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 2

1. Nesta primeira questão você deverá aplicar o padrão **composite** para representação e avaliação de expressões matemáticas em valores reais (double). Expressões matemáticas são expressões que são compostas por expressões, tal como pressupõe o padrão em questão. A sua implementação deverá ter como base o seguinte caso de teste, sendo que não será fornecido código auxiliar:

```
public class Demo {

      public static void main(String[] args) {

          Expressao e = new Divisao(
                            new Multiplicacao(
                               new Valor(40),
                               new Valor(3)),
                            new Soma(
                               new Valor(2),
                               new Subtracao(
                                  new Valor(10),
                                  new Valor(6.0))));
           // Espressao (em valores reais): (40*3)/(2+(10-6)) = 120

           System.out.println("Resultado: " + e.avaliar());
      }
}
```

A saída do programa deverá obrigatoriamente ser: `Resultado: 20.0`

Dicas: Expressao representa a interface. No exemplo anterior, identifique o que é composite e o que é folha, isto é importante para as decisões de implementação (___/5.0).

2. O propósito do padrão Adapter é separar uma abstração de sua implementação, para que as duas possam variar independentemente. Esta afirmação está correta ou errada? Justifique sua resposta (___/1.0).

3. Considere os seguintes objetivos de padrões de projeto.

   1. Separa a construção de um objeto complexo da sua representação, de forma que o mesmo processo de construção possa criar diferentes representações, ou seja, serve para encapsular a construção de um produto e permitir que ele seja construído em etapas.

2. Atribui responsabilidades adicionais a um objeto dinamicamente. Ele fornece uma alternativa flexível a subclasses para a extensão da funcionalidade, ou seja, envelopa um objeto para fornecer novos comportamentos.

3. Usa compartilhamento para dar suporte a vários objetos de forma eficiente, ou seja, permite que uma instância de uma classe possa ser usada para fornecer muitas "instâncias virtuais".
Correspondem, correta e respectivamente, aos objetivos de padrões de projeto:

Representam, respectivamente, os seguintes padrões (___/1.0):

A) Factory Method, Decorator, Facade
B) Adapter, Facade e Builder.
C) Builder, Mediator e Adapter.
D) Builder, Decorator, Flyweight.
E) Decorator, Factory Method, Abstract Factory.

4. Analise a seguinte afirmação: "Proxy é um padrão de projeto de software apropriado quando vários objetos devem ser manipulados em memória sendo que muitos deles possuem informações repetidas." Esta afirmação está correta ou errada? Justifique sua resposta (___/1.0).

5. Considere as seguintes afirmativas a respeito dos padrões de projeto Facade (Fachada) e Composite.

I Uma das consequências negativas do padrão Facade é que ele aumenta o acoplamento entre os subsistemas no qual é aplicado.

II O padrão Facade define uma interface única para acesso a um subsistema, tornando mais fácil a utilização de seus serviços.

III No padrão Composite, objetos que representam composições e objetivos primitivos são tratados (chamados) de forma idêntica pelo cliente.

IV O padrão Composite permite variar dinamicamente a quantidade de objetos primitivos, mas não de objetos composite.

Estão corretas as afirmativas (___/1.0):

A) I e III
B) I e IV
C) II e III
D) II e IV

6. Sobre padrões de projeto é correto afirmar que (___/1.0):
A) São considerados padrões estruturais: Adapter, Bridge e Builder.

B) São considerados padrões de criação: Abstract Factory, Prototype e Composite.

C) Os padrões "GoF" são organizados em 3 famílias : Padrões de criação, Padrões estruturais e Padrões arquiteturais.

D) Os padrões ajudam a tornar a arquitetura de um framework adequada a aplicações diferentes, minimizando a necessidade de modificações.

E) Um padrão de projeto descreve a arquitetura de um sistema orientado a objetos, os tipos de objetos e as interações entre os mesmos. Ele pode ser vislumbrado como o esqueleto – template – de uma aplicação que pode ser customizado pelo programador e aplicado a um conjunto de aplicações de um mesmo domínio.