

## Менеджеры компоновок

Расположение компонент в апплете или окне устанавливается менеджером компоновок.

Для создания используется интерфейс **LayoutManager**.

**void setLayout(LayoutManager Lm)** – установка менеджера компоновок (из Container).

**Lm** – менеджер компоновок.

По умолчанию задан менеджер **FlowLayout**: компоненты следуют друг за другом в строке, если строка закончилась, то перенос на следующую строку.

Отключить : **setLayout(null);**

Если отключили, то теперь размерами и положением управляем с помощью метода (из Component)

**void setBounds(int x, int y, int width,int height);**

## 1. **FlowLayout**

**FlowLayout ()** - компоненты размещаются по центру, между каждым компонентом остается промежуток, равный пяти пикселям.

**FlowLayout(int how)** позволяет определить способ расположения каждой строки.

**FlowLayout(int how, int horz, int vert)** - позволяет определить промежутки по горизонтали и вертикали между компонентами.  
**how: FlowLayout.LEFT , FlowLayout.CENTER , FlowLayout.RIGHT** и др.

**2. BorderLayout** - реализует общий стиль компоновки для окон переднего плана. Имеет четыре узких компонента с фиксированной шириной по краям и одну большую область в центре. Четыре стороны имеют по сторонам света: север, юг, запад и восток.

**3. GridLayout** - размещает компоненты в двухмерной сетке (определяется количество строк и столбцов).

**4. CardLayout** - хранит несколько различных компоновок. Каждую компоновку можно представить в виде отдельной карточки из картотеки.

## Работа с изображением. Класс Image пакет java.awt.

Изображение – прямоугольный графический объект.

Форматы: jpg, gif, png.

Метод создания изображения:

**Image createImage(int width, int height)** – пустое изображение заданного размера;

**Image createImage(ImageProduser imgProd)** – пустое изображение, созданное с помощью объекта класса, реализующего интерфейс ImageProduser.

Просмотр изображения:

**boolean drawImage(Image img, int left, int top, ImageObserver imO)**

imO – класс – наблюдатель за процессом загрузки изображения.

## Интерфейсы **ImageObserver** и **ImageProduser**

**1. ImageObserver** – для получения уведомления о формировании изображения;

**boolean imageUpdate(Image img, int flags, int left, int top, int width, int height);**

**flags:** (параметрам соответствуют биты)

**WIDTH** и **HEIGHT** – если доступны базовые ширина и высота изображения;

**SOMEBITS** – были получены новые пиксели для рисования (left, top, width, height - прямоугольник);

**ALLBITS** – изображение готово (left, top, width, height не используются);

**ERROR** – ошибка, процесс формирования остановлен;

**ABORT** – формирование было прервано.

**2. ImageProduser** - интерфейс для объектов, которые должны подготовить данные для изображений. Объект, реализующий ImageProducer, задает целочисленный или байтовый массив, представляющий данные изображений, и формирует объекты Image.