adapter.
- ☐ Façade e Composite.
- ☐ Composite e Bridge.
- ✓ ☒ Façade e Bridge.



**Gabarito
Coment.**

Saiba mais +

4. Qual padrão de projeto tem como propósito prover interface unificada para conjunto de interfaces em um subsistema, definindo uma interface de alto-nível.

- ☐ Singleton
- ☐ Strategy
- ✓ ☒ Facade
- ☐ Decorator
- ☐ Factory Method

Explicação: O Padrão Façade é do tipo estrutural . É usado quando um sistema é muito complexo ou difícil de entender, já que possui um grande número de classes independentes ou se trechos de código fonte estão indisponíveis. Este padrão esconde as complexidades de um sistema maior e provê uma interface simplificada ao cliente. Tipicamente envolve uma única classe responsável por englobar uma série de membros requeridos pelo cliente. Estes membros acessam o sistema em nome do Façade e escondem os detalhes de implementação.

Saiba mais +

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

- ✓ ☒ O Façade 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.
- ☐ O Adapter pode ser usado quando objetos se comunicam de forma definida, mas complexa; as interdependências entre os objetos são difíceis de entender; o reúso está sendo dificultado, pois um objeto se comunica com vários outros.
- ☐ 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.
- ☐ 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.
- ☐ 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.

Explicação:

Essa opção está correta porque o padrão **Façade** (fachada) é um padrão Fachada, ou seja, faz uma fachada única para todas as fachadas, ou seja, Fornece uma interface unificada para um



◀ Tópico Anterior

Próximo Tópico ▶



Explicação:

Essa opção está correta porque o padrão **Facade** (fachada) é um padrão Fachada, ou seja, faz uma fachada única para todas as fachadas, ou seja, Fornece uma interface unificada para um conjunto de objetos que constituem um subsistema. Define uma interface a um nível mais elevado, para tornar um subsistema mais fácil de usar.

Oferece uma interface única para um conjunto de interfaces de um subsistema. Facade define uma interface de nível mais elevado que torna o subsistema mais fácil de usar.

Define uma interface de mais alto nível que torna o subsistema mais fácil de usar.

[Saiba mais +](#)

6. O padrão _____ controla o acesso ao objeto original. Fornece um substituto ou marcador da localização de outro objeto para controlar o seu acesso.

Qual padrão abaixo corresponde ao descrito acima?

- ✓ ☒ Proxy
- ☐ Singleton
- ☐ Adapter
- ☐ Memento
- ☐ Interpreter

Explicação:

O padrão **Proxy** (procurador) controla o acesso a um objeto através de outro objeto, que atua como seu substituto, ou como um envoltório.

[Saiba mais +](#)

7. Converter a interface de uma classe em outra interface esperada pelos clientes. Permite a comunicação entre classes que não poderiam trabalhar juntas devido à incompatibilidade de suas interfaces. Para qual padrão de projeto a afirmação se relaciona?

- ☐ Memento
- ☐ Proxy
- ☐ Singleton
- ☐ Observer
- ✓ ☒ Nra

[Saiba mais +](#)

8. O padrão Flyweight é um Padrão Estrutural GoF. Qual o melhor momento em que podemos usar esse padrão?

I- Quando o tamanho do conjunto de objetos for significativamente menor que a quantidade de vezes em que eles são usados na aplicação.

[Tópico Anterior](#)[Próximo Tópico >](#)

7. Converter a interface de uma classe em outra interface esperada pelos clientes. Permite a comunicação entre classes que não poderiam trabalhar juntas devido à incompatibilidade de suas interfaces. Para qual padrão de projeto a afirmação se relaciona?

- ☐ Memento
☐ Proxy
☐ Singleton
☐ Observer
☒ Nra

8. O padrão Flyweight é um Padrão Estrutural GoF. Qual o melhor momento em que podemos usar esse padrão?

I- Quando o tamanho do conjunto de objetos for significativamente menor que a quantidade de vezes em que eles são usados na aplicação.

II- Quando objetos podem ser usados em diferentes contextos ao mesmo tempo (agindo sempre como um objeto independente).

III- Quando for necessário elaborar um algoritmo ou algo complicado para separar objetos mutáveis de imutáveis.

Est(á)ão correta(s) a(s) afirmativa(s):

- ☒ ☐ Somente os itens I e II;
☐ Somente os itens I e III;
☒ ☒ Somente o item II;
☐ Somente o item I;
☐ Somente os itens II e III;

Explicação:

O item III está **errado** porque não devemos usar esse padrão quando isso ocorre.

Col@bore

Sugira! Sinalize! Construa!
Antes de finalizar, clique aqui para dar a sua opinião sobre as questões deste exercício.

☐ Não Respondida

☐ Não Gravada

☒ Gravada

Exercício iniciado em 16/05/2020 12:52:17.



< Tópico Anterior

Próximo Tópico >

