

Java GUI Layout Managers

Bir çok işletim sisteminde, kullanıcı arayüzü component'leri, sabit pixel ölçümlerine göre düzenlenir. Örneğin, bir butonu frame'de (10,10)'luk yere eklediğimizde,x-kordinatında 10 pixel, y-kordinatında 10 pixel, bir işletim sisteminde bu buton düzgün gösterilirken, diğer bir işletim sisteminde düzgün görünmeyebilir. Bu nedenle, Java GUI'ye layout manager kavramı eklenmiştir. Bu layout manager sayesinde, işletim sisteminden bağımsız bir görünüm sağlanmış olur. Sonuçta ise tüm işletim sistemlerine göre layout manager eklenen componentlerin yerini belirlediği için düzgün bir görünüm sağlanmış olur.

Not: Content pane'lerin layout manager'ı olur. Bundan dolayı JFrame veya JPanel'in layout'u denildiği zaman aslında Content pane'in layout'u kastedilmektedir.

Temel olarak kullanılan üç layout manager vardır: **FlowLayout**, **GridLayout** ve **BorderLayout**

FlowLayout

En basit layout'tur. Component'ler soldan sağa doğru eklenme sırasına göre düzenlenir. Bir satır dolduğunda, yeni bir satır oluşur. Eklenen component'lerin dizilimini **FlowLayout.RIGHT**, **FlowLayout.CENTER** veya **FlowLayout.LEFT** sabitleriyle değiştirebiliriz.

Constructors:

FlowLayout(): Varsayılan constructor'dır.

FlowLayout(alignment: int): **FlowLayout.RIGHT**, **FlowLayout.CENTER** veya **FlowLayout.LEFT** değerlerinden birini parametre olarak alan constructor'dır.

FlowLayout(alignment: int, hgap: int, vgap: int): İlk parametresi **FlowLayout.RIGHT**, **FlowLayout.CENTER** veya **FlowLayout.LEFT** değerlerinden birini alırken, **hgap** parametresi horizontal gap(yatay boşluk) eklenen componentler arasındaki yatay boşluğun pixel cinsinden değerini belirtmesini sağlar. **vgap** (vertical gap:düsey boşluk) parametresi ise eklenen component'ler arasındaki düsey boşluğun pixel cinsinden değerini belirtmesini sağlar.

Örnek: `setLayout(new FlowLayout(FlowLayout.LEFT,10,20));`

GridLayout

GridLayout, componentleri matris formatına göre düzenler. Componentler soldan sağa doğru, ilk satırdan başlanarak eklenir.

Constructors:

GridLayout(): Varsayılan constructor'dır.

GridLayout(rows: int, columns: int): rows parametresi satır sayısını, columns parametresi sütun sayısını belirlemeyi sağlar.

GridLayout(rows: int, columns: int, hgap: int, vgap: int): Belirtilen sayıda satır ve sütundan oluşur. hgap arametresi iki component arasındaki yatay uzaklığı belirlemeyi, vgap parametresi iki component arasındaki düsey uzaklığı belirlemeyi sağlar.

Örnek: 10 componentin frame nesnesine eklendiğini farzedelim.

```
1 public class GridLayoutDemo{
2     public static void main(String[] args){
3         JFrame frame=new JFrame();
4         frame.setLayout(new GridLayout(3,0));
5         frame.setLayout(new GridLayout(0,3));
6         frame.setLayout(new GridLayout(0,0));
7         frame.setLayout(new GridLayout(3,3));
8         ....
9     }
10 }
```

4. Satır: Eğer frame nesnesinin layout'u bu şekilde tanımlanırsa satır sayısı sabit kalırken, sütun sayısı dinamik olarak belirlenir. Sonuç olarak ise 3 satır 4 sütunlu bir **GridLayout** oluşturulur.

5. Satır: Eğer frame nesnesinin layout'u bu şekilde tanımlanırsa, sütun sayısı sabit kalıp satır sayısı dinamik olarak belirlenir. Sonuç olarak ise, 4 atır 3 sütunlu bir **GridLayout** oluşturulur.

6. Satır: Hem satır sayısı hem sütun sayısı 0 olamaz. Bu yüzden 6. satırdaki gibi bir layout belirlenemez.

7. Satır: Hem satır sayısı hem de sütun sayısı sıfırdan farklı olursa, satır sayısı sabit kalır, sütun sayısı ise dinamik olarak belirlenir. Sonuç olarak ise, 3 satır 4 sütunlu bir **GridLayout** oluşturulur.

BorderLayout

Container sınıflarını(JFrame, JPanel...) 5 bölgeye ayırır: **Doğu, Batı, Kuzey, Güney, Merkez.**

Componentler, container sınıflarının add(Component c, index i) metodu ile eklenirler. Buradaki index parametresi, **BorderLayout.EAST**, **BorderLayout.WEST**, **BorderLayout.NORTH**, **BorderLayout.SOUTH**, **BorderLayout.CENTER** değerlerinden birini alır.

Örnek:

```
1  import javax.swing.JButton;
2  import javax.swing.JFrame;
3  import java.awt.BorderLayout;
4  public class ShowBorderLayout extends JFrame {
5      public ShowBorderLayout() {
6          // Yatay boşluk 5, dikey boşluk 10 birim olan BorderLayout
7          setLayout(new BorderLayout(5, 10) );
8          add(new JButton("East"), BorderLayout.EAST);
9          add(new JButton("South"), BorderLayout.SOUTH);
10         add(new JButton("West"), BorderLayout.WEST);
11         add(new JButton("North"), BorderLayout.NORTH);
12         add(new JButton("Center"), BorderLayout.CENTER);
13     }
14     /** Main metod */
15     public static void main(String[] args) {
16         ShowBorderLayout frame = new ShowBorderLayout();
17         frame.setTitle("ShowBorderLayout");
18         frame.setSize(300, 200);
19         //Frame'in ekranın ortasında açılmasını sağlar.
20         frame.setLocationRelativeTo(null);
21         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22         frame.setVisible(true);
23     }
24 }
```