



Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

Fundamentos de Sistemas Operativos - 2324SI

1º Trabalho Prático - Aula prática 2

Trabalho 1 - GUI Subdito

Robot

Nome do robot ☐ Abrir / Fechar Robot

Canal de Comunicação

Ficheiro do Canal ...

Nº msg ☐ Abrir Canal

Controle do Robot

Raio

Ângulo

Distância

☐ Ativar / Desativar Comportamento

Log

Docente Carlos Carvalho

Realizado por (Grupo 7):

Diogo Santos 48626

Pedro Silva 48965

João Fonseca 49707

26 de setembro de 2024

Conteúdo

1	Introdução	I
2	Desenvolvimento	II
2.1	Associação da biblioteca do robot	II
2.2	Novas classes	II
3	Conclusões	III
4	Bibliografia	III
5	Código Java GUI_Base, GUI_Subdito, GUI_Rei	IV

1 Introdução

Nesta segunda aula pretendia-se a implementação dos GUIs tanto do Rei quanto do Subdito através do GUI_Base previamente tratado no projeto anterior, assim como o tratamento dos elementos necessários para a conexão com o robot e a utilização da sua biblioteca, com o propósito de poder comunicar com este e passar-lhe instruções, tais como movimento em quatro direções e as características desse movimento (ângulo, raio e distância).

Adicionalmente, não só para questões de estética como também de utilização prática e eficiente de um ser humano, o campo de "Logs" será utilizado para comunicação com o utilizador dos processos que estão a acontecer.

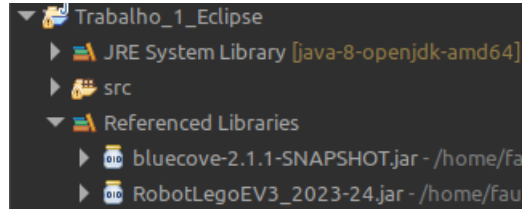
Como o tratamento da GUI do Rei e do Súbdito já foi efetuado na semana anterior e exposto no seu respetivo relatório, vamos seguir em frente e ignorar esse ponto neste presente.

2 Desenvolvimento

2.1 Associação da biblioteca do robot

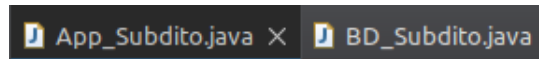
Antes de prosseguir com o projeto, é crucial que as bibliotecas que vão ser utilizadas estejam associadas ao projeto que está a ser desenvolvido. Para que tal aconteça no Eclipse IDE, basta fazer o seguinte:

$RMB[Projeto] \rightarrow Properties \rightarrow JavaBuildPath \rightarrow AddExternalJar$



2.2 Novas classes

Duas classes adicionais foram criadas com o propósito de melhor gestão de informação: **App.Subdito** e **BD.Subdito**.



A classe App.Subdito será responsável pela inicialização do programa e do GUI enquanto que a classe BD.Subdito será responsável por armazenar os métodos "GET" e "SET" das variáveis que correspondem à distância, raio, ângulo, nome do robot, ligado/desligado, terminar/não terminar e o próprio robot. Estas por sua vez serão chamadas no GUI.Subdito em métodos de addActionListener dos componentes do GUI.

Para além dos addActionListener, temos também os métodos actionPerformed associados aos botões. Estes métodos são específicos para os botões pois são estes que, após o input dos valores já ter sido feito, quando recebem o click do utilizador, irão fazer o robot cumprir as instruções na direção que for especificada.

3 Conclusões

Existem várias maneiras de se poder realizar esta porção do trabalho e uma das interações importantes dos membros do grupo foi a decisão da organização dos métodos e em que classes. Foi considerado fulcral a escolha de organização neste estágio do trabalho para tentar evitar ao máximo qualquer tipo de problemas que possam haver com o código daqui em diante.

Durante a aula um dos maiores problemas foi a própria conexão com o robot. Por alguma razão ainda desconhecida, certos portáteis reconheciam o robot mas outros não. Um dos que não reconheceu continha Windows e os que reconheceram eram um com Windows e outro com Linux. No entanto, nem sempre este era reconhecido pelo bluetooth. Certas vezes aparecia para emparelhar, outras necessitava dum refresh ao processo de bluetooth para que voltasse a aparecer. Pode-se perder imenso tempo com este processo todo por isso recomendamos alguma calma até que a conexão seja feita para que se evite perder demasiado tempo.

4 Bibliografia

1. Folhas de Computação Física - Jorge Pais, 2023/2024

5 Código Java GUI_Base, GUI_Subdito, GUI_Rei

```
1 Classe App_Subdito
2
3 package ptrabalho;
4
5 public class App_Subdito
6 {
7     @SuppressWarnings("unused")
8     private GUI_Subdito gui;
9     private BD_Subdito bd;
10
11     public App_Subdito()
12     {
13         bd = new BD_Subdito();
14         gui = new GUI_Subdito();
15     }
16
17     public BD_Subdito getBD()
18     {
19         return bd;
20     }
21
22     public void run() throws InterruptedException
23     {
24         while(!bd.getTerminar())
25             Thread.sleep(100);
26     }
27
28     public static void main(String[] args) throws
    InterruptedException
29     {
30         App_Subdito app = new App_Subdito();
31         System.out.println("A aplicação começou.");
32         app.run();
33         System.out.println("A aplicação terminou.");
34     }
35 }
36
37
38 Classe BD_Subdito
39
40 package ptrabalho;
41 import robot.RobotLegoEV3;
42
43 public class BD_Subdito
44 {
45     private RobotLegoEV3 robot;
46     private boolean terminar;
47     private boolean ligado;
48     private int distance;
49     private int angulo;
50     private float raio;
51     private String nome;
52
53     public BD_Subdito()
54     {
55         robot = new RobotLegoEV3();
56         terminar = false;
57         ligado = false;
58         distance = 30;
```

```

59     angulo = 45;
60     raio = 5;
61 }
62
63 public RobotLegoEV3 getRobot()
64 {
65     return robot;
66 }
67
68 public boolean getTerminar()
69 {
70     return terminar;
71 }
72
73 public void setTerminar(boolean b)
74 {
75     terminar = b;
76 }
77
78 public boolean isLigado()
79 {
80     return ligado;
81 }
82
83 public void setLigado(boolean b)
84 {
85     ligado = b;
86 }
87
88 public int getDist()
89 {
90     return distance;
91 }
92
93 public void setDist(int i)
94 {
95     distance = i;
96 }
97
98 public int getAng()
99 {
100     return angulo;
101 }
102
103 public void setAng(int i)
104 {
105     angulo = i;
106 }
107
108 public float getRaio()
109 {
110     return raio;
111 }
112
113 public void setRaio(float i)
114 {
115     raio = i;
116 }
117
118 public void setNome(String n)

```



```

119     {
120         nome = n;
121     }
122
123     public String getNome()
124     {
125         return nome;
126     }
127
128 }
129
130 GUI_Subdito
131
132 package ptrabalho;
133
134 import java.awt.EventQueue;
135
136 import javax.swing.BorderFactory;
137 import javax.swing.JFrame;
138 import javax.swing.JPanel;
139 import javax.swing.border.Border;
140 import javax.swing.border.EmptyBorder;
141 import javax.swing.border.LineBorder;
142 import javax.swing.border.TitledBorder;
143
144 import java.awt.Color;
145 import javax.swing.JLabel;
146 import java.awt.Font;
147 import java.awt.event.ActionEvent;
148 import java.awt.event.ActionListener;
149
150 import javax.swing.SwingConstants;
151 import javax.swing.JTextField;
152 import javax.swing.JRadioButton;
153
154 public class GUI_Subdito extends GUI_Base
155 {
156
157     private JPanel contentPane;
158     private JTextField txtNome;
159     private BD_Subdito bd;
160
161     /**
162      * Launch the application.
163      */
164     public GUI_Subdito()
165     {
166         super();
167         EventQueue.invokeLater(new Runnable()
168         {
169             public void run()
170             {
171                 try
172                 {
173                     init_GUI_Subdito();
174
175                 } catch (Exception e)
176                 {
177                     e.printStackTrace();
178                 }
179             }
180         });
181     }

```

```

179     }
180   });
181 }
182
183 /**
184  * Create the frame.
185  */
186 public void init_GUI_Subdito()
187 {
188     setTitle("Trabalho 1 - GUI Subdito");
189     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
190     setBounds(100, 100, 754, 695);
191
192
193     JLabel lblNomeDoRobot = new JLabel("Nome do robot");
194     lblNomeDoRobot.setHorizontalAlignment(SwingConstants.LEFT);
195     lblNomeDoRobot.setFont(new Font("Arial", Font.BOLD, 12));
196     lblNomeDoRobot.setBounds(30, 61, 103, 25);
197     getContentPane().add(lblNomeDoRobot);
198
199     btnFrt.addActionListener(new ActionListener()
200     {
201         public void actionPerformed(ActionEvent e)
202         {
203             bd.getRobot().Reta(bd.getDist());
204             bd.getRobot().Parar(false);
205         }
206     });
207
208     btnEsq.addActionListener(new ActionListener()
209     {
210         public void actionPerformed(ActionEvent e)
211         {
212             bd.getRobot().CurvarEsquerda(bd.getRaio(), bd.getAng());
213             bd.getRobot().Parar(false);
214         }
215     });
216
217     btnDir.addActionListener(new ActionListener()
218     {
219         public void actionPerformed(ActionEvent e)
220         {
221             bd.getRobot().CurvarDireita(bd.getRaio(), bd.getAng());
222             bd.getRobot().Parar(false);
223         }
224     });
225
226     btnTras.addActionListener(new ActionListener()
227     {
228         public void actionPerformed(ActionEvent e)
229         {
230             bd.getRobot().Reta(-(bd.getDist()));
231             bd.getRobot().Parar(false);
232         }
233     });
234
235     btnParar.addActionListener(new ActionListener()
236     {
237         public void actionPerformed(ActionEvent e)
238         {

```

```

239         bd.getRobot().Parar(true);
240     }
241 });
242
243     JRadioButton rdbtnAbrirFechar = new JRadioButton("Abrir /
Fechar Robot");
244     rdbtnAbrirFechar.addActionListener(new ActionListener()
245     {
246         public void actionPerformed(ActionEvent e) {
247             if (bd.isLigado())
248             {
249                 bd.getRobot().CloseEV3();
250                 bd.setLigado(false);
251             } else
252             {
253                 bd.setLigado(bd.getRobot().OpenEV3(bd.getNome()));
254             }
255         }
256     });
257
258     txtDist.addActionListener(new ActionListener()
259     {
260         public void actionPerformed(ActionEvent e)
261         {
262             bd.setDist(Integer.parseInt(txtDist.getText()));
263         }
264     });
265
266     txtAng.addActionListener(new ActionListener()
267     {
268         public void actionPerformed(ActionEvent e)
269         {
270             bd.setAng(Integer.parseInt(txtAng.getText()));
271         }
272     });
273
274     txtRaio.addActionListener(new ActionListener()
275     {
276         public void actionPerformed(ActionEvent e)
277         {
278             bd.setRaio(Float.parseFloat(txtRaio.getText()));
279         }
280     });
281
282     txtNome = new JTextField();
283     txtNome.addActionListener(new ActionListener()
284     {
285         public void actionPerformed(ActionEvent e)
286         {
287             bd.setNome(txtNome.getText());
288         }
289     });
290
291
292     txtNome.setFont(new Font("Arial", Font.BOLD, 12));
293     txtNome.setColumns(10);
294     txtNome.setBounds(143, 61, 279, 25);
295     getContentPane().add(txtNome);
296
297

```

```

298
299     rdbtnAbrirFechar.setFont(new Font("Arial", Font.BOLD, 12));
300     rdbtnAbrirFechar.setBounds(455, 61, 158, 25);
301     getContentPane().add(rdbtnAbrirFechar);
302
303
304     Border bords_robot= BorderFactory.createLineBorder(new Color
305     (0,0,0),1);
306     TitledBorder borda_robot = BorderFactory.createTitledBorder(
307     bords_robot, "Robot");
308     JPanel panel = new JPanel();
309     panel.setName("Canal de Comunicação");
310     panel.setBorder(new LineBorder(new Color(0, 0, 0)));
311     panel.setBounds(10, 34, 719, 66);
312     panel.setBorder(borda_robot);
313     getContentPane().add(panel);
314     contentPane = new JPanel();
315     contentPane.setBorder(new EmptyBorder(100, 100,100, 100));
316
317     setVisible(true);
318 }
319
320 Classe GUI_Rei
321
322 package ptrabalho;
323
324 import java.awt.Color;
325 import java.awt.EventQueue;
326 import java.awt.Font;
327
328 import javax.swing.BorderFactory;
329 import javax.swing.JButton;
330 import javax.swing.JFrame;
331 import javax.swing.JPanel;
332 import javax.swing.border.Border;
333 import javax.swing.border.EmptyBorder;
334 import javax.swing.border.LineBorder;
335 import javax.swing.border.TitledBorder;
336
337 public class GUI_Rei extends GUI_Base
338 {
339
340     private JPanel contentPane;
341
342     /**
343      * Launch the application.
344      */
345     public static void main(String[] args)
346     {
347         EventQueue.invokeLater(new Runnable()
348         {
349             public void run() {
350                 try
351                 {
352                     GUI_Rei frame = new GUI_Rei();
353                     frame.setVisible(true);
354                 } catch (Exception e)
355                 {

```

```

356         e.printStackTrace();
357     }
358 }
359 });
360 }
361
362 /**
363  * Create the frame.
364  */
365 public GUI_Rei()
366 {
367     super();
368     setTitle("Trabalho 1 - GUI Rei");
369     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
370     setBounds(100, 100, 754, 697);
371
372     Border cost = BorderFactory.createLineBorder(new Color(0,0,0),1);
373     TitledBorder borda_rei = BorderFactory.createTitledBorder(cost,
374 "Controle do Robot em Modo Automático");
375     JPanel panel_1_1 = new JPanel();
376     panel_1_1.setLayout(null);
377     panel_1_1.setName("Controle do Robot em Modo Automático");
378     panel_1_1.setBorder(new LineBorder(new Color(0, 0, 0)));
379     panel_1_1.setBounds(10, 34, 719, 66);
380     panel_1_1.setBorder(borda_rei);
381     getContentPane().add(panel_1_1);
382
383     JButton btnNewButton_2 = new JButton("8 Comandos Aleatórios");
384     btnNewButton_2.setFont(new Font("Arial", Font.BOLD, 12));
385     btnNewButton_2.setBounds(10, 26, 300, 25);
386     panel_1_1.add(btnNewButton_2);
387
388     JButton btnNewButton_2_1 = new JButton("16 Comandos Aleatórios");
389     btnNewButton_2_1.setFont(new Font("Arial", Font.BOLD, 12));
390     btnNewButton_2_1.setBounds(385, 26, 300, 25);
391     panel_1_1.add(btnNewButton_2_1);
392     contentPane = new JPanel();
393     contentPane.setBorder(new EmptyBorder(100, 100,100, 100));
394
395 }
396 }
397
398 Classe GUI_Base
399
400 package ptrabalho;
401
402 import java.awt.EventQueue;
403
404 import javax.swing.JFrame;
405 import javax.swing.JPanel;
406 import javax.swing.border.Border;
407 import javax.swing.border.EmptyBorder;
408 import javax.swing.BoxLayout;
409 import java.awt.GridLayout;
410
411 import javax.swing.BorderFactory;
412 import javax.swing.Box;

```

```

413 import java.awt.CardLayout;
414 import javax.swing.JTextField;
415 import java.awt.FlowLayout;
416 import javax.swing.JLabel;
417 import javax.swing.SwingConstants;
418 import java.awt.Font;
419 import javax.swing.JButton;
420 import javax.swing.JToggleButton;
421 import javax.swing.JSpinner;
422 import javax.swing.JRadioButton;
423 import javax.swing.JSeparator;
424 import javax.swing.JComboBox;
425 import javax.swing.JTextPane;
426 import java.awt.Panel;
427 import java.awt.Color;
428 import java.awt.Canvas;
429 import javax.swing.border.LineBorder;
430 import javax.swing.border.TitledBorder;
431 import javax.swing.JCheckBox;
432 import java.awt.event.ActionListener;
433 import java.awt.event.ActionEvent;
434 import javax.swing.JTextArea;
435
436 public class GUI_Base extends JFrame
437 {
438
439     private JPanel contentPane;
440     private JTextField textField;
441     protected JTextField txtRaio;
442     protected JTextField txtAng;
443     protected JTextField txtDist;
444     protected JButton btnFrt;
445     protected JButton btnEsq;
446     protected JButton btnDir;
447     protected JButton btnParar;
448     protected JButton btnTras;
449     protected JTextArea txtLog;
450
451     /**
452      * Launch the application.
453      */
454     public static void main(String[] args)
455     {
456         EventQueue.invokeLater(new Runnable()
457         {
458             public void run()
459             {
460                 try
461                 {
462                     GUI_Base frame = new GUI_Base();
463                     frame.setVisible(true);
464                 } catch (Exception e)
465                 {
466                     e.printStackTrace();
467                 }
468             }
469         });
470     }
471
472     /**

```

```

473  * Create the frame.
474  */
475  public GUI_Base()
476  {
477      setTitle("Trabalho 1 - GUI Base");
478      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
479      setBounds(100, 100, 754, 691);
480      contentPane = new JPanel();
481      contentPane.setBorder(new EmptyBorder(100, 100, 100, 100));
482
483      setContentPane(contentPane);
484      contentPane.setLayout(null);
485
486      textField = new JTextField();
487      textField.setFont(new Font("Arial", Font.BOLD, 12));
488      textField.setBounds(143, 156, 494, 25);
489      contentPane.add(textField);
490      textField.setColumns(10);
491
492      JLabel lblNewLabel = new JLabel("Ficheiro do Canal");
493      lblNewLabel.setFont(new Font("Arial", Font.BOLD, 12));
494      lblNewLabel.setHorizontalAlignment(SwingConstants.LEFT);
495      lblNewLabel.setBounds(30, 156, 103, 25);
496      contentPane.add(lblNewLabel);
497
498      JButton btnNewButton = new JButton("...");
499      btnNewButton.setFont(new Font("Arial", Font.BOLD, 12));
500      btnNewButton.setBounds(647, 157, 61, 25);
501      contentPane.add(btnNewButton);
502
503      JLabel lblNMsg = new JLabel("N    msg");
504      lblNMsg.setHorizontalAlignment(SwingConstants.LEFT);
505      lblNMsg.setFont(new Font("Arial", Font.BOLD, 12));
506      lblNMsg.setBounds(30, 200, 103, 25);
507      contentPane.add(lblNMsg);
508
509      JSpinner spinner = new JSpinner();
510      spinner.setFont(new Font("Arial", Font.BOLD, 12));
511      spinner.setBounds(143, 200, 50, 25);
512      contentPane.add(spinner);
513
514      JRadioButton rdbtnNewRadioButton = new JRadioButton("Abrir
Canal");
515      rdbtnNewRadioButton.setFont(new Font("Arial", Font.BOLD, 12));
516      rdbtnNewRadioButton.setBounds(459, 200, 103, 25);
517      contentPane.add(rdbtnNewRadioButton);
518
519      Border simpb = BorderFactory.createLineBorder(new Color(0,0,0)
,1);
520      TitledBorder border_simp = BorderFactory.createTitledBorder(
simpb,"Canal de Comunicação");
521      JPanel panel = new JPanel();
522      panel.setName("Canal de Comunicação");
523      panel.setBorder(new LineBorder(new Color(0, 0, 0)));
524      panel.setBounds(10, 128, 719, 113);
525      panel.setBorder(border_simp);
526      contentPane.add(panel);
527
528      JLabel lblRaio = new JLabel("Raio");
529      lblRaio.setHorizontalAlignment(SwingConstants.LEFT);

```

```

530 lblRaio.setFont(new Font("Arial", Font.BOLD, 12));
531 lblRaio.setBounds(30, 270, 103, 25);
532 contentPane.add(lblRaio);
533
534 txtRaio = new JTextField();
535
536 txtRaio.setFont(new Font("Arial", Font.BOLD, 12));
537 txtRaio.setColumns(10);
538 txtRaio.setBounds(143, 270, 100, 23);
539 contentPane.add(txtRaio);
540
541 JLabel lblngulo = new JLabel("Ângulo");
542 lblngulo.setHorizontalAlignment(SwingConstants.LEFT);
543 lblngulo.setFont(new Font("Arial", Font.BOLD, 12));
544 lblngulo.setBounds(30, 300, 103, 25);
545 contentPane.add(lblngulo);
546
547 JLabel lblDistncia = new JLabel("Distância");
548 lblDistncia.setHorizontalAlignment(SwingConstants.LEFT);
549 lblDistncia.setFont(new Font("Arial", Font.BOLD, 12));
550 lblDistncia.setBounds(30, 330, 103, 25);
551 contentPane.add(lblDistncia);
552
553 txtAng = new JTextField();
554 txtAng.setFont(new Font("Arial", Font.BOLD, 12));
555 txtAng.setColumns(10);
556 txtAng.setBounds(143, 300, 100, 23);
557 contentPane.add(txtAng);
558
559 txtDist = new JTextField();
560 txtDist.setFont(new Font("Arial", Font.BOLD, 12));
561 txtDist.setColumns(10);
562 txtDist.setBounds(143, 330, 100, 23);
563 contentPane.add(txtDist);
564
565 btnParar = new JButton("Parar");
566 btnParar.setFont(new Font("Arial", Font.BOLD, 12));
567 btnParar.setBounds(470, 297, 100, 25);
568 contentPane.add(btnParar);
569
570 btnFrt = new JButton("Frente");
571 btnFrt.setFont(new Font("Arial", Font.BOLD, 12));
572 btnFrt.setBounds(470, 269, 100, 25);
573 contentPane.add(btnFrt);
574
575 btnDir = new JButton("Direita");
576 btnDir.setFont(new Font("Arial", Font.BOLD, 12));
577 btnDir.setBounds(575, 298, 100, 25);
578 contentPane.add(btnDir);
579
580 btnEsq = new JButton("Esquerda");
581 btnEsq.setFont(new Font("Arial", Font.BOLD, 12));
582 btnEsq.setBounds(365, 297, 100, 25);
583 contentPane.add(btnEsq);
584
585 btnTras = new JButton("Tras");
586 btnTras.setFont(new Font("Arial", Font.BOLD, 12));
587 btnTras.setBounds(470, 325, 100, 25);
588 contentPane.add(btnTras);
589

```



```

590     Border borda_cont_robot = BorderFactory.createLineBorder(new
Color(0,0,0),1);
591     TitledBorder borda1 = BorderFactory.createTitledBorder(
borda_cont_robot,"Controle do Robot");
592     JPanel panel_1 = new JPanel();
593     panel_1.setFont(new Font("Arial", Font.BOLD, 12));
594     panel_1.setName("Controle do Robot");
595     panel_1.setBorder(new LineBorder(new Color(0, 0, 0)));
596     panel_1.setBounds(10, 251, 719, 113);
597     panel_1.setBorder(borda1);
598     contentPane.add(panel_1);
599
600     JCheckBox chckbxNewCheckBox = new JCheckBox("Ativar / Desativar
Comportamento");
601     chckbxNewCheckBox.setFont(new Font("Arial", Font.BOLD, 12));
602     chckbxNewCheckBox.setBounds(10, 365, 262, 45);
603     contentPane.add(chckbxNewCheckBox);
604
605     JLabel lblLog = new JLabel("Log");
606     lblLog.setHorizontalAlignment(SwingConstants.LEFT);
607     lblLog.setFont(new Font("Arial", Font.BOLD, 12));
608     lblLog.setBounds(10, 429, 103, 25);
609     contentPane.add(lblLog);
610
611     JButton btnLimpaLog = new JButton("Limpar Log");
612     btnLimpaLog.setFont(new Font("Arial", Font.BOLD, 12));
613     btnLimpaLog.setBounds(10, 630, 719, 25);
614     contentPane.add(btnLimpaLog);
615
616     JTextArea txtLog = new JTextArea();
617     txtLog.setBounds(30, 459, 683, 148);
618     contentPane.add(txtLog);
619 }
620 }

```