



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AP3 1º semestre de 2017.

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)

Um senhor está trabalhando em uma mina, tentando retirar o máximo que consegue de diamantes "<>". Ele deve excluir todas as partículas de areia (" . ") existentes do processo e, a cada retirada de diamante, novos diamantes poderão se formar. Se ele tem como uma entrada ".<...<<...>>...>...>>>.", três diamantes são formados. O primeiro é retirado de "<...>", resultando ".<...<>...>...>>>." Em seguida o segundo diamante é retirado, restando ".<.....>...>>>." O terceiro diamante é então retirado, restando no final ".>>>.", sem possibilidade de extração de novo diamante.

Outros exemplos de entrada e saída são:

→ "< . . > < . . >>" que contém 3 diamantes; e

→ "<<< . . < <<<< >" que contém 1 diamante.

Nesta questão você deve:

(a)(2.0 pontos) Implementar uma pilha de char.

(b)(3.0 pontos) Utilizar a pilha implementada no item (a) para resolver o problema de extração de diamantes. Você deve receber, como argumento do método **Main**, o nome do arquivo de entrada que informa o formato da mina, e implementar o método `int garimparDiamantes(String nome_arquivo_entrada)`, que retorna a quantidade de diamantes garimpados.

RESPOSTA:

```
import java.io.*;

//LETRA (a)
class Pilha {
    private char[] elems = new char[0];
    private int topo = -1;

    public void empilhar(char c) {
        this.topo++;
        if (this.topo == this.elems.length) {
            char[] novo = new char[this.elems.length + 1];
            System.arraycopy(this.elems, 0, novo, 0, this.elems.length);
            this.elems = novo;
        }
        this.elems[this.topo] = c;
    }

    public void desempilhar() {
        if (this.topo == -1) throw new RuntimeException("Pilha vazia.");
        this.topo--;
    }

    public char topo() {
        if (this.topo == -1) throw new RuntimeException("Pilha vazia.");
        return this.elems[this.topo];
    }

    public boolean estaVazia() { return this.topo == -1; }
}

//LETRA (b)
public class Q1_AP3_2017_1 {
    public static void main(String[] args) throws IOException{
        System.out.println(args[0] + ": " + garimparDiamantes(args[0]));
    }

    public static int garimparDiamantes(String arquivo) throws
    IOException {
        if (arquivo == null)
            throw new IllegalArgumentException("Entrada invalida.");
        BufferedReader in;
        in = new BufferedReader(new FileReader(new File(arquivo)));
        int qtd = 0;
        try{
            String s = in.readLine();
            Pilha p = new Pilha();

            for (int i = 0; i < s.length(); i++) {
                switch (s.charAt(i)) {
                    case '.':
                        // Não fazer nada.
                        break;
                    case '<':
                        p.empilhar('<');
                        break;
                    case '>':
                        if (!p.estaVazia() && (p.topo() == '<')) {
```

```

        p.desempilhar();
        qtd++;
    }
    else p.empilhar('>');
    break;
default:
    throw new IllegalArgumentException("Entrada inválida.");
}
}
}
}
catch (Exception e) { System.out.println("Excecao\n"); }
finally { in.close(); }
return qtd;
}
}
}

```

Questão 2) (5.0 pontos)

Suponha o código abaixo:

```

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

public class AP3_2017_1_Q2 {
    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_consortio_cederj.png", 25);
        List<Recurso> pagina = new ArrayList<Recurso>();
        pagina.add(e);
        pagina.add(i);
        for (Recurso r : pagina)
            System.out.println("0 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;

interface IRecurso { // Opcional
    boolean valido();
}

```

```
abstract class Recurso implements IRecurso {
    String nome;
    String url;

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

    public String toString() {
        return url;
    }
}

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");
    }
}
```