



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Arquitetura e Projeto de Sistemas**

**Gabarito da AP2 2019/1**

**Nome –**

**Assinatura –**

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

### Questão 1 (3 pontos)

Observe a classe “Animal” e o trecho de implementação de seu método público “emitirSom”. Considerando que outros tipos de animal e outros métodos públicos, representando outros comportamentos, poderão ser adicionados ao sistema na medida que ele evolui, responda.

Animal
-tipo
+emitirSom()
-latir()
-miar()
-rugir()
-uivar()

```
public class Animal{
    private String tipo;
    ...
    public void emitirSom(){
        if (tipo.equals("Cachorro")){
            latir();
        }
        if (tipo.equals("Gato")){
            miar();
        }
        if (tipo.equals("Leão")){
            rugir();
        }
        ...
    }
}
```

(a) Qual o problema desta solução (1 ponto)?

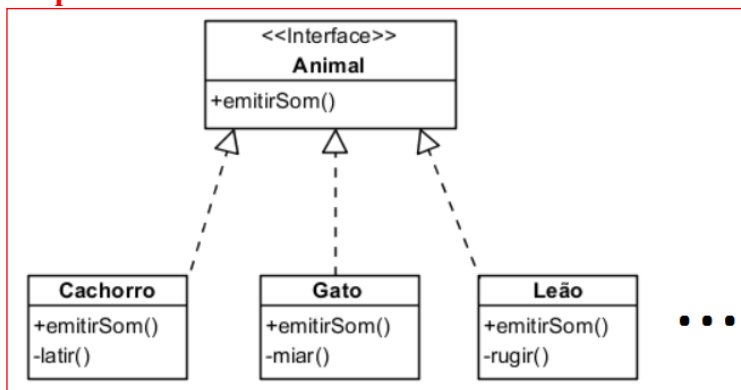
**Resposta:** O problema é a implementação do comportamento com base na testagem do tipo da classe (IF TIPADO).

(b) Qual padrão GRASP poderia ser utilizado para resolver este problema (1 ponto)?

**Resposta:** O padrão GRASP que pode ser utilizado para resolver este problema é o Polimorfismo.

(c) Resolva o problema aplicando o padrão GRASP, desenhando o novo diagrama de classes (1 ponto).

**Resposta:**



Observação: Note que o atributo tipo não é mais necessário. A solução permite facilmente acrescentar novos tipos de animais e novos métodos públicos. Também facilita a manutenção com métodos mais curtos e mais focados.

## Questão 2 (4 pontos)

Para o código fonte a seguir, responda com suas palavras:

```
public class Banco {

    private Banco instance =null;

    public static Banco getInstance() {
        if (instance == null)
            instance = new Banco();
        return instance;
    }

    ...

}
```

(a) Esse código fonte implementa qual padrão GoF (1 ponto)?

**Resposta:** O padrão implementado nesse código fonte é o Singleton.

- (b) Esse código apresenta três problemas, que podem fazer com que o padrão não atinja seu objetivo. Faça uma nova versão desse código corrigindo os problemas. Circule no código os três trechos corrigidos (3 pontos).

**Resposta:**

```
public class Banco {  
  
    private static Banco instance = null;  
  
    private Banco() {  
    }  
  
    public static synchronized Banco getInstance() {  
        if (instance == null)  
            instance = new Banco();  
        return instance;  
    }  
  
    ...  
}
```

### Questão 3 (3 pontos)

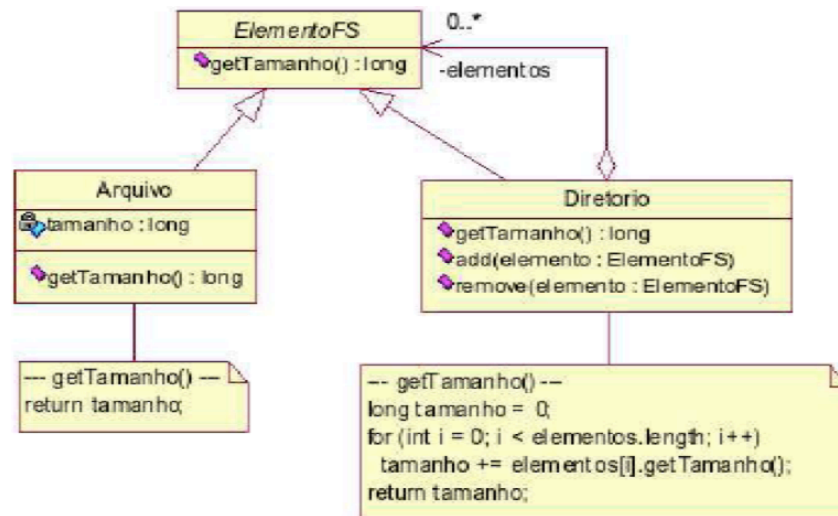
Imagine um sistema que precisa manipular um diretório, composto por arquivos e subdiretórios, recursivamente. Esse sistema deve ser capaz de computar o tamanho de um diretório considerando recursivamente o tamanho dos seus arquivos e subdiretórios.

- (a) Qual padrão de projeto GoF é indicado para essa situação (1 ponto)?

**Resposta: Padrão *Composite*.**

- (b) Faça um diagrama de classes que solucione esse problema fazendo uso do padrão de projeto apropriado, indicando os atributos e métodos necessários para o funcionamento do padrão (1 ponto).

**Resposta:**



- (c) Escreva o código do método de cálculo do tamanho de um diretório na sua linguagem de programação preferida, respeitando o diagrama feito no item anterior (1 ponto).

**Resposta:**

```

long tamanho = 0;
for (int i = 0; i < elementos.length; i++) {
    tamanho += elementos[i].getTamanho();
}
return tamanho;

```