

**Universidade de Aveiro**  
**Exame Teórico de Programação 3**

Nome: \_\_\_\_\_ N.º Mec. \_\_\_\_\_

**Parte A**

Relativamente às perguntas 1 a 20, assinale na tabela ao lado, um X na coluna “V” para as declarações que estão correctas e na “F” para as que estão incorrectas.

Cada uma destas perguntas vale 0,5 valor e cada resposta errada desconta 0,2 valores.

1. Uma classe pode implementar várias interfaces.
2. A declaração “final int var[] = {3, 6, 11};” significa que o conteúdo do array “var” não pode ser alterado.
3. Uma super classe “public class XPTO {}” só tem definido um único construtor privado. É possível criar uma classe derivada de XPTO.
4. Entre as entidades vaca, couro, bola deverá haver uma relação de herança.
5. As asserções permitem verificar invariantes.
6. A classe FileReader não apresenta nenhum método para ler uma linha de texto.
7. Podemos usar um único catch para capturar eventos gerados por múltiplos blocos try.
8. Podemos adicionar múltiplos JPanel a um JPanel existente.
9. É possível obter um Iterator de um Map.
10. Se x for derivada de y e G for uma classe genérica, então G<x> é derivada de G<y>.
11. A referência this pode ser usada num método estático.
12. Os membros private de uma classe só podem ser acedidos dentro da declaração dessa classe.
13. Um método final não pode ser redefinido numa classe derivada.
14. Uma classe abstracta pode ser derivada de uma outra classe abstracta.
15. Os construtores podem ser definidos como abstractos.
16. O termo throws é usado para gerar uma excepção.
17. Um método que gere uma excepção checked tem obrigatoriamente de ser invocado dentro de um bloco try .. catch.
18. Na invocação do método processa(final Pessoa p) o objecto referenciado por p não poderá ser modificado dentro do método processa.
19. Não é possível criar um array de um tipo genérico: T[] array = new T[MAX];
20. A expressão c ref1 = new B(); está correcta desde que todo o c seja um B.

	V	F
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

## Parte B

1. Considere a classe Forma. Descreva as razões e as consequências principais de definir esta classe como abstracta (Quando é que uma classe deve ser definida abstracta? Como se definem classe e métodos? Como se instancia? Como se herda? Qual a vantagem?). *Nota: pode apresentar exemplos em código para ajudar a explicar conceitos.*

2. A janela da figura tem 4 botões e uma caixa de texto. Quando se clica num dos botões, a caixa de texto fica com o valor do label do botão. Complete o código em falta.

```
public class Jogo2x2 extends _____ implements _____ {
    private JPanel panel1, panel2;
    private JButton b[] = new JButton[4];
    private String JLabels[] = {"1", "2", "3", "4"};
    private JTextField jtxt1;

    public Jogo2x2(String nome) {
        _____ (nome);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel1 = new JPanel();
        panel2 = new JPanel();
        _____ .setLayout(new GridLayout(2,2));
        _____ .setLayout(new BorderLayout());
        panel1. _____ (panel2, BorderLayout.CENTER);
        for (int i=0; i < b.length; i++) {
            b[i] = new JButton(JLabels[i]);
            _____ . _____ (b[i]);
            b[i].addActionListener(this);
        }
        jtxt1 = new JTextField(10);
        panel1.add(jtxt1, BorderLayout.SOUTH);
        panel1.setOpaque(true);
        setSize(400,400);
        setContentPane( _____ );
        setVisible(true);
    }

    public void _____ (ActionEvent e) {
        jtxt1.setText(((JButton) (e.getSource())).getText());
    }

    public static void main(String[] args) {
        new Jogo2x2( _____ );
    }
}
```



3. Converta a “Stack\_String” abaixo numa Stack do tipo Genérico:

```
public class Stack_String {  
  
    private Node top = null;  
  
    private class Node {  
        String val;  
        Node next;  
        Node(String v, Node n) {  
            val = v;  
            next = n;  
        }  
    }  
  
    public boolean empty() {  
        return top == null;  
    }  
  
    public String top() {  
        return top.val;  
    }  
  
    public String pop() {  
        String result = top.val;  
        top = top.next;  
        return result;  
    }  
  
    public void push(String v) {  
        top = new Node(v, top);  
    }  
}
```

4. Considere as entidades Musico, Saxofonista, Flautista, Guitarrista. Que tipo de padrão de software deverá considerar para que o código seguinte seja válido? Use um diagrama de classes em UML para apresentar a solução.

```
MusicoInterface m = new Musico("Luis");  
Flautista fl = new Flautista(m);  
Guitarrista gl = new Guitarrista( new Saxofonista( new Musico("John Paredes") ) );
```