



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AP1 2º semestre de 2016.

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) (4.0 pontos)

Considere dois números inteiros $m, n > 0$, que representam, respectivamente, a quantidade de linhas e colunas de uma matriz de S de Strings, cujos elementos s_{ij} são definidos pela lei de formação:

$$s_{ij} = \begin{cases} =, & \text{se } i=j \\ <, & \text{se } i < j \\ i+j, & \text{se } i > j \end{cases}$$

Observe que, na terceira condição da lei de formação, o resultado dever ser armazenado como uma String. Por exemplo, se o número de linhas m for igual a 3 e o número de colunas n igual a 2 a matriz S será:

$$S = \begin{bmatrix} = & < \\ 1 & = \\ 2 & 3 \end{bmatrix}$$

Escreva um programa em Java, que receba $m, n > 0$ como parâmetros de entrada, composto pelo método principal e quatro métodos auxiliares, de acordo com as instruções dos itens abaixo:

- a)** O método principal deve receber do usuário os valores de m e n e invocar, em sequência, os métodos definidos nos itens **(b)** e **(c)**. **(1.0 ponto)**

- b) Escreva um método que receba os valores de m e n , construa e retorne a matriz S utilizando um Array bidimensional. A assinatura do método deve ser: **String[][] constroiMatriz (int m, int n).** (1.5 pontos)
- c) Escreva um método que receba uma matriz S , calcule e imprima na saída padrão a quantidade de entradas contendo strings numéricas que compõem a matriz. A assinatura do método deve ser **void numéricos (String[][] s).** (1.5 pontos)

RESPOSTA:

```
public class AP1_Q1_2016_2 {
    public static void main(String[] args){
        int m = Integer.parseInt(args[0]);
        int n = Integer.parseInt(args[1]);
        String[][] s = constroiMatriz(m, n);
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                System.out.print(s[i][j] + " ");
            }
            System.out.println();
        }
        numéricos(s);
    }

    private static String[][] constroiMatriz(int m, int n) {
        String[][] s = new String[m][n];
        for(int i = 0; i < m; ++i)
            for(int j = 0; j < n; ++j)
                if(i == j) s[i][j] = "=";
                else if(i < j) s[i][j] = "<";
                else s[i][j] = Integer.toString(i + j);
        return s;
    }

    private static boolean ehNumero(String s) {
        if ((s == null) || (s.length() == 0)) return false;
        for (int i = (s.charAt(0) == '+' || s.charAt(0) == '-') ?
1 : 0; i < s.length(); ++i)
            if ((s.charAt(i) < '0') || ('9' < s.charAt(i)))
                return false;
        return true;
    }

    private static void numéricos(String[][] s) {
        int cont = 0;
        for(int i = 0; i < s.length; ++i)
            for(int j = 0; j < s[i].length; ++j)
                if(ehNumero(s[i][j])) cont++;
        System.out.println(cont);
    }
}
```

Questão 2) (3.0 pontos)

Considere a classe abaixo que modela produtos de forma geral:

```
class Produto {
    private int codigo;
```

```

private String nome;
private double preco;
public Produto(int codigo, String nome, double preco) {
    this.codigo = codigo;
    this.nome = nome;
    this.preco = preco;
}
public int getCodigo() {
    return codigo;
}
public void setCodigo(int codigo) {
    this.codigo = codigo;
}
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public double getPreco() {
    return preco;
}
public void setPreco(double preco) {
    this.preco = preco;
}
}

```

Crie uma classe chamada CarrinhoDeCompras, a qual permite a manipulação de um conjunto de produtos. Como sugestão, esse conjunto pode ser representado como um vetor. Crie nesta classe um método chamado adiciona, o qual insere produtos num carrinho. Crie também um método chamado soma, o qual retorna a soma de todos os preços dos produtos do carrinho. Para testar a classe e os métodos criados, crie uma classe Teste, a qual cria um objeto do tipo carrinho e adiciona os produtos abaixo:

Código	Nome	Preço
1	TV Plasma	1500
2	Geladeira Frost Free	1000
3	Sofa 3 lugares	500

RESPOSTA:

```

class CarrinhoDeCompras {
    Produto produtos[];
    int ultimo = 0;

    public CarrinhoDeCompras() {
        produtos = new Produto[1000];
    }

    public void adiciona (Produto p) {
        produtos[ultimo] = p;
        ultimo++;
    }
}

```

```

        public double soma () {
            double s = 0;
            for (int i=0; i<ultimo; i++)
                s = s + produtos[i].getPreco();
            return s;
        }
    }

    public class Teste {
        public static void main(String[] args) {
            Produto tv = new Produto(1, "TV Plasma", 1500);
            Produto geladeira = new Produto(2, "Geladeira Frost Free", 1000);
            Produto sofa = new Produto(3, "Sofa 3 lugares", 500);
            CarrinhoDeCompras carrinho = new CarrinhoDeCompras();
            carrinho.adiciona(tv);
            carrinho.adiciona(geladeira);
            carrinho.adiciona(sofa);
            System.out.println("Valor do carrinho: " + carrinho.soma());
        }
    }
}

```

Questão 3) (3.0 pontos)

Defina uma classe de objetos que representam uma data (dia, mês e ano), contendo os seguintes métodos:

- um método toString, que retorna uma cadeia de caracteres correspondente à data representada pelo objeto alvo da chamada, no formato “dd/mm/aaaa”, onde dd, mm e aaaa correspondem, respectivamente, ao dia, mês e ano desta data;
- um método compara, que compara a data representada pelo objeto alvo da chamada com uma data passada como argumento para o método; o valor retornado deve ser 0 se essas datas são iguais, ou um número negativo se a primeira data é anterior à última, ou um número positivo se a primeira é posterior à última.

RESPOSTA:

```

class Data {
    int dia;
    int mes;
    int ano;

    public Data(int dia, int mes, int ano) {
        super();
        this.dia = dia;
        this.mes = mes;
        this.ano = ano;
    }

    public String toString() {
        return dia + "/" + mes + "/" + ano;
    }

    public int compara (Data d) {
        if (dia == d.getDia() && mes == d.getMes() && ano == d.getAno())
            return 0;
    }
}

```

```
        if (ano < d.getAno() ||
            (ano == d.getAno() && mes < d.getMes()) ||
            (ano == d.getAno() && mes == d.getMes() && dia <
d.getDia()))
            return -1;
        return 1;
    }

    public int getDia() {
        return dia;
    }

    public int getMes() {
        return mes;
    }

    public int getAno() {
        return ano;
    }
}
```