GUI Layout Managers

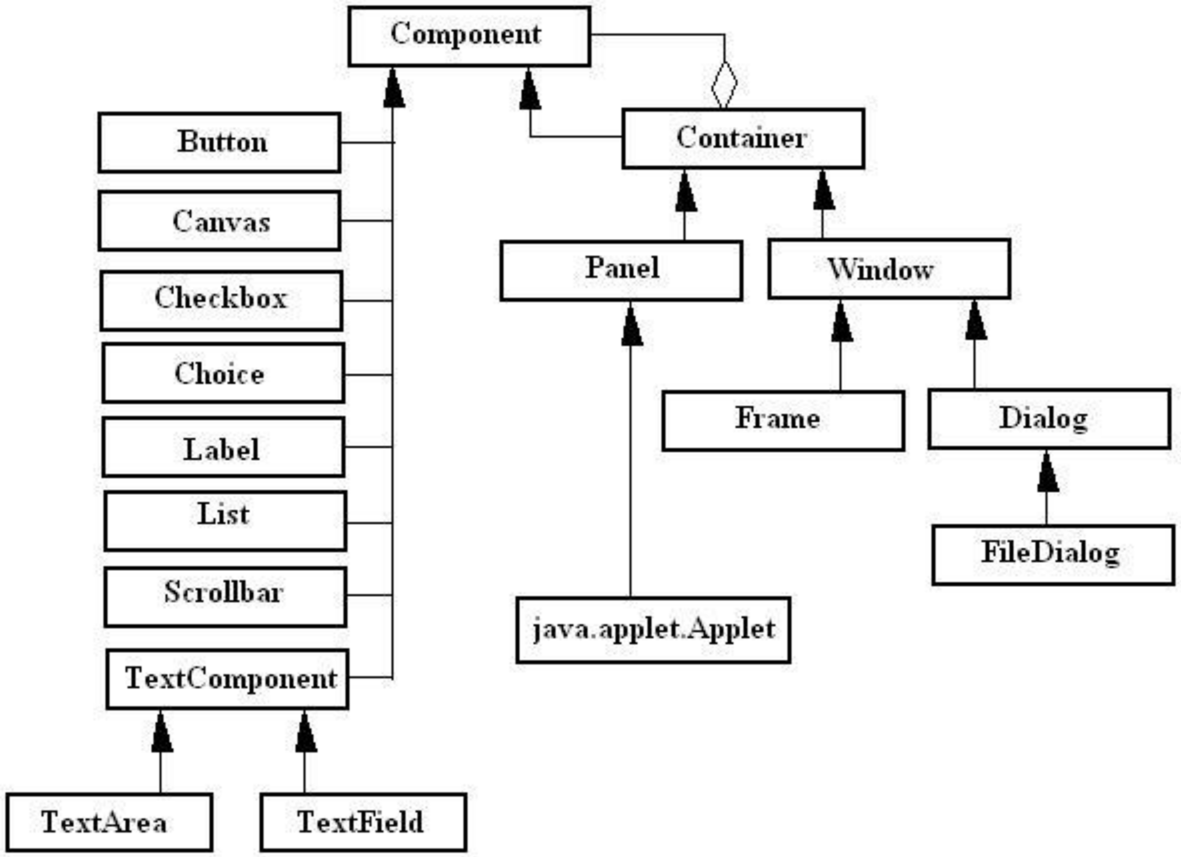
Outline

- Components and Containers
- Layout Managers
- Flow Layout
- Grid Layout
- Border Layout
- Nested Containers with Layouts
- Overview of Advanced Layout Managers

Components and Containers

- Components are building blocks of the visual aspect of the graphical user interface (GUI). Each GUI component has a characteristic appearance and behavior.
- <u>Components are divided into</u>: ones that can contain other components, *containers*, and the ones which may not, *primitive components*.

GUI component classes in AWT



Layout Managers

- Each container has a <u>layout manager, which controls the way the components are positioned in the container</u>.
- One of the advantages of using layout managers is that <u>there is no need for absolute coordinates</u> where each component is to be placed or the dimensions of the component. The <u>layout manager automatically</u> handles the calculation of these.
- Programmer <u>only specifies relative positions</u> of the components within the container.

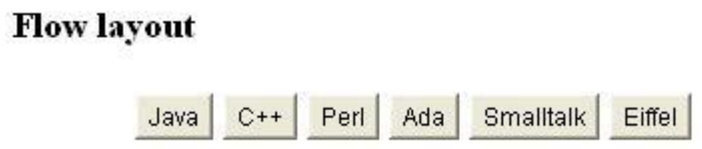
More Layout Managers

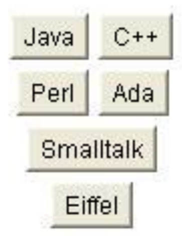
- Whenever the dimensions of the container change (e.g. user resizes the window), <u>layout manager recalculates</u> the absolute coordinates of components for them to fit the new container size.
- There are many different layout managers, this presentation describes the most popular ones.
- *Different layout managers can be used interchangeably and one inside the other.*

Flow Layout

- The Flow Layout manager arranges the components left-to-right, top-to-bottom in the order they were inserted into the container.
- When the container is not wide enough to display all the components, the remaining components are placed in the next row, etc.
- Each row is centered.

Flow Layout Examples





Flow Layout Constructors

`FlowLayout(align, hgap, vgap)`

align – alignment used by the manager

hgap – horizontal gaps between components

vgap – vertical gaps between components

`FlowLayout(align)`

align – alignment used by the manager

A default 5-unit horizontal and vertical gap.

`FlowLayout()`

A centered alignment and a default 5-unit horizontal and vertical gap.

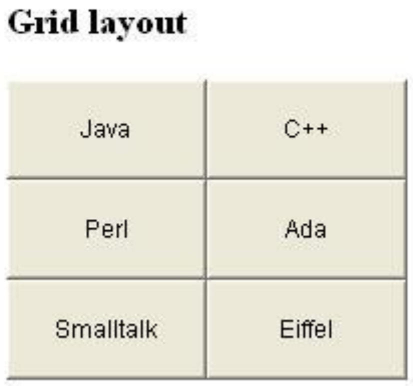
Flow Layout Alignment

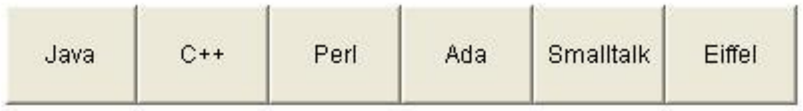
- The line alignment can be:
  - `FlowLayout.LEFT`
  - `FlowLayout.CENTER`
  - `FlowLayout.RIGHT`

Grid Layout

- The Grid Layout manager lays out all the components in a rectangular grid. All the components have identical sizes, since the manager automatically stretches or compresses the components as to fill the entire space of the container.

Grid Layout Examples





Grid Layout Constructors

`GridLayout(r, c, hgap, vgap)`

r – number of rows in the layout

c – number of columns in the layout

hgap – horizontal gaps between components

vgap – vertical gaps between components

`GridLayout(r, c)`

r – number of rows in the layout

c – number of columns in the layout

No vertical or horizontal gaps.

`GridLayout()`

A single row and no vertical or horizontal gaps.

Grid Layout - Notes

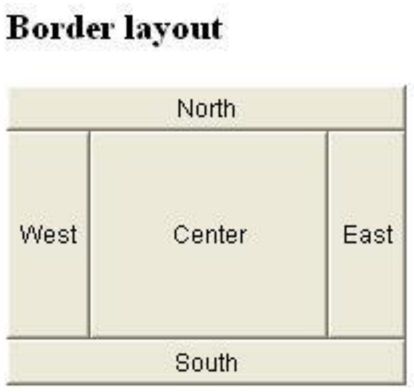
- Important constructor notes:
  - Parameter *r* (number of rows) or *c* (number of columns) can be equal to 0, then the grid can have any number of rows, or columns, respectively, depending on the number of components in the container.
  - Both *r* and *c* cannot be made 0 at the same time.

Border Layout

- The Border Layout manager arranges components into **five regions**: **North, South, East, West, and Center**.
- Components in the North and South are set to their natural heights and horizontally stretched to fill the entire width of the container.
- Components in the East and West are set to their natural widths and stretched vertically to fill the entire width of the container.
- The Center component fills the space left in the center of the container.

Border Layout Arrangement

- If one or more of the components, except the Center component, are missing then the rest of the existing components are stretched to fill the remaining space in the container.



Border Layout Constructors

`BorderLayout(hgap, vgap)`

hgap – horizontal gaps between components

vgap – vertical gaps between components

`BorderLayout()`

No vertical or horizontal gaps.

Border Layout Constraints

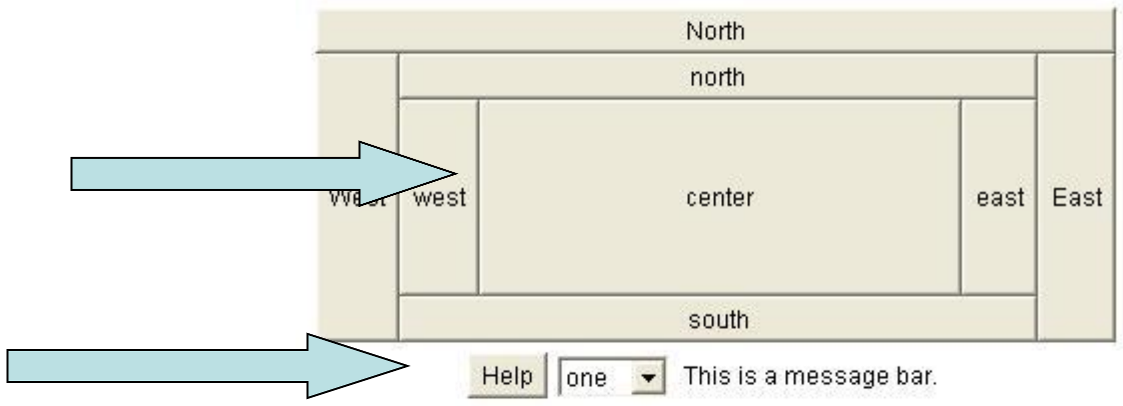
- The positional constraints are:
  - `BorderLayout.NORTH`
  - `BorderLayout.SOUTH`
  - `BorderLayout.EAST`
  - `BorderLayout.WEST`
  - `BorderLayout.CENTER`

Nested Containers with Layouts

- Each of the nested containers can have the same type or totally different type of a layout manager.
  - Outer Layout: Border Layout
  - Inner Layouts:

  Border Layout

  Flow Layout



Overview of Advanced Layout Managers

- Card Layout Manager
  - The Card Layout manager helps you manage two or more components (usually JPanel instances) that share the same display space. When using Card Layout, you need to provide a way to let the user choose between the components.
  - It is similar to tabbed panes, but a tabbed pane provides its own GUI, so using a tabbed pane is simpler than using Card Layout.

Overview of Advanced Layout Managers

  - Conceptually, each component a Card Layout manages is like a playing card or trading card in a stack, where only the top card is visible at any time.
- GridBag Layout Manager
  - GridBag Layout manager is one of the most flexible and complex layout managers the Java platform provides.
  - A GridBag Layout places components in a grid of rows and columns, allowing specified components to span multiple rows or columns.

Overview of Advanced Layout Managers

  - Not all columns or rows have to have the same dimensions.
  - GridBag Layout places components in cells in a grid, and then uses the components' preferred sizes to determine how big the cells should be.
- Group Layout Manager
  - Group Layout works with the horizontal and vertical layouts separately. The layout is defined for each dimension independently.