



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AP3 2º semestre de 2018.

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) (5.0 pontos)

Suponha o código abaixo:

```
import java.util.ArrayList;
import java.util.List;

public class AP3_2018_2_Q1 {
    public static void main(String[] args) {
        //Nome e url
        EnderecoWWW e = new EnderecoWWW("Cederj", "www.cederj.edu.br");
        //Nome, url e tamanho da imagem
        Imagem i = new Imagem("Cederj", "http://cederj.edu.br/fundacao/wp-content/
uploads/2014/03/marca_fundacao_cecierj_consorcio_cederj.png", 25);
        List<Recurso> pagina = new ArrayList<Recurso>();
        pagina.add(e);
        pagina.add(i);
        for (Recurso r : pagina)
            System.out.println("O recurso " + r.toString() + " eh " +
r.valido());
    }
}
```

Implemente as classes/interfaces necessárias para que o código acima funcione. A chamada ao método *valido()* retorna verdadeiro quando o recurso se tratar de uma imagem e terminar com *jpg* ou *png*, ou quando o recurso for um endereço *www* e iniciar com a string *www* (Dica: utilize os métodos de instância *startsWith* e *endsWith* da classe *String*). O método *toString()* deve retornar a url respectiva de cada recurso. Utilize os

conceitos de OO vistos sempre que possível para, por exemplo, evitar redundância no código apresentado.

RESPOSTA:

```
import java.util.ArrayList;
import java.util.List;

abstract class Recurso {
    String nome;
    String url;

    public Recurso (String nome, String url) {
        this.nome = nome;
        this.url = url;
    }

    public String toString() {
        return url;
    }

    public abstract boolean valido();
}

class EnderecoWWW extends Recurso {
    public EnderecoWWW (String nome, String url) {
        super(nome, url);
    }

    public boolean valido() {
        return this.url.startsWith("www");
    }
}

class Imagem extends Recurso {
    int tamanho;
    public Imagem (String nome, String url, int tamanho) {
        super(nome, url);
        this.tamanho = tamanho;
    }

    public boolean valido() {
        return this.url.endsWith(".jpg") || this.url.endsWith(".png");
    }
}
```

Questão 2) (5.0 pontos)

Considere que sua empresa seja contratada para avaliar competições de programação nacionais. A maioria dos alunos que participam dessas competições com exercícios de programação concordam em quatro características que toda competição deve alcançar. Embora nem todas sejam sempre alcançadas, quanto mais melhor. As características são as seguintes:

1. Ninguém resolveu todos os problemas.

2. Todo problema foi resolvido por pelo menos uma pessoa (não necessariamente a mesma).
3. Não há nenhum problema resolvido por todos.
4. Todos resolveram ao menos um problema (não necessariamente o mesmo).

Dadas as informações sobre cada competição, como o número de participantes, o número de problemas, e qual participante resolveu quais problemas, descubra o número de características que foram alcançados nesta competição.

Seu programa deve receber um arquivo com um resumo de várias competições. Cada competição inicia com dois inteiros N e M ($3 \leq N$, $M \leq 100$), indicando, respectivamente, o número de participantes e o número de problemas. Em seguida, haverá N linhas com M inteiros cada, onde o inteiro da linha i e coluna j é 1 caso o competidor i resolveu o problema j , ou 0 caso contrário. A última competição é indicada quando $N = M = 0$, o qual não deverá ser processado.

Para cada competição, imprima uma linha contendo um inteiro, representando quantas das características citadas foram alcançadas na competição.

Um exemplo de execução desse programa seria:

Arquivo de entrada: entrada.txt	Resposta na tela:
3 3	2
1 1 0	
0 1 0	
0 0 0	
3 3	4
1 1 0	
0 1 0	
0 0 1	
0 0	FIM

LEMBRE-SE: SEU PROGRAMA SÓ PODE LER O ARQUIVO DE ENTRADA UMA ÚNICA VEZ, E ELE DEVE EXECUTAR COM QUAISQUER DADOS INFORMADOS COMO PARÂMETROS DE ENTRADA. SE O SEU PROGRAMA RESOLVER SOMENTE O EXEMPLO SUPRACITADO, SUA QUESTÃO SERÁ TOTALMENTE DESCONTADA.

RESPOSTA:

```
import java.io.*;
import java.util.*;

public class Q2_AP3_POO_2018_2{
    public static void main(String[] args) throws IOException{
        BufferedReader in;
        in = new BufferedReader(new FileReader(args[0]));
        int n, m;
```

```

try {
    String s = in.readLine(), vs[];
    while(s != null){
        vs = s.split(" ");
        n = Integer.parseInt(vs[0]);
        m = Integer.parseInt(vs[1]);
        if((n == 0) && (m == 0)){
            System.out.println ("FIM");
            in.close();
            return;
        }
        if((n == 0) || (m == 0)) continue;
        if((n < 2) || (m < 2) || (n > 100) || (m > 100)) continue;

        int mat[][] = new int[n][m], i, j;
        for(i = 0; i < n; i++){
            s = in.readLine();
            vs = s.split(" ");
            for(j = 0; j < m; j++)
                mat[i][j] = Integer.parseInt(vs[j]);
        }

        int aux, carac1, carac2, carac3, carac4;
        carac1 = carac2 = carac3 = carac4 = 0;
        //carac1: Ninguém resolveu todos os problemas
        for(i = 0; i < n; i++){
            aux = 0;
            for(j = 0; j < m; j++) aux += mat[i][j];
            if(aux == m){
                carac1 = 0;
                break;
            }
        }
        if(i == n) carac1 = 1;

        //carac2: Todo problema foi resolvido por pelo menos uma
        //pessoa (não necessariamente a mesma)
        for(j = 0; j < m; j++){
            aux = 0;
            for(i = 0; i < n; i++) aux += mat[i][j];
            if(aux == 0){
                carac2 = 0;
                break;
            }
        }
        if(j == m) carac2 = 1;

        //carac3: Não há nenhum problema resolvido por todos
        for(j = 0; j < m; j++){
            aux = 0;
            for(i = 0; i < n; i++) aux += mat[i][j];
            if(aux == n){
                carac3 = 0;
                break;
            }
        }
        if(j == m) carac3 = 1;
    }
}

```

```

        //carac4: Todos resolveram ao menos um problema
        //(não necessariamente o mesmo)
        for(i = 0; i < n; i++){
            aux = 0;
            for(j = 0; j < m; j++) aux += mat[i][j];
            if(aux == 0){
                carac4 = 0;
                break;
            }
        }
        if(i == n) carac4 = 1;

        System.out.println(carac1 + carac2 + carac3 + carac4);
        s = in.readLine();
    }
}
catch (Exception e){
    System.out.println("Excecao\n");
    in.close();
}
}
}

```