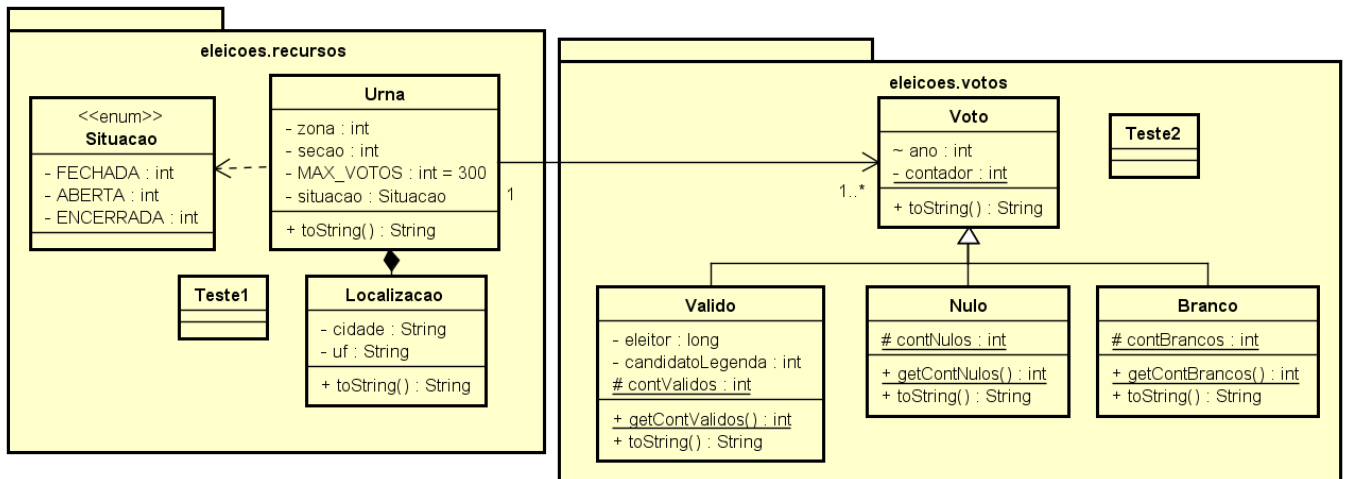


Disciplina: Programação para Web I - 3º Semestre	Nota:
Turma: Manhã	
Data: 16/10/2018	Professora: Silvia Bertagnolli

PROVA

Analise o diagrama abaixo para responder as questões 1 a 3 da prova.



powered by Astah

1) Responda os itens abaixo:

- Implemente as classes Urna, Localizacao, Voto e VotoValido, sabendo que um Urna tem vários votos e uma localização
- Use a classe LinkedList para determinar que uma urna tem vários votos
- Use super e this sempre que aplicável
- É necessário definir a classe Urna de modo que seus objetos possam ser ordenados pela sua cidade
- Determine qual linha de código você usaria para definir a classe Localizacao como final? Explique com suas palavras as vantagens de declarar uma classe como final.
- Determine qual linha de código você usaria para definir a classe Voto como abstrata?
- Todos os contadores devem ser declarados como variáveis de classe.
- O método getContador() na classe Voto deve ser declarado como abstrato? Explique sua resposta.
- Como você faria a declaração de MAX_VOTOS sabendo que ela é uma constante e deve ser compartilhado por todos os objetos da classe Urna?

2) Analise o código abaixo, identifique os erros e explique com suas palavras como corrigi-los:

```
public class Teste1 {  
    public static void main(String[] args) {  
        Voto v = new Voto();  
        v.ano = 2018;  
        v.contador = 12;  
        Nulo n = new Nulo();  
        n.ano = 2018;  
        n.contNulos = 10;  
    }  
}
```

```
-----  
public class Teste2 {  
    public static void main(String[] args) {  
        Voto v = new Voto();  
        v.ano = 2018;  
        v.contador = 12;  
        Nulo n = new Nulo();  
        n.ano = 2018;  
        n.contNulos = 10;  
    }  
}
```

3) Monte a classe de testes que:

- 1 – Cadastra Urna (com a situação fechada)
- 2 – Abre Urna (que modifica a situação para aberta)
- 3 – Votar (permite cadastrar um voto)
- 4 – Fechar Urna (que modifica a situação para fechada e mostra o número total de votos, o número total de votos válidos, votos nulos e votos brancos)

4) Defina a interface ConjuntoOrdenado como genérica. Ela deve conter os métodos abaixo:

```
public void adicionarElemento(Object obj)  
public String toString()  
public int totalizar()  
public void removerTodosElementos()  
public boolean contemElemento(Object obj)  
public boolean removerElemento(Object obj)
```

5) Crie a classe Conjunto que deve ser definida como genérica e utilizar a classe TreeSet. Para resolver essa questão você ainda deve usar a interface definida na questão anterior.

6) Agora, determine:

- 6.1 Qual código deveria ser incluído/excluído na classe anterior se o método adicionar propagasse a exceção NullPointerException?
- 6.2 Qual código deveria ser incluído/excluído na classe anterior se o método substituir causasse a exceção NullPointerException?

7) A linha de código Conjunto <Number> conjunto = new Conjunto<>(); funcionaria se fosse colocada em uma classe de testes. Justifique sua resposta.

8) Faça o tratamento das exceções do código abaixo usando try com recursos e catch, ou try, catch e finally.

```
public class Questao8 {  
    public static void main(String[] args) {  
        FileReader in = new FileReader("testeQ8.txt");  
        BufferedReader buff = new BufferedReader(in, 1024);  
  
        StringBuilder builder = new StringBuilder();  
        String s = null;  
        while ((s = buff.readLine()) != null) {  
            builder.append(s).append("\n");  
        }  
        System.out.println("Conteudo do arquivo:\n\n"+builder);  
        buff.close();  
        in.close();  
    }  
}
```