

Lista de exercícios 5

Programação Orientada a Objetos II

1) Implemente uma classe `MinhaArvore`, que modela a estrutura de dados de uma árvore. Essa classe deverá implementar a seguinte interface:

```
public interface Arvore<T> {  
    // remove todos nodos  
  
    public void limpa();  
  
    // procura se o valor existe em algum nodo e retorna uma referência para este nodo  
  
    public Object existe(T valor);  
  
    // adiciona raiz  
  
    public void criaRaiz(T valor);  
}
```

Objetos da classe `MinhaArvore` deverão guardar uma referência para a raiz da árvore, para que possam percorrer a árvore a qualquer momento. Além disso, crie uma outra classe `Nodo`, que guardará informações próprias de cada nodo, como o seu valor, uma referência para seu pai, e uma lista com todos seus filhos. Adicione métodos nessa classe `nodo` para que os métodos da classe `MinhaArvore` possam funcionar.

`MinhaArvore` terá um formato parecido com esse:

```
public class MinhaArvore<T> implements Arvore<T> {  
    Nodo raiz;  
  
    @Override  
    public Object existe(T valor) {  
        return null;  
    }  
  
    ...  
}
```

O formato da classe `Nodo` será parecido com esse:

```
public class Nodo<T> {  
    T valor;  
    // nodo atual adiciona como filho outro nodo, retorna referência para novo nodo  
    public Object adicionaFilho(T valor) {  
  
    }  
  
    // remove nodos filhos que tenham valor igual a valor  
    public boolean removeFilho(T valor) {  
  
    }  
    ...  
}
```

Na main, teste suas classes com objetos da classe String, simulando uma árvore genealógica.

2) Com as classes criadas em “1”, implemente um leitor de arquivos HTML que guarda na árvore todas as informações de uma página. Na main, implemente um teste para mostrar que o arquivo foi corretamente lido.