



**INSTITUTO FEDERAL**  
Ceará  
Campus Fortaleza

Avaliação 1

Disciplina: Prog. Orientada à Objetos

Depto/Curso: Telemática/Eng. de Computação

Prof. Lucas Sousa

Semestre: 2023.2

Nota: 9.1 + 0.9 = 10.0

Nome: Kevin de Lima Rodrigues

Matrícula: \_\_\_\_\_

### Instruções:

- Todos os programas devem ser feitos utilizando a linguagem JAVA, como estudado em sala.
- Lembre-se das regras de indentação dos códigos. Sempre que abrir um bloco "{", todos os comandos internos ao bloco devem ser tabulados.
- A regra de indentação também vale para blocos if/else sem abertura explícita de blocos "{}". Nesses casos, o bloco deve conter somente um comando e ele deve ser também tabulado.
- Utilize materiais legíveis para escrever suas respostas. Porém, em caso do uso de lápis, não haverá protesto da correção.

**Questão 1.** Defina e diferencie **classes** e **objetos** no contexto de programação orientada à objetos. (1.42) 1.42

**Questão 2.** Defina o que é encapsulamento e como podemos criar classes encapsuladas em JAVA. Além disso, quais os modificadores de acesso e qual a função de cada um deles? (1.42) 1.42

**Questão 3.** Faça um programa em JAVA que leia N números e imprima o maior, o menor e a média entre eles. (1.42) 1.42

**Questão 4.** Um cartão de crédito possui um nome de titular, número, código de segurança e um limite. Com o cartão de crédito o usuário pode realizar compras desde que o total comprado não exceda o limite. O valor de todas as compras realizadas são acumuladas no cartão até que seja realizado o pagamento, que libera uma certa quantidade do limite do cartão (abate compras realizadas). Crie uma classe encapsulada em JAVA que descreva o cartão de crédito com seus atributos e os métodos `realizaCompras(float valor)` e o método `realizaPagamento(float valor)`. Crie os métodos e atributos extras que acharem necessários para o funcionamento do cartão de crédito. (1.42) 1.42

**Questão 5.** Modifique a classe criada na questão anterior de modo que seja possível realizar um aumento do limite de compras do cartão. (1.42) 1.42

**Questão 6.** Sobrecarregue o método `realizaPagamento` de modo que o valor total em compras seja pago e todo o limite seja liberado para compras novamente. (1.42) 0.9

**Questão 7.** Crie uma classe em JAVA que testa a operação do cartão de crédito. (1.42) 1.42

Boa prova!



Na orientação a objetos, classe é um componente que reúne as propriedades (atributos) e métodos para um grupo de objetos definindo uma espécie de "molde" para os objetos que serão criados a partir dela, e um objeto é a representação concreta do modelo definido pela classe, podendo atribuir valores para seus atributos, bem como chamar seus métodos.

2) Encapsulamento consiste em isolar componentes do código (classes, atributos ou métodos), separando-os dos demais para evitar que outros componentes tenham acesso livre a eles, garantindo uma maior segurança no projeto. No Java podemos criar classes encapsuladas através da utilização dos modificadores de acesso que irão definir seu nível de encapsulamento.

Java possui os modificadores: public, que pode ser usado em classes, métodos e atributos e permite seu acesso a partir de qualquer classe. private, que pode ser usado em classes internas, atributos e métodos e permite o acesso a eles apenas na classe que o define. protected, assim como o private, pode ser usado em classes, atributos e métodos, e permite o acesso a partir de classes internas.

do mesmo pacote por herança ou instâncias, e de pacotes distintos apenas por herança, e o default ou no-modifier, onde nenhum modificador é definido, pode ser usado em atributos, métodos e classes e seu acesso é permitido através de classes do mesmo pacote.



3)

```

public class Beste {
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        int soma = 0;
        int n;
        int[] v = {};
        int aux;
        n = sc.nextInt();

        for (int i = 0; i < n; i++) {
            v[i] = sc.nextInt();
            soma += v[i];
        }

        sc.close();

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < i; j++) {
                if (v[i] < v[j]) {
                    aux = v[i];
                    v[i] = v[j];
                    v[j] = aux;
                }
            }
        }

        double media;
        media = (double) (soma / n);

        System.out.println("Maior elemento: " + v[n-1]);
        System.out.println("Menor elemento: " + v[0]);
        System.out.println("Média: " + media);
    }
}

```



```

public class Cartao {
    private String nomeTitular;
    private String numCartao;
    private int codigoSeguranca;
    private float limite;

    public void setNomeTitular(String nomeTitular) {
        this.nomeTitular = nomeTitular;
    }

    public String getNomeTitular() {
        return this.nomeTitular;
    }

    public void setNumCartao(String numCartao) {
        this.numCartao = numCartao;
    }

    public String getNumCartao() {
        return this.numCartao;
    }

    public void setCodigoSeguranca(int codigoSeguranca) {
        this.codigoSeguranca = codigoSeguranca;
    }

    public int getCodigoSeguranca() {
        return this.codigoSeguranca;
    }

    public void setLimite(float limite) {
        this.limite = limite;
    }

    public float getLimite() {
        return this.limite;
    }

    public String toString() {
        String str = "";
        str += "Nome do Titular: " + this.nomeTitular;
        str += "\nNúmero do Cartão: " + this.numCartao;
        str += "\nCód. de Segurança: " + this.codigoSeguranca;
        str += "\nLimite: " + this.limite;
        return str;
    }
}

```



```

public void realizaCompra (float valor) {
    if (valor <= this.limite) {
        this.limite -= valor;
        System.out.println("Compra realizada!");
    } else {
        System.out.println("Compra recusada!");
        System.out.println("Valor superior ao seu limite");
    }
}

```

```

public void realizaPagamento (float valor) {
    if (valor <= this.limite) {
        System.out.println("Pagamento realizado");
        this.limite -= valor;
    } else {
        System.out.println("Pagamento recusado");
        System.out.println("Valor superior ao seu limite");
    }
}

```

→ perde o valor de ref. do limite do cartão.

```

5) public float aumentarLimite (float aumento) {
    this.limite += aumento;
    return this.limite;
}

```



```
public void realizaPagamento(float limite, float valor){
```

```
    if (valor <= limite){
```

```
        this.limite -= valor;
```

```
        System.out.println("Limite reestabelecido");
```

```
        System.out.println("Pagamento realizado");
```

```
    }
```

```
}
```

+ O usuário  
diz o limite?



```

7) public class BesteCartas {
    public static void main(String[] args) {
        Cartas c1 = new Cartas();
        c1.setNomeTitular("Kelvin Rodriguez");
        c1.setNumCartas("0000 1111 2222 3333");
        c1.setCodigoSeguranca(123);
        c1.setLimite(1350.00f);

        c1.realizaCompra(770.35f);
        c1.realizaPagamento(337.42f);
        c1.aumentarLimite(850.00f);
        c1.realizaPagamento(c1.getLimite, 875.84f);
    }
}

```