



Universidade do Sul de Santa Catarina – UNISUL

Curso de Ciência da Computação

Disciplina: Programação Orientada a Objetos

Professor: Clávison Martinelli Zapelini

AValiação TEÓRICA I

E-mail: [clavison.zapelini@unisul.br](mailto:clavison.zapelini@unisul.br)

Aluno(a): Guigo Belling

**Observações:**

A avaliação se encerrará exatamente às 22:00 (sem intervalo). Deve ser executada de forma individual e sem consulta. As dúvidas em relação às questões serão esclarecidas nos momentos iniciais juntamente com a leitura da prova. Somente serão corrigidas as questões respondidas à caneta e com letra legível.

\*\*\*\*A CLASSE "ProgramaTV" SERÁ UTILIZADA PARA RESPONDER AS QUESTÕES 1 E 2:

```
public class ProgramaTV {
    private String nome= "YYY";    // YYY
    private double nota = 5.0;    // 5
    public ProgramaTV(){
        this("ABC",8.0);    //
    }
    public ProgramaTV(double p){
        this.nota = p;
    }
    public ProgramaTV(String n, double p){
        this.nome = n;
        this.nota = p;
    }
    public String toString() {
        String classe = "";
        if(nota < 5)
            classe = "RUIM";
        else{
            if(nota >= 5 && nota < 7)
                classe = "BOM";
            else
                classe = "ÓTIMO";
        }
        return this.nome + " - " + classe;
    }
    public ProgramaTV clone(){
        ProgramaTV novo = new ProgramaTV();
        novo.setNome(getNome());
        novo.setNota(getNota());
        return novo;
    }
    public int compareTo(ProgramaTV outro){
        if(this.nota > outro.nota)
            return 1;
        else
            if(this.nota == outro.nota)
                return 0;
        return -1;
    }
}
//GETS E SETS IMPLEMENTADOS
```

n = nome  
p = nota

// ABC  
// 8.0

1

0

-1

1 – De acordo com o conceito de construtores e valores default para os atributos indique o que será escrito na tela ao executar o trecho: (2,0 pontos):

```
public static void main(String[] args) {
    ProgramaTV p1 = new ProgramaTV(); // ABC; 9.0
    ProgramaTV p2 = new ProgramaTV(2.5); // new ProgramaTV("", 2.5)
    ProgramaTV p3 = new ProgramaTV("Jornal Nacional", 9.0);
    System.out.println(p1); // ABC - ÓTIMO
    System.out.println(p2); // XY - RUIM // 2.5
    System.out.println(p3); // Jornal Nacional - ÓTIMO
    p2.setNota(5.5);
    System.out.println(p2); // YY - Bom
```

2 – De acordo com os métodos especiais de classe e a instância de objetos a partir de referências em memória identifique ao final da execução do código abaixo, o que será escrito na tela (3,0 pontos):

```
public static void main(String[] args) {
    ProgramaTV p1 = new ProgramaTV("XXX", 2);
    ProgramaTV p2 = new ProgramaTV(8);
    ProgramaTV p3 = p1.clone();
    ProgramaTV p4 = p3;
    p3.setNome("AAA");
    System.out.println(p1); // XXX - RUIM
    System.out.println(p2); // XY - ÓTIMO
    System.out.println(p3); // AAA - RUIM
    System.out.println(p4); // XXX - RUIM
    System.out.println(p3.compareTo(p2)); // -1
    System.out.println(p4.compareTo(p3)); // 0
```

3 – Um atributo ou método estático deixa de pertencer ao objeto e pertence à classe. Analise a classe "P1Q3" e responda o que será escrito na tela em cada linha destacada (2,00 pontos).

```
public class P1Q3 {
    static int y = 5;
    int x = 10;
    public static void main(String[] args) {
        P1Q3 a1 = new P1Q3();
        P1Q3 a2 = new P1Q3();
        P1Q3.y++; // 6
        a1.y += 3; // 9
        a1.x += 3; // 13
        a2.y += 5; // 11
        a2.x += 5; // 15
        System.out.println(a1.y); // 9
        System.out.println(a1.x); // 13
        System.out.println(a2.y); // 11
        System.out.println(a2.x); // 15
    }
}
```

4 - Um Objeto pode ser composto por outros objetos. A esse conceito, na POO, chamamos de Relacionamento entre classes. Abaixo temos duas classes que se relacionam: Cidade e Voo. Voo TEM-UMA Cidade de origem e TEM-UMA cidade de destino. Analise as classes e indique o que será escrito na tela nas linhas destacadas (3,00 pontos):

```
public class Cidade {
    private String nome,uf;

    public Cidade(String n, String u){
        this.nome = n;
        this.uf = u;
    }

    public String toString() {
        return nome + "(" + uf + ")";
    }

    //GETS E SETS IMPLEMENTADOS
}
```

```
public class Voo {
    private int numero;
    private Cidade origem;
    private Cidade destino;

    public Voo(int n){
        numero = n;
        origem = new Cidade("FLN", "SC");
        destino = new Cidade("BSB", "GO");
    }

    public String toString() {
        return numero + ": " + origem + "/" + destino;
    }

    //GETS E SETS IMPLEMENTADOS
}
```

```
public static void main(String[] args) {
    Voo v1 = new Voo(1);
    Voo v2 = new Voo(2);
    Cidade c1 = new Cidade("GRU", "SP");
    Cidade c2 = new Cidade("POA", "RS");
    v2.setOrigem(c2); // 2: POA(RS) / GRU(SP)
    v2.setDestino(c1); // 1: FLN(SC) / BSB(GO)
    System.out.println(v1); // 1: FLN(SC) / BSB(GO)
    System.out.println(v2); // 2: POA(RS) / GRU(SP)
    Cidade aux = v1.getOrigem();
    v1.setOrigem(v1.getDestino()); // BSB(GO)
    v1.setDestino(aux); // Florianópolis
    System.out.println(v1); // 1: BSB(GO) / FLN(SC)
    System.out.println(v2.getOrigem().getUf()); // POA(RS) RS
    System.out.println(v2.getDestino().getNome()); // GRU(SP) GRU
    v1 = v2;
    System.out.println(v1); // 2: POA(RS) / GRU(SP)
}
```

cidade.getUF() ?