

Unidade Curricular:	Programação Orientada a Objetos
Ano letivo:	2015-01
Carga horária:	72 horas
Professor:	Francisco de Assis S. Santos, Dr.

## Exercícios

1. Considere os códigos abaixo e esboce o diagrama de classes correspondente.

```
public class SemFio extends Telefone{
    private float frequencia, distancia;
    private int canais;

    public SemFio(int c, String s, String m, float p, Dimensao
        int ca, float f, float dis){
        super(c, s, m, p, d); // invocando o construtor da
            superclasse
        this.frequencia = f;
        this.distancia = dis;
        this.canais = ca;
    }

    // sobreescrita do metodo da superclasse
    public void imprimirDados(){
        super.imprimirDados(); // invocando o metodo de mesmo n
            superclasse
        System.out.println("Freq: " + this.frequencia);
        ...
    }
}
```

```

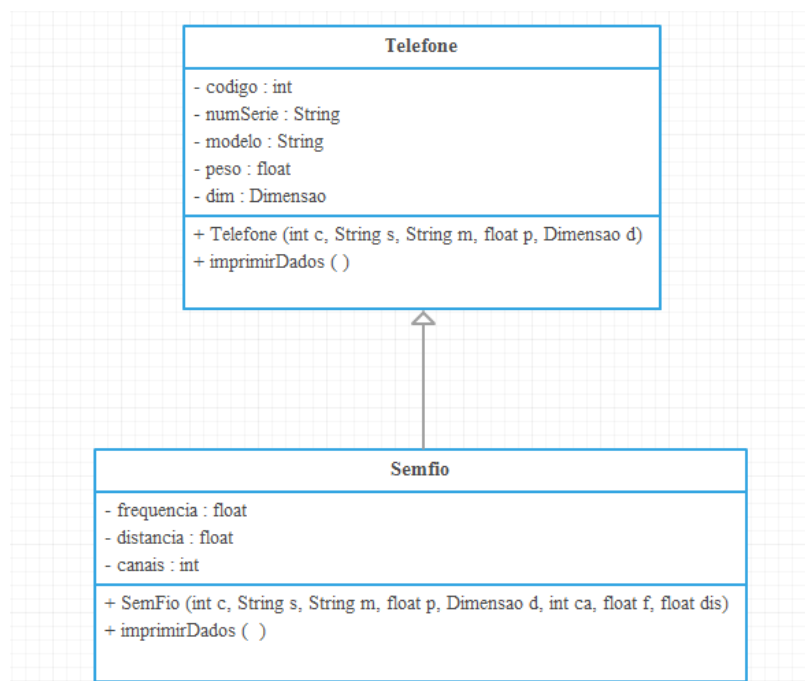
public class Telefone{
    private int codigo;
    private String numSerie, modelo;
    private float peso;
    private Dimensao dim;

    public Telefone(int c, String s, String m, float p, Dimensao d)
    {
        this.codigo = c; this.peso = p; this.dim = d;
        this.numSerie = s;this.modelo = m;
    }

    public void imprimirDados(){
        System.out.println("Codigo: " + this.codigo);
        ...
        this.dim.imprimirDados();
    }
}

```

**Resolução:**



**2. Identifique o(s) erro(s) no código em Java abaixo:**

```

Public void LerArquivo {
    File arquivo = File("dados.dat");
    try {
        FileInputStream fon = new FileOutputStream(arquivo);
        ObjectOutputStream oon = new ObjectOutputStream(fon);

```

```

this.dados = (Livro[ ]) oin.readObject();

oin.close();
fin.close();

for (Livro v : this.dados) {
    v.imprimir();
}
} catch (Exception ex) {
System.er.println("erro: " + ex.toString());
}
}
}

```

#### Resolução:

```

public void LerArquivo () {
    File arquivo = new File("dados.dat");
    try {
        FileInputStream fin = new FileInputStream(arquivo);
        ObjectInputStream oin = new ObjectInputStream(fin);

        this.dados = (Livro[ ]) oin.readObject();

        oin.close();
        fin.close();

        for (Livro v : this.dados) {
            v.imprimir();
        }
    } catch (Exception ex) {
System.er.println("erro: " + ex.toString());
}
}
}

```

### 3. Apresente em Java a codificação mínima necessária para executar dois processos em paralelo. Para isso utilize Threads.

#### Resolução:

```

public class Processo extends Thread{
    public void run{
        //código..
    }
}

public class OutroProcesso extends Thread{
    public void run{
        //código..
    }
}

//implementar na Void Main
Processo P1 = new Processo();

```

```
Thread threadDoProcesso = new Thread(P1);

OutroProcesso P2 = new OutroProcesso ();
Thread threadDoOutroProcesso = new Thread(P2);

threadDoOutroProcesso.start();

threadDoProcesso.start();
```

#### **4. O que é possível realizar com objetos serializados? Como isso pode ser implementado em Java?**

##### **Resolução:**

É possível gravar os objetos em arquivos e, por exemplo, transmiti-los pela rede. Para isso deve-se importar `java.io.Serializable`.

---

---

#### **5) Analise o código abaixo:**

```
public class empregado
{
    public String CPF;
    private String nome;
    private String funcao;
    protected float salario;

    public void salariobase (float valor)
    {
        this.salario= valor;
    }
    public empregado (String CPF, String nome, String funcao)
    {
        this.CPF= CPF;
        this.nome= nome;
        this.funcao= funcao;
        this.salario=0;
    }

    public void imprime ()
    {
        System.out.println("Nome: " +this.nome);
        System.out.println("CPF: " +this.CPF);
        System.out.println("Funcao: " +this.funcao);
        System.out.println("Salario R$: " +this.salario);
    }
}
```

```

public class gerente extends empregado
{
private float comissao;

public void valorcomissao (float valor)
{
this.comissao=valor;

}

public void salariobase (float valor)
{
this.salario=this.comissao+valor;

}

}
public static void main (String [] args)
{
gerente novogerente = new gerente("07789631890", "José", "Gerente");

novogerente.valorcomissao(830);
novogerente.salariobase(4900);
novogerente.imprime();

}
}

```

**Considere as assertivas:**

- I – Todos os atributos da classe gerente são acessíveis somente a própria classe gerente e as suas classes filhas.
- II – Há uma relação de herança e polimorfismo entre as classes empregado e gerente.
- III – O valor do atributo salario do objeto novogerente totalizara em R\$ 5730.
- IV – A classe empregado é uma subclasse da superclasse gerente.
- V – Os conteúdos: "07789631890", "José", 4900 e "Gerente" não serão atribuídos ao objeto novogerente, devido que a classe gerente possui apenas o atributo comissao.

**Assinale a única alternativa correta:**

- A ( ) – Somente as assertivas I, II, e IV estão corretas;
- B ( ) – Somente as assertivas I, III e V estão erradas;
- C ( ) – Somente as assertivas III, IV e V estão corretas;
- D (X) – Somente as assertivas I, IV e V estão erradas;**
- E ( ) – Somente as assertivas I e II estão corretas;

**6) Analise as seguintes afirmações relativas à Programação Orientada a Objetos:**

- I. Em um programa orientado a objetos, as instâncias de uma classe armazenam os mesmos tipos de informações e apresentam o mesmo comportamento.
- II. Em uma aplicação orientada a objetos, podem existir múltiplas instâncias de uma mesma classe.
- III. Em um programa orientado a objetos, as instâncias definem os serviços que podem ser solicitados aos métodos.

IV. Em um programa orientado a objetos, o método construtor não pode ser executado quando a classe a qual pertence é executada.

Indique a opção que contenha todas as afirmações verdadeiras

**A (X) I e II**

B ( ) II e III

C ( ) III e IV

D ( ) I e III

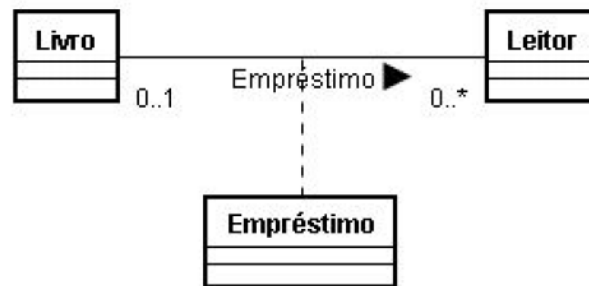
E ( ) II e IV

**7) Considere as seguintes assertivas sobre o modelo de classes mostrado abaixo (notação UML padrão)**

I. Um objeto livro pode ser associado a mais de um objeto leitor

II. Um objeto leitor está associado a no máximo um único objeto livro

III. Nenhum objeto empréstimo está associado a uma associação entre livro e leitores



**As assertivas corretas são:**

A ( ) Somente o item I

B ( ) Somente o item II

C ( ) Somente o item III

**D (X) Somente os itens I e II**

E ( ) Itens I, II e III