

### **Atividade Supervisionada AS9(a)**

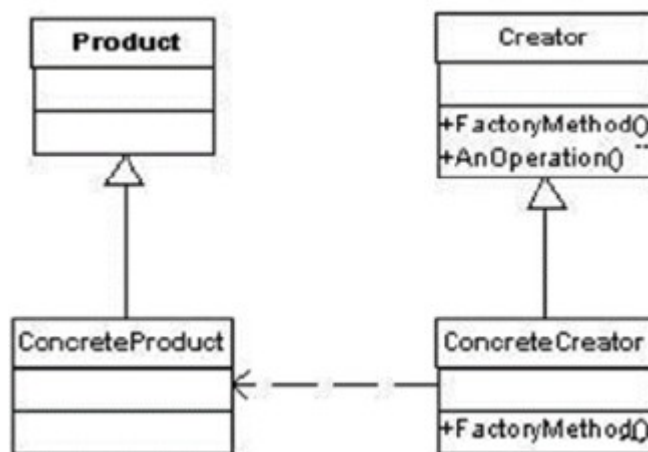
1. Em programação orientada a objetos, o padrão de projeto denominado *Singleton* define uma classe que possui apenas uma instância e provê um ponto de acesso local a ela.

C. Certo

E. Errado

2 – Conhecido como padrões da gangue dos quatro, por terem sido desenvolvidos por quatro autores, os Padrões GoF (Group of Four) estão divididos pelas seguintes famílias de padrões: Padrões de Criação ou de Construção, Padrões Estruturais e Padrões Comportamentais.

Considerando os padrões de Criação ou de Construção, analise o modelo abaixo e em seguida marque a alternativa que define a representação



A. *MEDIATOR*.

B. *SINGLETON*.

C. *FACTORY METHOD*.

D. *FACADE*.

E. *BUILDER*.

3. Por meio do padrão *Iterator*, é possível definir o acesso direto aos métodos de manipulação aos elementos de um objeto agregado.
- C. Certo
- E. Errado
4. Dentre as alternativas abaixo identifique a que NÃO define uma situação em que deve ser utilizado o padrão *Factory Method*?
- A. Quando uma classe quer que suas subclasses especifiquem os objetos criados.
- B. Quando o algoritmo de criação de um objeto deve ser independente das suas partes constituintes e da maneira como ele é "montado".
- C. Quando uma classe (o criador) não pode antecipar a classe dos objetos que deve criar.
- D. Quando se quer localizar num ponto único a conhecimento de qual subclasse está sendo usada.
- E. Quando classes delegam responsabilidade para uma entre várias subclasses de apoio.
5. Para converter a interface de uma classe em outra interface esperada pelo cliente, é correto utilizar o padrão *Adapter*.
- C. Certo
- E. Errado
6. Assinale a afirmativa correta sobre o padrão *Builder*:
- A. A legibilidade da solução final, ou seja, para entender como um objeto é criado e sob quais condições, fica comprometida.
- B. Um dos principais objetivos do padrão *Builder* é separar o algoritmo de criação de um objeto complexo tanto da especificação, quanto das partes que o compõem.
- C. Deve-se separar no construtor da própria classe a lógica para criação de um objeto e concentrar a lógica de criação em uma hierarquia de herança.
- D. é uma abordagem que não facilita a criação de objetos com diferentes configurações e representações, tornando o código dependente da complexidade das classes relacionadas.
- E. Deve-se embutir no construtor da própria classe a lógica para criação de um objeto ou ainda distribuir a lógica de criação em vários métodos adicionais.
7. Por meio do padrão *Prototype*, é possível especificar uma interface genérica para o acesso a informações internas da classe.
- C. Certo
- E. Errado
8. Um dos princípios do padrão de arquitetura MVC é a separação da lógica da apresentação do modelo.

## PORQUE

É, muitas vezes, mais prático manter a visão e o controlador unificados, inclusive, por demandas da tecnologia.

Analisando-se as afirmações acima, conclui-se que ...

- A.** a primeira afirmação é falsa, e a segunda é verdadeira.
- B.** As duas afirmações são falsas.
- C.** as duas afirmações são verdadeiras, e a segunda justifica a primeira.
- D.** a primeira afirmação é verdadeira, e a segunda é falsa.
- E.** as duas afirmações são verdadeiras, e a segunda não justifica a primeira.

9. O padrão de projeto *singleton* é usado para restringir

- A.** a instanciação de uma classe para objetos simples.
- B.** a instanciação de uma classe para apenas um objeto.
- C.** a quantidade de classes.
- D.** as relações entre classes e objetos.
- E.** classes de atributos complexos.

10. O padrão de projeto *Factory* provê uma classe de decisão que retorna

- A.** um objeto de uma de suas subclasses, sem fixação de parâmetros.
- B.** um atributo de uma de suas classes conexas, com base em um parâmetro reservado.
- C.** um objeto de uma de suas subclasses, com base em um parâmetro recebido.
- D.** um atributo de uma de suas classes conexas, sem fixação de parâmetros.
- E.** um objeto de uma de suas subclasses, com parâmetros fatorados.