

```

1  /* Argomenti:
2      1) Layout Manager
3         BorderLayout
4      2) Le classi JLabel, JButton
5      3) Cenni alla progettazione della GUI (Graphic User Interface)
6
7
8
9      il JFrame è il componente che offre il contenitore principale (top
level container)
10     che accoglie gli altri componenti grafici disponendoli in tanti
modi diversi: per mezzo di Layout diversi!
11     Un Layout consente quindi di scegliere come disporre i componenti
grafici nel contenitore
12     Per fare ciò è necessario affidare questo compito a un
13     Layout Manager che ha lo scopo di fissare e gestire la disposizione
degli elementi all'interno di un
14     contenitore grafico (JFrame o altri contenitori grafici)
15     Un layout manager sarà un oggetto che dovremo creare scegliendo la
classe più opportuna tra quelle
16     che ci vengono messe a disposizione.
17     Uno dei layout manager disponibili è la BorderLayout (il layout
manager di default del JFrame)
18     Il Layout del BorderLayout suddivide il contenitore in 5 aree
diverse (vedi immagine)
19     Ciascuna area di un BorderLayout può accogliere solo un
componente che viene espanso
20     per occupare tutta l'area a lui riservata (area disponibile)
21
22     Come abbiamo detto il JFrame offre un solo contenitore principale
(top level), il ContentPane che, di default,
23     ha un layout manager di tipo BorderLayout.
24     Per ottenere il riferimento al ContentPane basta invocare il metodo
getContentPane() su un oggetto di tipo JFrame
25     e poi ...
26     Scopriremo la classe JLabel che permette di creare etichette e la
classe JButton che crea
27     bottoni (meglio pulsanti) che saranno collocati nel ContentPane
rispettando il layout prestabilito (quello di default BorderLayout).
28
29     Vedremo, nella lezione "Lezione_02_B_LayoutManager" che sarà
preferibile usare un contenitore intermedio JPanel ...
30
31
32
33
34
35 */
36
37 import java.awt.BorderLayout;
38 import java.awt.Color;
39 import java.awt.Container;
40 import java.awt.Dimension;
41 import java.util.HashSet;
42 import java.util.Set;
43 import javax.swing.JButton;
44 import javax.swing.JFrame;
45 import javax.swing.JLabel;
46

```

```

47 public class MiaFinestra extends JFrame{
48     private final int LARGHEZZA = 400;
49     private final int ALTEZZA = 300;
50
51
52     public MiaFinestra(String titolo, int larghezza, int altezza){
53         super(titolo);
54         init(larghezza,altezza);
55     }
56     public MiaFinestra(String titolo){
57         super(titolo);
58         init(LARGHEZZA,ALTEZZA);
59     }
60     private void init(int larghezza,int altezza){
61         this.setSize(larghezza,altezza);
62         this.setLocationRelativeTo(null);
63         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
64
65         // Da qui parte la lezione 2 A Layout Manager -> BorderLayout
66
67         Container topContainer = this.getContentPane();
68         /* il layout manager prestabilito sarà il BorderLayout */
69
70         JLabel lblNord = new JLabel("Etichetta NORD");
71         lblNord.setBackground(Color.ORANGE);
72         lblNord.setOpaque(true);
73         lblNord.setPreferredSize(new Dimension(150,150));
74
75         JLabel lblCentro = new JLabel("Etichetta CENTRO");
76
77         lblCentro.setBackground(Color.YELLOW);
78         lblCentro.setOpaque(true);
79         lblCentro.setPreferredSize(new Dimension(150,150));
80
81         JLabel lblOvest = new JLabel("Etichetta OVEST");
82         lblOvest.setBackground(Color.GREEN);
83         lblOvest.setOpaque(true);
84         lblOvest.setPreferredSize(new Dimension(150,150));
85
86         JButton btnEst = new JButton("Bottone EST");
87         btnEst.setPreferredSize(new Dimension(150,150));
88
89         JButton btnSud = new JButton("Bottone SUD");
90         btnSud.setPreferredSize(new Dimension(150,150));
91
92         topContainer.add(lblNord,BorderLayout.NORTH);
93         topContainer.add(lblCentro,BorderLayout.CENTER);
94         topContainer.add(lblOvest,BorderLayout.WEST);
95         topContainer.add(btnEst,BorderLayout.EAST);
96         topContainer.add(btnSud,BorderLayout.SOUTH);
97
98         this.setVisible(true);
99     }
100 }

```