



INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Gaspar

Fase: V Turno: Noturno

U.C.:  
Padrões de Projeto  
de Software

Prof.  
Andreu Carminati

**Orientações:**

1) Esta avaliação a distância terá duração máxima definida na atividade aberta no Moodle e deverá ser realizada **individualmente**.

**Avaliações com respostas similares terão penalização.**

2) Leia atentamente cada questão antes de respondê-la. A interpretação faz parte da avaliação.

4) Respostas deverão ser enviadas em **formato editável ODT e código fonte**.

**Prova 1**

1. Para a realização da prova será necessário utilizar os códigos disponíveis com esta avaliação (leia os códigos com muita atenção). Você deverá terminar a implementação do padrão Builder para o projeto fornecido. Perceba que em **Demo** são instanciadas duas implementações do Builder, uma para criação (através de uma classe Diretor) de um carro e outra para criação de um manual de carro, sendo sua responsabilidade criar estas classes Builder. (Nota \_\_\_\_/5,0).

Classes e interfaces a serem criadas e implementadas:

- Interface Builder.java
- Classe CarroBuilder.java
- Classe ManualDeCarroBuilder.java

A saída do programa deverá obrigatoriamente ser:

```
Carro construido:  
CARRO_ESPORTIVO
```

```
Manual de carro construido:  
Tipo de carro: CARRO_ESPORTIVO  
Contagem de assentos: 2  
Motor: volume - 3.0; kilometragem - 0.0  
Transmissao: SEMI_AUTOMATICO  
Computador de bordo: Funcional  
Navegador GPS: Funcional
```

2. (FGV - 2012 - Senado Federal - Analista Legislativo - Análise de Sistemas) Padrões de Projeto têm sido utilizados com grande sucesso em programação de software, apresentando vantagens, desvantagens e possuindo características próprias. Duas características dos padrões de projeto são (Nota \_\_\_\_/1,0):

- A) melhora a criatividade e aumenta o tamanho dos códigos.
- B) reduz o tamanho dos códigos e diminui a eficiência.
- C) favorece a reusabilidade e melhora a produtividade.
- D) diminui a produtividade dificulta a reusabilidade.
- E) aumenta a eficiência e prejudica a criatividade

3. (FCC - 2018 - SEGEF-MA - Analista Executivo - Programador de Sistemas) A intenção do padrão de projeto Abstract Factory é (Nota \_\_\_\_/1,0):
- A) definir uma interface para a criação de um objeto, deixando as subclasses decidirem que classe instanciar. Delega a instanciação para as subclasses.
  - B) converter a interface de uma classe na interface esperada pelos clientes. Permite que classes com interfaces incompatíveis trabalhem em conjunto.
  - C) fornecer uma interface para a criação de famílias de objetos relacionados ou dependentes sem especificar suas classes completas.
  - D) garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso à mesma.
  - E) separar a construção de um objeto complexo de sua representação, de modo que o mesmo processo de construção possa criar diferentes representações.
4. (INSTITUTO AOCP - 2019 - IBGE - Analista Censitário - Análise de Sistemas - Desenvolvimento de Aplicações Web Mobile) Supõe-se que, ao desenvolver um software, é possível se deparar com uma barreira na hora de fazer a ligação com o banco de dados, pois ele só pode ter uma única instância, visto que é necessário manter a integridade da aplicação. Para esse caso, pode-se utilizar os padrões de projeto, para garantir que uma classe tenha apenas uma instância de si mesma e que forneça um ponto global de acesso a ela. Assim, uma classe gerencia a própria instância dela, além de evitar que qualquer outra classe crie uma instância dela. Assinale a alternativa que apresenta o Padrão de Projeto que possui essas características (Nota \_\_\_\_/1,0).
- A) Singleton.
  - B) Façade.
  - C) Abstract Factory.
  - D) Adapter.
  - E) Observer
5. (FGV - 2019 - DPE-RJ - Técnico Superior Especializado - Tecnologia da Informação) O Governo Federal, por meio do Banco Central, está desenvolvendo um sistema que possibilitará a todos os Bancos do país o acesso a algumas de suas informações. Um requisito fundamental desse sistema é que a taxa de juros utilizada em todas as transações de todos os Bancos seja a mesma e haja um único acesso a essa informação. Além disso, esse sistema deve poder ser executado em diferentes plataformas, como computadores e diversos dispositivos móveis. Para garantir que a taxa de juros seja única e para evitar a necessidade de criar diferentes soluções para cada plataforma, a empresa desenvolvedora decidiu adotar padrões de projeto. O primeiro padrão deverá garantir uma única instanciação para a classe "Taxa de juros"; e o segundo padrão deverá definir uma família de componentes para cada plataforma e uma implementação que os instancie de acordo com a plataforma na qual a aplicação estará sendo executada. Os padrões de projeto a serem adotados nessa implementação são, respectivamente (Nota \_\_\_\_/1,0):
- A) Prototype e Adapter;
  - B) Singleton e Abstract Factory;
  - C) Template Method e Prototype;
  - D) Adapter e Singleton;
  - E) Abstract Factory e Command.

6. (INSTITUTO AOCP - 2014 - MPE-BA - Analista Técnico - Sistemas) Nos padrões GoF, o padrão Builder é constituído, dentre os seus elementos, do "builder" e "concrete builder". A diferença entre eles, respectivamente, é dada por qual alternativa (Nota \_\_\_\_/1,0)?

A) O primeiro especifica uma interface para um construtor de partes do objeto-produto, enquanto que o segundo constrói um objeto utilizando a interface do builder.

B) O primeiro constrói um objeto utilizando a interface do concrete builder, enquanto que o segundo especifica uma interface para um construtor de partes do objeto-produto.

C) O primeiro especifica uma interface para um construtor de partes do objeto-produto, enquanto que o segundo define uma implementação da interface builder além de manter a representação que cria e fornece a interface para recuperação do produto.

D) O primeiro representa o objeto complexo acabado de construir e inclui classes que definem as partes constituintes, enquanto que o segundo especifica uma interface para um construtor de partes do objeto-produto.

E) O primeiro define uma implementação da interface builder além de manter a representação que cria e fornece a interface para recuperação do produto, enquanto que o segundo representa o objeto complexo acabado de construir e inclui classes que definem as partes constituintes.