

Android Studio e Linguaggio Java: Disegnare in Grafica

<p>classe Bitmap</p>	<p>Un oggetto di classe android.graphics.Bitmap contiene e gestisce una immagine di tipo Bitmap, ossia definita come un <i>insieme di "pixels"</i> ciascuno di un dato <i>colore</i></p> <p>Il metodo statico Bitmap.createBitmap consente di "creare" una immagine vuota, specificandone la <i>Risoluzione</i> e il tipo di <i>Codifica del Colore</i>:</p> <pre style="background-color: #e0f0e0; padding: 5px;">Bitmap miaImmagine = Bitmap.createBitmap (400, 400, Bitmap.Config.ARGB_8888);</pre> <p>... crea un oggetto <i>miaImmagine</i> di classe <i>Bitmap</i> contenente un'immagine di 400x400 pixels il cui colore è codificato con i 4 canali ARGB (A=trasparenza, Red, Green, Blue) ciascuno a 8 bit</p>
<p>classe BitmapFactory</p>	<p>La classe android.graphics.BitmapFactory offre dei metodi statici per recuperare immagini a partire da diverse fonti come una <i>Risorsa "drawable"</i>, un <i>File</i> su disco, un <i>Array di Byte</i>, ecc.</p> <p>Per creare un'immagine, recuperandola dalle Risorse "drawable", si usa il metodo statico BitmapFactory.decodeResource: è necessario indicare l'oggetto che contiene le Risorse dell'Applicazione (si ottiene con GetResources()) e l'ID della Risorsa Immagine desiderata:</p> <pre style="background-color: #e0f0e0; padding: 5px;">Bitmap miaImmagine = BitmapFactory.decodeResource (getResources(), R.drawable.pacman);</pre> <p>... recupera, dalle Risorse "drawable", l'immagine "pacman" e la memorizza nell'oggetto <i>miaImmagine</i></p>
<p>classi Paint e Color</p>	<p>Un oggetto di classe android.graphics.Paint consente di definire un "pennello", ossia un insieme di specifiche su come disegnare le linee (colore, spessore, tratteggio, ecc.)</p> <p>Per Creare un oggetto Paint, si usa normalmente new:</p> <pre style="background-color: #e0f0e0; padding: 5px;">Paint Rosso = new Paint ();</pre> <p>I metodi .setColor e .setStrokeWidth consentono di impostare Colore e Spessore:</p> <pre style="background-color: #e0f0e0; padding: 5px;">Rosso.setColor (Color.RED); Rosso.setStrokeWidth (5);</pre> <p>... ora il nostro pennello "Rosso" è impostato per tracciare in rosso e con spessore 5 pixels. La classe android.graphics.Color offre delle costanti che indicano i colori di base (Color.RED, Color.GREEN, ecc.)</p>
<p>classe Canvas</p>	<p>Un oggetto di classe android.graphics.Canvas viene <i>associato a una immagine</i> e consente di "disegnare" su di essa, tramite una serie nutrita di <i>metodi</i> specifici.</p> <p>Per Creare un oggetto Canvas, si usa normalmente new e lo si associa a una immagine:</p> <pre style="background-color: #e0f0e0; padding: 5px;">Bitmap miaImmagine = Bitmap.createBitmap (400, 400, Bitmap.Config.ARGB_8888); Canvas C = new Canvas (miaImmagine);</pre> <p>... adesso è possibile usare l'oggetto C per "disegnare" sull'immagine presente in <i>miaImmagine</i></p> <p>Alcuni metodi offerti dalla classe Canvas:</p> <p>C.drawColor (Color.GREEN); colora tutta l'immagine di verde C.drawLine (0, 0, 100, 100, Rosso); disegna una linea da (0,0) a (100,100) con il "pennello" Rosso C.drawBitmap (mioPac, 50, 50, Rosso); disegna un'immagine bitmap (mioPac) in posizione (50, 50)</p>
<p>classe ImageView</p>	<p>Un oggetto di classe android.widget.ImageView è una <i>View</i>, ossia si posiziona sulla Activity (come una <i>TextView</i>) e consente di visualizzare una immagine sulla Activity (usare il prefisso imv)</p> <p>Il metodo .setImageBitmap specifica l'immagine Bitmap da visualizzare:</p> <pre style="background-color: #e0f0e0; padding: 5px;">Bitmap miaImmagine = Bitmap.createBitmap (400, 400, Bitmap.Config.ARGB_8888); imvDisegno = findViewById (R.id.imvDisegno); imvDisegno.setImageBitmap (miaImmagine);</pre>