



Universidade Eduardo Mondlane

Faculdade de Engenharia

Departamento de Engenharia Electrotécnica

Curso de Engenharia Informática

Teste 1 de Programação Orientada a Objectos II (11/09/2015)

Duração: 120 minutos

1. Qual a saída produzida pela execução do código abaixo? (2.5v)

```
1 public class MT implements Runnable
2 {
3     String ms = "S ";
4     public void run()
5     {
6         this.ms = "N ";
7     }
8     public static void main(String[] args)
9     {
10        MT t = new MT();
11        new Thread(t).start();
12        for (int i=0; i < 10; i++)
13            System.out.print(t.ms);
14    }
15 }
```

a) Não é possível determinar

b) S S S S S S S S S S ...

c) N N N N N N N N N N ...

d) S N S N S N S N S N ...

2. Suponha que esteja a desenvolver um sistema de inscrições online, em arquitectura de 3 camadas. Entretanto para um estudante possa inscrever-se em determinada cadeira, essa não pode ser precedente a uma cadeira por ele reprovada. Em que camada faria a validação? Justifique a sua resposta. (1.5v)

a) Camada de Apresentação por trás do botão de inscrição.

b) Camada de Negócio.

c) Camada de Acesso a Dados

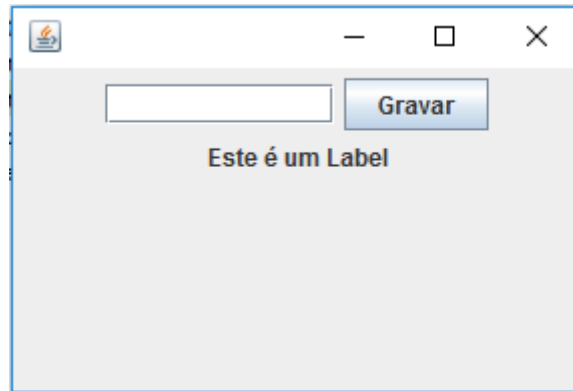
d) Na Base de dados como um Procedimento armazenado

3. Crie uma pequena janela com tamanho 300 x 200 que é um JFrame, que tenha um JTextField e um JLabel e um botão. Nessa mesma classe deve ser implementada uma Thread que ouve se já se passaram 5 minutos (Use Thread.sleep()) para facilitar a contagem do tempo) desde que o programa foi aberto, caso tenham se passado 5 minutos ele fecha o programa através do método **System.exit(0)**. (6v)

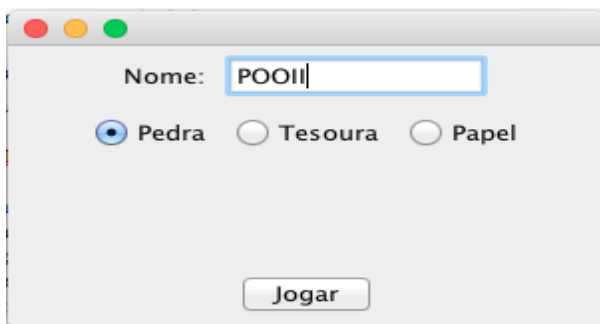
```
22 public class Exercicio3 extends JFrame implements Runnable{
23
24     public JLabel label;
25
26     public Exercicio3(){
27         label = new JLabel("Este é um Label");
28         JTextField textfield = new JTextField(10);
29         JButton gravar = new JButton("Gravar");
30         JPanel painel = new JPanel();
31         painel.add(textfield);
32         painel.add(gravar);
33         painel.add(label);
34         this.setSize(300, 200);
35         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36         this.getContentPane().add(painel);
37         this.setLocationRelativeTo(null);
38         this.setVisible(true);
39         Thread t = new Thread(this);
40         t.start();
41     }
42
43
44     @Override
45     public void run() {
46         for (int i = 0; i<300; i++){
47             try {
48                 Thread.sleep(1000);
49             } catch (InterruptedException ex) {
50                 Logger.getLogger(Exercicio3.class.getName()).log(Level.SEVERE, null, ex);
51             }
52         }
53         System.exit(0);
54     }
55
56     public static void main(String[] args) {
57         new Exercicio3();
58     }
59
60 }
61 }
```

3 Valores pela
GUI

3 Valores pela
Thread



4. Escreva um programa em Java que implemente uma interface gráfica para o jogo pedra, papel, tesoura. Neste jogo, o utilizador e o computador escolhem entre pedra, papel ou tesoura. Sabendo que pedra ganha de tesoura, papel ganha de pedra e tesoura ganha de papel, exiba no ecrã o vencedor: Utilizador (Este deverá introduzir o seu nome através da interface) ou Computador. Para essa implementação, assuma que o número 0 representa pedra, 1 representa papel e 2 representa tesoura. O seu programa deve permitir que o jogador possa escolher entre pedra, papel e tesoura através de um grupo de RadioButtons e depois através de um botão saber quem foi o vencedor.
- a) Apresente um Mockup/desenho de Interface de utilizador para o problema apresentado. Seja criativo para montar uma interface gráfica interessante para o jogo! (3v)



- b) Implemente a Interface da alínea anterior, lembrando que para cada jogada do computador deverá implementar uma função que calcula aleatoriamente um valor entre 0 e 2 para determinar quem ganhou, para isso use a seguinte função estática **randInt(int min, int max)** que pertence a classe **POOII**, que esta no pacote **Teste2**. A função **randInt** gera aleatoriamente um número entre min e max. Por exemplo, para sortear um número entre 0 e 10, a seguinte chamada deve ser realizada: (7v)

```
x = randInt(0, 10);
```

```

26 public class Exercicio4 extends JFrame {
27
28     JRadioButton pedra;
29     JRadioButton tesoura;
30     JRadioButton papel;
31     JTextField Jtnome;
32     String [] elementos;
33
34     public Exercicio4() {
35         this.elementos = new String [] {"Pedra","Papel","Tesoura"};
36         this.setSize(300, 200);
37         JPanel norte = new JPanel();
38         JPanel centro = new JPanel();
39         JPanel sul = new JPanel();
40         Container contentor = this.getContentPane();
41         JLabel nome = new JLabel("Nome: ");
42         this.Jtnome = new JTextField(10);
43         norte.add(nome);
44         norte.add(Jtnome);
45
46         this.pedra = new JRadioButton("Pedra");
47         this.tesoura = new JRadioButton("Tesoura");
48         this.papel = new JRadioButton("Papel");
49         ButtonGroup bg = new ButtonGroup();
50         bg.add(pedra);
51         bg.add(tesoura);
52         bg.add(papel);
53         centro.add(pedra);
54         centro.add(tesoura);
55         centro.add(papel);
56         this.pedra.setSelected(true);
57
58         JButton butao = new JButton("Jogar");
59         butao.addActionListener(new ActionListener() {
60
61             @Override
62             public void actionPerformed(ActionEvent e) {
63                 verificar();
64             }
65         });
66         sul.add(butao);
67
68         contentor.add(norte, BorderLayout.NORTH);
69         contentor.add(centro, BorderLayout.CENTER);
70         contentor.add(sul, BorderLayout.SOUTH);
71         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
72
73         this.setLocationRelativeTo(null);
74         this.setVisible(true);
75     }
76
77     private void verificar() {
78         int escolhaUtilizador = -1;
79         if (pedra.isSelected()) {
80             escolhaUtilizador = 0;
81         } else if (papel.isSelected()) {
82             escolhaUtilizador = 1;
83         } else if (tesoura.isSelected()) {
84             escolhaUtilizador = 2;
85         }
86         int eschoaPC = POOII.randInt(0, 2);
87         if (eschoaPC > escolhaUtilizador) {
88             if ((eschoaPC - escolhaUtilizador) < 2) {
89                 JOptionPane.showMessageDialog(rootPane, "Computador Escolheu: "+this.elementos[eschoaPC]+" \n"
90                     + Jtnome.getText()+" Escolheu: "+this.elementos[escolhaUtilizador]+" \n"
91                     + "Vencedor: Computador");
92             } else {
93                 JOptionPane.showMessageDialog(rootPane, "Computador Escolheu: "+this.elementos[eschoaPC]+" \n"
94                     + Jtnome.getText()+" Escolheu: "+this.elementos[escolhaUtilizador]+" \n"
95                     + "Vencedor: "+ Jtnome.getText());
96             }
97         } else if (eschoaPC < escolhaUtilizador) {
98             if ((escolhaUtilizador - eschoaPC) < 2) {
99                 JOptionPane.showMessageDialog(rootPane, "Computador Escolheu: "+this.elementos[eschoaPC]+" \n"
100                     + Jtnome.getText()+" Escolheu: "+this.elementos[escolhaUtilizador]+" \n"
101                     + "Vencedor: "+ Jtnome.getText());
102             } else {
103                 JOptionPane.showMessageDialog(rootPane, "Computador Escolheu: "+this.elementos[eschoaPC]+" \n"
104                     + Jtnome.getText()+" Escolheu: "+this.elementos[escolhaUtilizador]+" \n"
105                     + "Vencedor: Computador");
106             }
107         } else {
108             JOptionPane.showMessageDialog(rootPane, "Computador Escolheu: "+this.elementos[eschoaPC]+" \n"
109                 + Jtnome.getText()+" Escolheu: "+this.elementos[escolhaUtilizador]+" \n"
110                 + "Vencedor: Empate");
111         }
112     }
113
114     public static void main(String[] args) {
115         new Exercicio4();
116     }
117 }

```

3.5 Valores pela GUI

3.5 Valores pelo evento do jogo

POOII

```
7   package Teste1;
8
9   /**
10    *
11    * @author Manhica
12    */
13   public class POOII {
14
15       public static int randInt(int min, int max){
16           java.util.Random rand = new java.util.Random();
17           int randomNum = rand.nextInt((max-min)+1)+min;
18           return randomNum;
19       }
20
21   }
```

Sem cotação...
não foi pedido