Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação III
AP3 2° semestre de 2012.

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)

Escreva um programa que receba, como parâmetro de entrada, um arquivo texto e, após SOMENTE UMA LEITURA neste arquivo de entrada, seu programa deve ser capaz de fazer um resumo deste arquivo, mostrando todas as palavras que estão contidas nele, quantas vezes cada palavra aparece e em que linhas elas aparecem, ignorando diferenças entre letras maiúsculas e minúsculas. Observe que números e pontuações NÂO são consideradas palavras.

O resumo deve ser gravado no arquivo **"output-"** acrescida do nome do arquivo de entrada. Para que seu programa funcione corretamente, pontuações (ponto, vírgula, entre outros) devem ser separadas das palavras por espaço em branco. Por exemplo, se o arquivo de entrada, **exemplo.txt**, fosse composto por (OBSERVE QUE SEU PROGRAMA DEVE FUNCIONAR PARA QUALQUER ARQUIVO TEXTO QUE SEGUE O PADRÃO DE ENTRADA):

```
Meu arquivo tem várias palavras . Palavras não faltam . Várias palavras
repetidas . Arquivo que eu criei , meu arquivo .
```

A execução deste programa (**java AP3_Q1_2012_2 exemplo.txt**) geraria o arquivo **saida-exemplo.txt** com o seguinte formato:

```
pal         cont        linhas
Meu         2           1 2
arquivo     3           1 2
tem         1           1
várias      2           1 2
```

```
palavras    3           1
não         1           1
faltam      1           1
repetidas   1           2
que         1           2
eu          1           2
criei       1           2
```

RESPOSTA:

```
import java.io.*;

class no{
  int num;
  no prox;

  no(int n){
    num = n;
    prox = null;
  }

  public String toString(){ return num + " "; }
}


class lista{
  String palavra;
  int n;
  no linha;
  lista prox_pal;

  lista(String p, int l){
    palavra = p;
    n = 1;
    linha = new no(l);
    prox_pal = null;
  }

  //busca: retorna um nó de lista se a palavra já está na lista.
  lista busca(String p){
    lista q = this;
    while(q != null){
      if(p.equalsIgnoreCase(q.palavra)) return q;
      q = q.prox_pal;
    }
    return q;
  }

  //insere_Linha: no nó de lista, inserir a linha em que ocorre a
  //palavra.
  void insere_Linha(int l){
    no ant = null, p = linha;
    while((p != null) && (p.num != l)){
      ant = p; p = p.prox;
    }
```

```
    //a linha não está na lista. É necessário incluir a linha.
    //se a linha já está citada, não faça nada.
    if(p == null) ant.prox = new no(l);
  }

  //insere_fim: insere palavra na última posição da lista.
  void insere_fim(String p, int l){
    lista q = this;
    while(q.prox_pal != null) q = q.prox_pal;
    q.prox_pal = new lista(p, l);
  }

  //insere: verifica se a palavra já está na lista. Se estiver,
  //só atualiza o número de linhas que ela aparece. Senão, insere
  //no final da lista de palavras.
  void insere(String p, int l){
    lista q = busca(p);
    if(q != null){
      q.insere_Linha(l);
      q.n++;
    }
    else insere_fim(p, l);
  }

  //toString: imprime no formato do arquivo de saída.
  public String toString(){
    String resp = "pal\tcont\tlinhas\n";
    lista p = this;
    while(p != null){
      resp = resp + p.palavra + "\t" + p.n + "\t";
      no q = p.linha;
      while(q != null){
        resp += q.toString();
        q = q.prox;
      }
      resp += "\n";
      p = p.prox_pal;
    }
    return resp;
  }
}


public class Q1_AP3_2012_2{
  public static void main(String[] args) throws IOException{
    BufferedReader in = new BufferedReader(new FileReader(args[0]));
    int n = 0;
    String s;
    lista l = null;
    try{
      String[] vs;
      while ((s = in.readLine()) != null){
        n++;
        vs = s.split(" ");
```

```
        for(int i = 0; i < vs.length; i++){
          if(!vs[i].equals(".") && !vs[i].equals(",")){
            if(l == null) l = new lista(vs[i], n);
            else l.insere(vs[i], n);
          }
        }
      }
      in.close();
      BufferedWriter out;
      out = new BufferedWriter(new FileWriter("saida-" + args[0]));
      out.write(l.toString());
      out.close();
    }catch (Exception e) { System.out.println("Excecao\n"); }
  }
}
```

Questão 2) (5.0 pontos)

Suponha o código abaixo:

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

public class AP3_2012_2_Q3 {
	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://www.cederj.edu.br/fundacao/imagenssuperior_arquivos/menu_logo.jpg", 50);
		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*. 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.

RESPOSTA:

```
interface IRecurso {
	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");
	}
}
```