

# Padrões de Projeto de Software

## Lista de Exercícios AV2 - 01

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

<http://www.luizleao.com>



**Estácio**

## Questão 1

Qual o objetivo dos padrões Comportamentais, segundo o catálogo GOF?

## Questão 1 – Resposta

Qual o objetivo dos padrões Comportamentais, segundo o catálogo GOF?

***Definir formas de gerenciar e combinar diferentes comportamentos e responsabilidades de classes e objetos de uma aplicação.***

## Questão 2

Qual o objetivo dos padrões Estruturais, segundo o catálogo GOF?

## Questão 2 – Resposta

Qual o objetivo dos padrões Estruturais, segundo o catálogo GOF?

***Facilitar o design do sistema identificando maneiras de realizar o relacionamento entre as entidades, deixando o desenvolvedor livre desta preocupação.***

## Questão 3

Qual o objetivo dos padrões de Criação, segundo o catálogo GOF?

## Questão 3 – Resposta

Qual o objetivo dos padrões de Criação, segundo o catálogo GOF?

***Abstrair o processo de criação de objetos, ou seja, a sua instanciação. Desta maneira o sistema não precisa se preocupar com questões sobre, como o objeto é criado, como é composto, qual a sua representação real.***

## Questão 4

Assinale a opção cujos padrões de projeto estão todos classificados como Comportamentais, segundo o catálogo GoF

- A) Command, Bridge, Iterator, Mediator, Observer, State, Strategy
- B) Command, Bridge, Iterator, Mediator, Bridge, State, Strategy
- C) Command, Bridge, Iterator, Mediator, Composite, State, Strategy
- D) Command, Interpreter, Iterator, Mediator, Observer, State, Strategy
- E) Command, Interpreter, Iterator, Mediator, composite, State, Strategy



## Questão 4 – Resposta

Assinale a opção cujos padrões de projeto estão todos classificados como Comportamentais, segundo o catálogo GoF

- A) Command, Bridge, Iterator, Mediator, Observer, State, Strategy
- B) Command, Bridge, Iterator, Mediator, Bridge, State, Strategy
- C) Command, Bridge, Iterator, Mediator, Composite, State, Strategy
- D) Command, Interpreter, Iterator, Mediator, Observer, State, Strategy**
- E) Command, Interpreter, Iterator, Mediator, composite, State, Strategy

## Questão 5

Assinale a opção cujos padrões de projeto estão todos classificados como criação, segundo o catálogo GoF:

- A) Command, Builder, Factory Method, Prototype, Singleton
- B) Abstract Factory, Builder, Factory Method, Decorator, Singleton
- C) Abstract Factory, Builder, Factory Method, Prototype, Singleton
- D) Abstract Factory, Builder, Composite, Prototype, Singleton
- E) Abstract Factory, Bridge, Factory Method, Prototype, Singleton

## Questão 5 – Resposta

Assinale a opção cujos padrões de projeto estão todos classificados como criação, segundo o catálogo GoF:

- A) Command, Builder, Factory Method, Prototype, Singleton
- B) Abstract Factory, Builder, Factory Method, Decorator, Singleton
- C) Abstract Factory, Builder, Factory Method, Prototype, Singleton**
- D) Abstract Factory, Builder, Composite, Prototype, Singleton
- E) Abstract Factory, Bridge, Factory Method, Prototype, Singleton

## Questão 6

Assinale a opção cujos padrões de projeto estão todos classificados como Estruturais, segundo o catálogo GoF:

- A) Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Singleton
- B) Adapter, Bridge, Prototype, Decorator, Façade, Flyweight, Singleton
- C) Singleton, Bridge, Composite, Decorator, Façade, Flyweight, Proxy
- D) Singleton, Bridge, Prototype, Decorator, Façade, Flyweight, Proxy
- E) Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy

## Questão 6 – Resposta

Assinale a opção cujos padrões de projeto estão todos classificados como Estruturais, segundo o catálogo GoF:

- A) Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Singleton
- B) Adapter, Bridge, Prototype, Decorator, Façade, Flyweight, Singleton
- C) Singleton, Bridge, Composite, Decorator, Façade, Flyweight, Proxy
- D) Singleton, Bridge, Prototype, Decorator, Façade, Flyweight, Proxy
- E) Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy**

## Questão 7

O que é o MVC e como ele é estruturado?

## Questão 7 – Resposta

O que é o MVC e como ele é estruturado?

*É um padrão arquitetura que tem como objetivo separar a apresentação dos dados da sua manipulação. O MVC é composto, basicamente, de 3 camadas:*

*View(Visão) – Apresentação do sistema para o usuário*

*Controller(Controle) – Efetuará o processamento das informações fornecidas pela visão*

*Model(Modelo) – Responsável pela persistência (armazenamento) dos dados na estrutura escolhida (Banco de Dados, XML, Etc.)*

## Questão 8

Baixo acoplamento é um princípio-chave na Orientação objetos. O que deve ser feito para garantir o baixo acoplamento na relação entre classes e objetos de um sistema?



## Questão 8 – Resposta

Baixo acoplamento é um princípio-chave na Orientação objetos. O que deve ser feito para garantir o baixo acoplamento na relação entre classes e objetos de um sistema?

***Deve-se manter os objetos/classe o mais independente possível, ou seja, não devemos evitar que objetos/classe dependam de forma excessiva de outros objetos/classe do sistema.***

## Questão 9

Em uma janela pode-se adicionar objetos como barras de rolagem, caixas de texto, labels, etc. Pode-se criar uma classe Auxiliar que será estendida pelos decoradores que irão inserir propriedades na janela. O padrão de projetos que possibilita essa implementação chama-se:

- A) Memento
- B) Template Method
- C) Decorator
- D) Prototype
- E) Builder

## Questão 9 – Resposta

Em uma janela pode-se adicionar objetos como barras de rolagem, caixas de texto, labels, etc. Pode-se criar uma classe Auxiliar que será estendida pelos decoradores que irão inserir propriedades na janela. O padrão de projetos que possibilita essa implementação chama-se:

- A) Memento
- B) Template Method
- C) Decorator**
- D) Prototype
- E) Builder

## Questão 10

Dois dos principais patterns utilizados atualmente são descritos a seguir:

I. Visa garantir que uma classe só tenha uma única instância e prover um ponto de acesso global a ela.

II. Visa definir uma dependência um-para-muitos entre objetos para que quando um objeto mudar de estado os seus dependentes sejam notificados e atualizados automaticamente.

Os Design Patterns descritos em I e II são, respectivamente:

- A) Singleton e Observer.
- B) Composite e Adapter
- C) Singleton e Command.
- D) Facade e Observer.
- E) Facade e Adapter.

## Questão 10 – Resposta

Dois dos principais patterns utilizados atualmente são descritos a seguir:

I. Visa garantir que uma classe só tenha uma única instância e prover um ponto de acesso global a ela.

II. Visa definir uma dependência um-para-muitos entre objetos para que quando um objeto mudar de estado os seus dependentes sejam notificados e atualizados automaticamente.

Os Design Patterns descritos em I e II são, respectivamente:

**A) Singleton e Observer.**

B) Composite e Adapter

C) Singleton e Command.

D) Facade e Observer.

E) Facade e Adapter.

## Questão 11

Qual a responsabilidade correta relacionada ao padrão GRASP *Creator* (Criador)?

## Questão 11 – Resposta

Qual a responsabilidade correta relacionada ao padrão GRASP *Creator* (Criador)?

***Tem a seguinte responsabilidade de fazer, pois sua funcionalidade está relacionado com a resposta a seguinte pergunta: quem deve criar um objeto A.***

## Questão 12

O padrão de projetos cuja finalidade é oferecer uma interface de acesso a todos os objetos da aplicação é \_\_\_\_\_ e pertence a categoria de padrões do catálogo GOF chamada \_\_\_\_\_. A resposta correta é:

- A) Observer – Criação
- B) Façade – Estrutural
- C) Observer – Estrutural
- D) Façade – Criação
- E) Observer – Comportamental



## Questão 12 – Resposta

O padrão de projetos cuja finalidade é oferecer uma interface de acesso a todos os objetos da aplicação é \_\_\_\_\_ e pertence a categoria de padrões do catálogo GOF chamada \_\_\_\_\_. A resposta correta é:

- A) Observer – Criação
- B) Façade – Estrutural**
- C) Observer – Estrutural
- D) Façade – Criação
- E) Observer – Comportamental