

# Padrões de Projeto de Software

## Lista de Exercícios AV1 01

Luiz Leão – [luizleao@gmail.com](mailto:luizleao@gmail.com)

<http://www.luizleao.com>



**Estácio**

## Questão 1

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.

## Questão 1 – Resposta

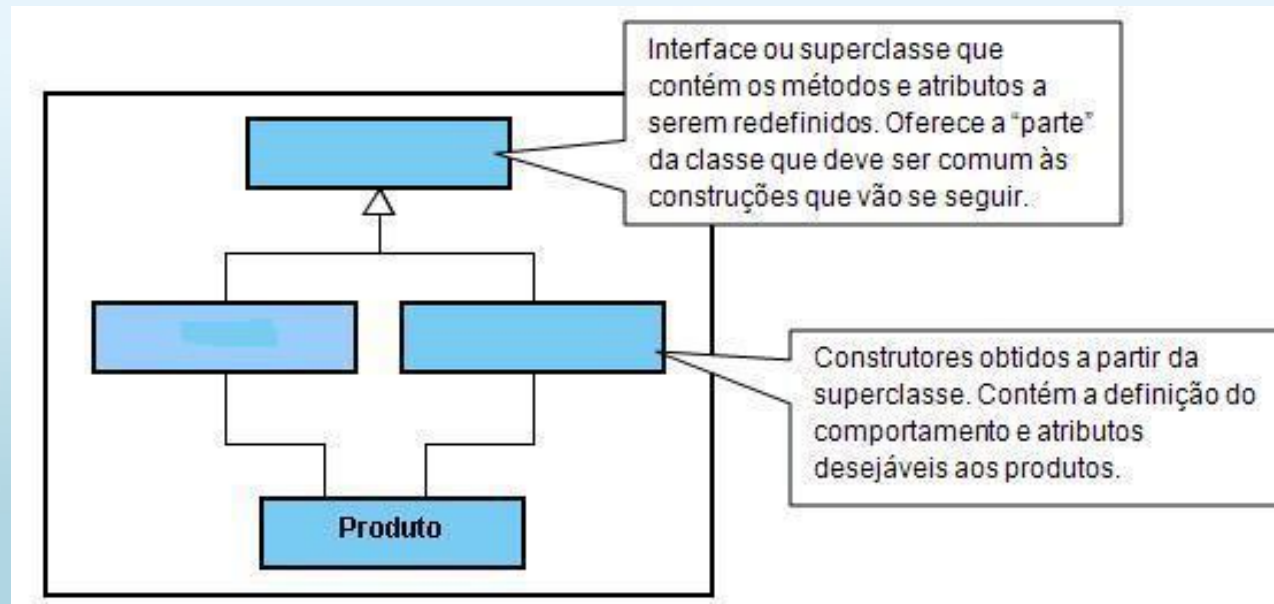
Dentre as alternativas abaixo identifique a que NÃO define uma situação em que deve ser utilizado o padrão *Factory Method*?

- 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.

## Questão 2

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

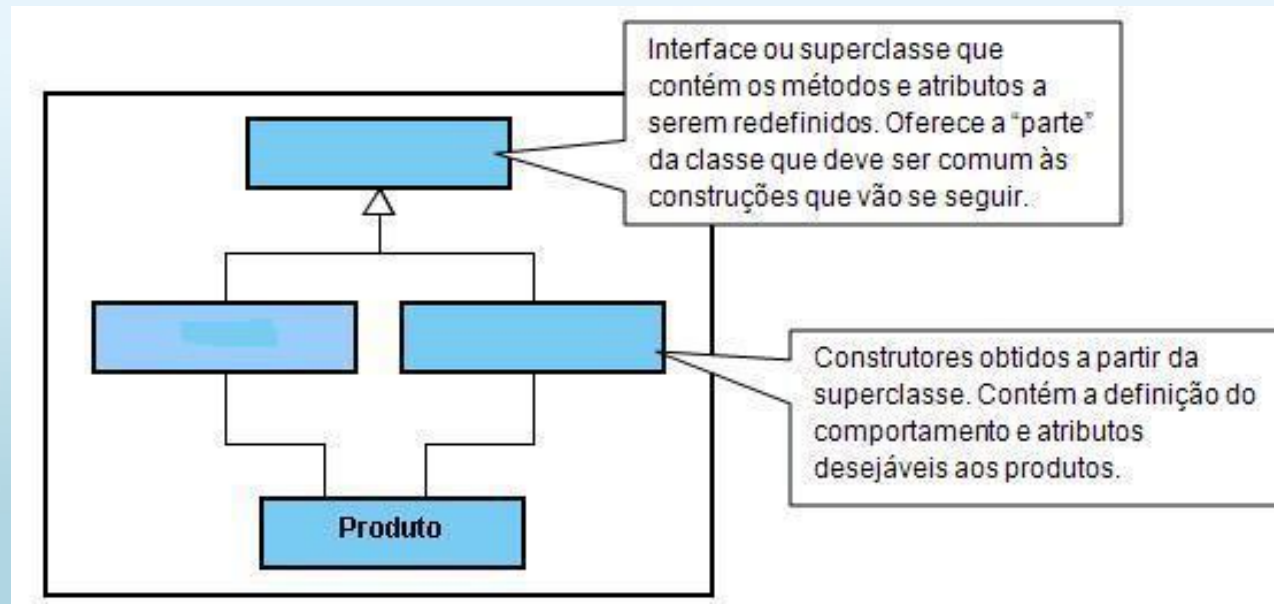
- a) MEDIATOR.
- b) BUILDER.
- c) FACADE.
- d) SINGLETON.
- e) FACTORY METHOD.



## Questão 2 – Resposta

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

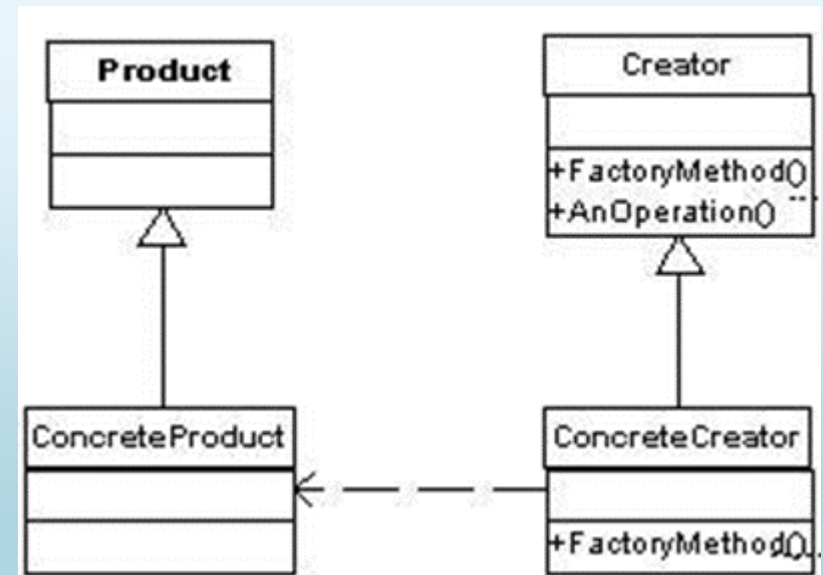
- a) MEDIATOR.
- b) BUILDER.**
- c) FACADE.
- d) SINGLETON.
- e) FACTORY METHOD.



## Questão 3

Considerando os padrões de Criação ou de Construção do GoF, 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.

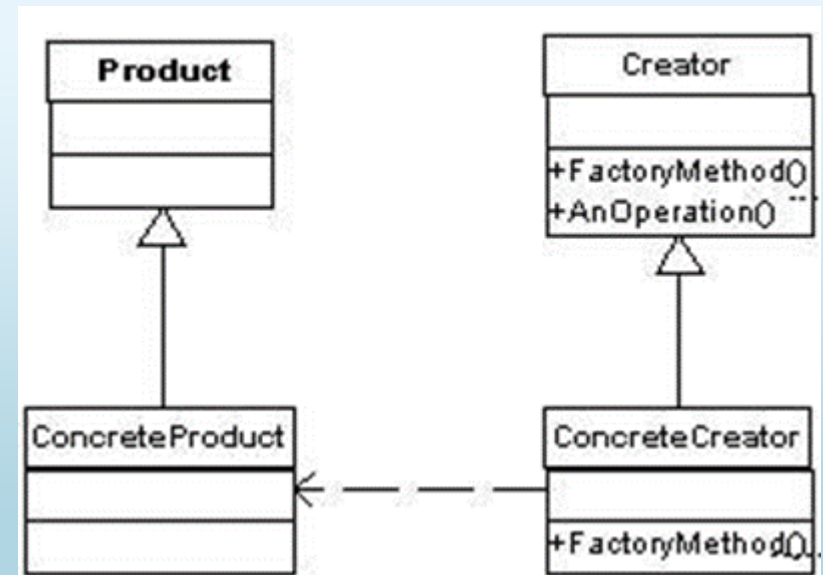




## Questão 3 – Resposta

Considerando os padrões de Criação ou de Construção do GoF, 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.



## Questão 4

A família de padrões GoF é dividida em três grupos principais de padrões, a saber:

- a) Padrões de Criação; Padrões Metodológicos; Padrões de Ponte.
- b) Padrões de Processo; Padrões de Singularidade; Padrões de Prototipação.
- c) Padrões de Proxy; Padrões de Criação; Padrões de Encadeamento.
- d) Padrões Estruturais; Padrões de Processo; Padrões de Responsabilidade.
- e) Padrões Comportamentais; Padrões de Criação; Padrões Estruturais.



## Questão 4 – Resposta

A família de padrões GoF é dividida em três grupos principais de padrões, a saber:

- a) Padrões de Criação; Padrões Metodológicos; Padrões de Ponte.
- b) Padrões de Processo; Padrões de Singularidade; Padrões de Prototipação.
- c) Padrões de Proxy; Padrões de Criação; Padrões de Encadeamento.
- d) Padrões Estruturais; Padrões de Processo; Padrões de Responsabilidade.
- e) Padrões Comportamentais; Padrões de Criação; Padrões Estruturais.**

## Questão 5

Baseando-se nas necessidades apresentadas do lado direito do quadro abaixo, relacione-as ao padrão adequado a utilização e, em seguida marque a alternativa que corresponde a sequência numerada correspondente.

1- <u>Factory Method</u>	( <u>    </u> ) Para construir objetos complexos em várias etapas e/ou que possuem representações diferentes.
2- <u>Abstract Factory</u>	( <u>    </u> ) Para criar objetos usando outro como base.
3- <u>Prototype</u>	( <u>    </u> ) Para isolar a classe concreta do produto criado da interface usada pelo cliente.
4- <u>Builder</u>	( <u>    </u> ) Para criar famílias inteiras de objetos que têm algo em comum sem especificar suas interfaces.

- a) 4 - 2 - 1 - 3
- b) 4 - 3 - 1 - 2
- c) 2 - 3 - 4 - 1
- d) 4 - 1 - 2 - 3
- e) 3 - 4 - 1 - 2

## Questão 5 – Resposta

Baseando-se nas necessidades apresentadas do lado direito do quadro abaixo, relacione-as ao padrão adequado a utilização e, em seguida marque a alternativa que corresponde a sequência numerada correspondente.

1- <u>Factory Method</u>	<input type="checkbox"/> Para construir objetos complexos em várias etapas e/ou que possuem representações diferentes.
2- <u>Abstract Factory</u>	<input type="checkbox"/> Para criar objetos usando outro como base.
3- <u>Prototype</u>	<input type="checkbox"/> Para isolar a classe concreta do produto criado da interface usada pelo cliente.
4- <u>Builder</u>	<input type="checkbox"/> Para criar famílias inteiras de objetos que têm algo em comum sem especificar suas interfaces.

- a) 4 - 2 - 1 - 3
- b) 4 - 3 - 1 - 2**
- c) 2 - 3 - 4 - 1
- d) 4 - 1 - 2 - 3
- e) 3 - 4 - 1 - 2

## Questão 6

Ter uma baixa coesão nos objetos do sistema pode gerar difícil compreensão e reutilização, além de afetar a manutenibilidade. Nesse contexto, o que é ter baixa coesão?

## Questão 6 – Resposta

Ter uma baixa coesão nos objetos do sistema pode gerar difícil compreensão e reutilização, além de afetar a manutenibilidade. Nesse contexto, o que é ter baixa coesão?

**É quando se tem uma mesma classe executando muitos trabalhos, realizando muitas coisas não relacionadas.**

## Questão 7

Considere as seguintes assertivas sobre as vantagens do uso de Padrões de Projeto (*Design Patterns*):

- I. Padrões de projeto proporcionam um vocabulário comum de projeto, facilitando comunicação, documentação e aprendizado dos sistemas de software.
- II. Padrões de projeto auxiliam no desenvolvimento de software por meio da reutilização do projeto de soluções computacionais já testadas e aprovadas.
- III. Uma biblioteca de padrões pode ajudar a melhorar e padronizar o desenvolvimento de software.

As assertivas corretas são:

- a) somente I e III.
- b) somente II e III.
- c) somente I e II.
- d) I, II e III.
- e) somente II.



## Questão 7 – Resposta

Considere as seguintes assertivas sobre as vantagens do uso de Padrões de Projeto (*Design Patterns*):

- I. Padrões de projeto proporcionam um vocabulário comum de projeto, facilitando comunicação, documentação e aprendizado dos sistemas de software.
- II. Padrões de projeto auxiliam no desenvolvimento de software por meio da reutilização do projeto de soluções computacionais já testadas e aprovadas.
- III. Uma biblioteca de padrões pode ajudar a melhorar e padronizar o desenvolvimento de software.

As assertivas corretas são:

- a) somente I e III.
- b) somente II e III.
- c) somente I e II.
- d) I, II e III.**
- e) somente II.

## Questão 8

O uso de classes "statics" garante que somente uma instância estará em memória e que a destruição pelo "garbage collection" será mais rápida do que o uso do padrão singleton. Por que então devemos usar o padrão singleton?

## Questão 8 – Resposta

O uso de classes "statics" garante que somente uma instância estará em memória e que a destruição pelo "garbage collection" será mais rápida do que o uso do padrão singleton. Por que então devemos usar o padrão singleton?

**Porque uma classe static SEMPRE é carregada na memória quando a aplicação é executada e a classe SINGLETON não, sendo carregada na memória quando solicitada a primeira instância.**

## Questão 9

Sobre padrões de projeto escolha a opção INCORRETA:

- a) Padrões de projeto estão relacionados a diferentes níveis de abstração no desenvolvimento de aplicações orientadas a objetos, podendo aparecer ao longo de todo ciclo de análise e projeto de um sistema.
- b) A diversidade de padrões disponíveis é bastante grande, pode-se ter, por exemplo, padrões arquiteturais, padrões de análise, padrões de projeto e padrões de código.
- c) Cada padrão descreve um problema que ocorrem repetidas vezes em nosso ambiente e fornece o núcleo da solução para aquele problema, de tal maneira que se pode usar essa solução milhões de vezes sem nunca fazê-la da mesma forma.
- d) Os padrões de projeto são descrições de objetos que se comunicam e classes que são customizadas para resolver um problema genérico de design em um contexto específico.
- e) Um padrão de projeto define uma estrutura que obrigatoriamente não poderá ser alterada pelo desenvolvedor.

## Questão 9 – Resposta

Sobre padrões de projeto escolha a opção INCORRETA:

- a) Padrões de projeto estão relacionados a diferentes níveis de abstração no desenvolvimento de aplicações orientadas a objetos, podendo aparecer ao longo de todo ciclo de análise e projeto de um sistema.
- b) A diversidade de padrões disponíveis é bastante grande, pode-se ter, por exemplo, padrões arquiteturais, padrões de análise, padrões de projeto e padrões de código.
- c) Cada padrão descreve um problema que ocorrem repetidas vezes em nosso ambiente e fornece o núcleo da solução para aquele problema, de tal maneira que se pode usar essa solução milhões de vezes sem nunca fazê-la da mesma forma.
- d) Os padrões de projeto são descrições de objetos que se comunicam e classes que são customizadas para resolver um problema genérico de design em um contexto específico.
- e) Um padrão de projeto define uma estrutura que obrigatoriamente não poderá ser alterada pelo desenvolvedor.**

## Questão 10

Os métodos polimórficos utilizam os conceitos de overloading e overriding. Apresente a diferença entre os dois conceitos.



## Questão 10 – Resposta

Os métodos polimórficos utilizam os conceitos de *overloading* e *overriding*. Apresente a diferença entre os dois conceitos.

***Overloading***: Sobrecarga de Método. Reescrever o método com uma nova assinatura (atributos de entrada)

***Overriding***: Sobrescrita de Método. Reescrever o método na classe filho (especialista), mesmo que ele já tenha sido implementada na classe mãe (generalista), com propósitos específicos para tal.

## Questão 11

Segundo Metsker, o padrão de projeto GoF \_\_\_\_\_ é aplicado para substituir a geração de instâncias não-inicializadas de uma classe, fornecendo novos objetos a partir de uma classe-exemplo.

- a) MEDIATOR.
- b) SINGLETON.
- c) PROTOTYPE.
- d) BUILDER.
- e) FACTORY METHOD.

## Questão 11 – Resposta

Segundo Metsker, o padrão de projeto GoF \_\_\_\_\_ é aplicado para substituir a geração de instâncias não-inicializadas de uma classe, fornecendo novos objetos a partir de uma classe-exemplo.

- a) MEDIATOR.
- b) SINGLETON.
- c) PROTOTYPE.**
- d) BUILDER.
- e) FACTORY METHOD.

## Questão 12

Assinale a afirmativa correta sobre o padrão *Builder*.

- a) Deve-se é separar no construtor da própria classe a lógica para criação de um objeto e concentrar a lógica de criação em uma hierarquia de herança.
- b) é uma abordagem que não facilita a criação de objetos com diferentes configurações e representações, tornando o código dependente a complexidade das classes relacionadas.
- c) 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.
- d) A legibilidade da solução final, ou seja, para entender como um objeto é criado e sob quais condições, fica comprometida.
- e) Deve-se é embutir no construtor da própria classe a lógica para criação de um objeto ou ainda distribuir a lógica de criação em vários métodos adicionais.

## Questão 12 – Resposta

Assinale a afirmativa correta sobre o padrão *Builder*.

- a) Deve-se é separar no construtor da própria classe a lógica para criação de um objeto e concentrar a lógica de criação em uma hierarquia de herança.
- b) é uma abordagem que não facilita a criação de objetos com diferentes configurações e representações, tornando o código dependente a complexidade das classes relacionadas.
- c) 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.**
- d) A legibilidade da solução final, ou seja, para entender como um objeto é criado e sob quais condições, fica comprometida.
- e) Deve-se é embutir no construtor da própria classe a lógica para criação de um objeto ou ainda distribuir a lógica de criação em vários métodos adicionais.

## Questão 13

Em padrão de projeto existe uma situação onde uma classe chama um método abstrato especificado em alguma classe abstrata (ou interface) e a subclasse concreta vai decidir que tipo exato de objeto criar e retornar.

Baseado nessa descrição marque a alternativa que aponta o padrão relacionado.

- a) Singleton.
- b) Builder.
- c) Mediator.
- d) Factory Method.
- e) Facade.



## Questão 13 – Resposta

Em padrão de projeto existe uma situação onde uma classe chama um método abstrato especificado em alguma classe abstrata (ou interface) e a subclasse concreta vai decidir que tipo exato de objeto criar e retornar.

Baseado nessa descrição marque a alternativa que aponta o padrão relacionado.

- a) Singleton.
- b) Builder.
- c) Mediator.
- d) Factory Method.**
- e) Facade.

## Questão 14

Uma classe com alto acoplamento faz com que o reuso fique comprometido. Apresente uma justificativa para esse problema.

## Questão 14 – Resposta

Uma classe com alto acoplamento faz com que o reuso fique comprometido. Apresente uma justificativa para esse problema.

**O problema ocorre pois para reaproveitar métodos acoplados torna-se necessária a presença adicional das classes relacionadas, dificultando o processo de reutilização.**