



Inicie cada grupo numa folha separada. Escreva o seu número de aluno em todas as folhas. Escreva na folha de rosto o número de folhas que utilizou. Duração: três horas.

Considere o seguinte esboço de uma classe que implementa uma lista ordenada, sem repetidos e com um número limitado de elementos.

```
1 public class BoundedSortedList<E extends Comparable<E>> {
2     private class Node {...} // a node in the list
3     // creates a list given its maximum capacity
4     public BoundedSortedList (int capacity) {...}
5     public E get (int i) {...} // gets the i-th element
6     public void remove (int i) {...} // removes the i-th element
7     public boolean isEmpty () {...} // is the list empty?
8     public boolean isFull () {...} // is the list full?
9     // adds a non-existing element v to the list, in ascending order
10    public boolean add (E v) {
11        if (isFull())
12            return false;
13
14        Node newNode = new Node ();
15        newNode.v = v;
16
17        Node previous = null;
18        Node current = head;
19
20        while (current != null && current.v.compareTo(v) <= 0) {
21            if (current.v.compareTo(v) == 0)
22                return false;
23            previous = current;
24            current = current.next;
25        }
26
27        newNode.next = current;
28        if (previous == null)
29            head = newNode;
30        else
31            previous.next = newNode;
32
33        size++;
34        return true;
35    }
36 }
```

Grupo 1. [4 valores]

- Desenhe o grafo de controlo de fluxo do método `add`.
- Apresente uma classe de testes (compatível com JUnit) que satisfaça a cobertura de arestas para o método `add`. Para cada teste indique o caminho de testes correspondente.
- Identifique os conjuntos definição (*def*) e utilização (*use*) para cada nó e aresta para o grafo encontrado em a). Liste todos os caminhos definição-utilização respeitantes às variáveis `v` e `previous`.

d) Indique um conjunto minimal de testes que satisfaça a cobertura de *todos os caminhos definição-utilização* para as variáveis `v` e `previous`. Será que este conjunto de testes também satisfaz os critérios *todas as definições* e *todas as utilizações*? Porquê?

Grupo 2. [4 valores]

- a) Identifique os predicados constantes no método `add`.
- b) Analise o problema da acessibilidade (*reachability*), indicando para cada um dos predicados da alínea anterior o seu predicado de acessibilidade expresso em função dos parâmetros e atributos da classe. Se não for possível, indique o porquê?
- c) Indique um conjunto de testes que satisfaça a cobertura de *cláusula activa restrita* para cada predicado do programa. Se tal não for possível, justifique a razão e indique como pode relaxar o critério de cobertura.
- d) Indique em que consiste a cobertura de cláusulas inactivas e porque razão não faz sentido existir uma cobertura *correlacionada de cláusulas inactivas*.

Grupo 3. [4 valores]

A actividade de teste está intimamente ligada à escolha de elementos particulares do espaço de entrada. A partição deste espaço em regiões permite escolher um elemento para testar cada uma das regiões.

- a) Identifique os parâmetros dos métodos e os estados abstractos da classe `BoundedSortedList` esboçada.
- b) Dados os estados abstractos e os parâmetros identificados em a), identifique cinco características que sugiram partições.
- c) Apresente os vários blocos em que se divide cada característica identificada acima.
- d) Escolha um representante para cada bloco, sugerindo valores para os parâmetros e estados da lista.
- e) Para cada característica, designe um dos seus blocos como *base*. Para duas destas características designe um segundo bloco base. Defina um conjunto de testes que satisfaça a cobertura de *escolha básica múltipla* (*multiple base choice coverage*) para o método em teste.

Grupo 4. [3 valores]

- a) De entre as seguintes categorias de mutantes: troca de operador relacional, troca de operador aritmético, troca de constante numérica, e troca de variável, sugira mutantes para as linhas de código números 20, 24 e 33.
- b) Para cada um destes mutantes apresente um teste que *não* alcance o mutante. Quando tal for impossível explique porquê.
- c) Para cada um dos mutantes descreva um teste que alcance o mutante mas que não provoque infecção. Quando tal for impossível explique porquê.
- d) Para cada um dos mutantes descreva um teste que provoque infecção mas que não a propague. Quando tal for impossível explique porquê.
- e) Para cada um dos mutantes descreva um teste que mate o mutante. Quando tal for impossível explique porquê.