



UNIVERSIDADE
VILA VELHA
ESPÍRITO SANTO

Curso de Ciência da Computação		
Disciplina: Programação Orientada a Objetos I	Nota: 4,5	Rubrica Coordenador
Professor: Cássio Capucho Peçanha		
Aluno: Lucas Carrijo Ferrari		
Turma: CC2M	Semestre: 1º	Valor: 7,0 pontos
Data: 14/04/2023	Avaliação: Bimestral	

INSTRUÇÕES DA PROVA

- Leia atentamente as questões antes de respondê-las;
- Todas as questões deverão ser respondidas com CANETA azul ou preta;
- Prova a lápis não tem direito à revisão;
- As questões objetivas rasuradas serão consideradas nulas;
- Desligue o celular, não consulte material, colegas ou fontes de qualquer outra natureza. Evite que sua prova seja recolhida pelo professor por atitudes indevidas.
- PROVA SEM CONSULTA E INDIVIDUAL.

objeto

1ª. questão. (0,5 ponto). (AOCP - 2021) Considerando o paradigma orientado a objetos, o que é uma Instância de Classe?

- a) Uma abstração de um objeto do mundo real. ✗
- ☒ b) Um modelo que representa um objeto. ✓
- c) Uma variável de tipo primitivo da classe.
- d) Um endereço de memória relacionado às definições da classe.
- e) Uma ocorrência específica de uma classe.

X OP

2ª. questão. (0,5 ponto). (CCV - 2013) Com relação aos conceitos de programação orientada a objetos, um objeto é:

- ☒ a) uma instância de uma classe.
- b) usado para instanciar classes ou outros objetos.
- c) uma variável de um tipo primitivo ou uma referência.
- d) uma referência para uma classe ou para outros objetos.
- e) um projeto (molde) para criar representações concretas de uma abstração.

✓ 0,5

3ª. questão. (0,5 ponto). Dados os seguintes trechos de código, sobre a relação de Cliente e Conta podemos afirmar que:

```
public class Cliente{
    public String nome;
    public String cpf;
    public String endereco;
}
/*****/
public class Conta{
    private double saldo;
    public Cliente dono;

    public Conta(double saldo_criacao){
        this.saldo = saldo_criacao;
    }
    public double getSaldo() { return this.saldo; }

    public void Sacar(double valor_saque){
        this.saldo = this.saldo - valor_saque;
    }
}
/*****/
public class Main{
    public static void main(String[] args) {
        Cliente c = new Cliente();
        c.nome = "João Silva";
        c.cpf = "123.456.789-00";
        c.endereco = "Rua Teste , S/N";

        Conta conta = new Conta(1000);
        conta.dono = c;
    }
}
```

É correto afirmar que:

- a) A relação de Cliente e Conta pode ser definida como Associação Simples
- b) A relação Cliente e Conta pode ser definida como Composição
- ☒ c) A relação Cliente e Conta pode ser definida como Agregação
- d) Cliente e Conta não possuem relação direta
- e) Um cliente não pode existir sem uma conta X

4ª. questão. (1,5). (SEGEF – 2023) Programação orientada a objetos é um paradigma de programação baseado no conceito de objetos. Considerando as estruturas utilizadas na programação orientada a objetos, relacione adequadamente as colunas a seguir.

1. Classe. 2. Objeto. 3. Método. 4. Atributo.

(3) Define-se dentro de uma classe para descrever o comportamento de um objeto. Programadores podem reutilizar ou manter a funcionalidade encapsulada dentro de um objeto.

(f) Tipo de dados definido pelo usuário que atua como um modelo para objetos, atributos e métodos individuais

(M) Define-se na classe e representa o estado de um objeto; pertence à própria classe.

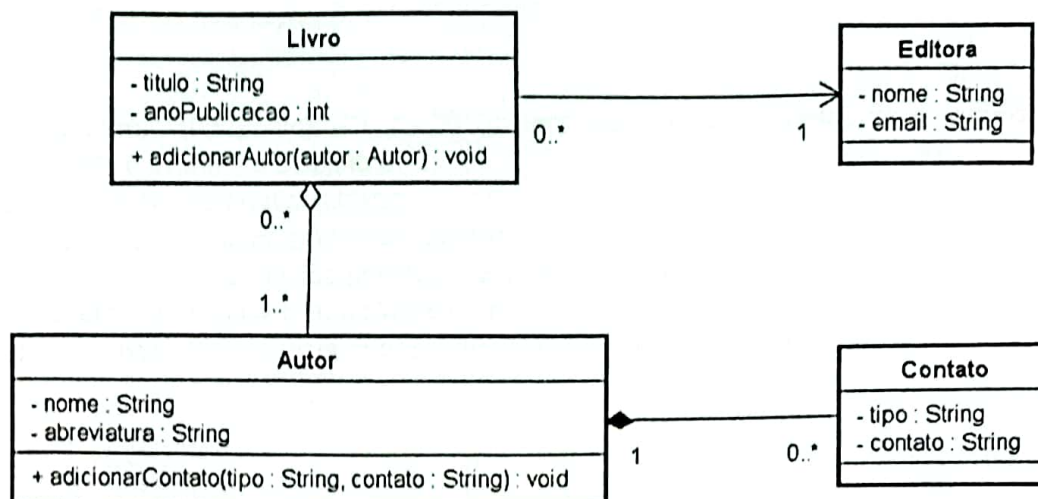
(2) Instância de uma classe criada com dados definidos; pode corresponder a objetos do mundo real ou a uma entidade abstrata.

A sequência está correta em

- a) 1, 3, 2, 4.
- b) 4, 2, 1, 3.
- c) 2, 4, 3, 1.
- ☒ d) 3, 1, 4, 2.
- e) 4, 1, 2, 3.

1,5

5ª. questão. (4,0). Utilizando tudo que aprendeu em nossas aulas com desenvolvimento em Java, realize a codificação do sistema representado pela modelagem abaixo. Em seu código, realize a inclusão de pelo menos 2 novos registros para cada classe. Não esqueça de demonstrar todas as associações.



2,8

Boa prova!

Aluno: Lucas Larijo Ferrari

→ Contato.java

```
public class Contato {  
    private String tipo;  
    private String contato;  
  
    public Contato (String tip, String cont) {  
        this.tipo = tip;  
        this.cont = cont;  
    }  
}
```

→ Editora.java

```
public class Editora {  
    private String nome;  
    private String email;  
  
    public Editora (String no, String em) {  
        this.nome = no;  
        this.email = em;  
    }  
}
```

→ Livro.java

```
public class Livro {  
    private String titulo;  
    private int anoPublicacao;  
    private Autor nomeAutor;  
    public Livro (String ti, int anoP){  
        this.titulo = ti;  
        this.anoPublicacao = anoP;  
    }  
  
    public void adicionarAutor (Autor autor){  
        this this.nomeAutor = autor;  
    }  
}
```

→ Autor.java

```
public class Autor {  
    private String nome;  
    private String abreviatura;  
    private Contato tipo;  
    private Contato contato; ↓ classe  
  
    public Autor (String no, String ab, Contato tip, Contato cont){  
        this.nome = no;  
        this.abreviatura = ab;  
        this.tipo = tip;  
        this.contato = cont;  
    }  
}
```



```

public void adicionar Contato (String tip , String cont ) {
    this.tipo = tip ;
    this.contato = cont ; }
}

```

→ Main . java

```

public class Main {
    public static void main (String [] args) {
        Editora ed1 = new Editora ("Árvore", "arvore@contato.com");
        Editora ed2 = new Editora ("Cachoeira", "cachoeira@contato.com");

        Livro livro1 = new Livro ("Percos Longos", "2020");
        Livro livro2 = new Livro ("Bhaskara", "1992");

        Autor autor1 = new Autor ("João Fulano", "JF", "Telefone", "27 99661000");
        Autor autor2 = new Autor ("Marina Fulana", "MF", "Fax", "+55 000000");

    }
}

```