

Paket: *xPanel*, Klasse: *XBild* bzw. *B*

void **bereich**(double links, double unten, double rechts, double oben);

Das Koordinatensystem wird so eingestellt, dass die Ränder des Bildes bei den angegebenen Koordinaten liegen. Einer der Werte kann gleich der Konstante `B.AUTO` sein. Dieser Rand wird dann automatisch eingestellt, so dass die Einheiten auf der x- und der y-Achse gleich groß sind.

void **verschiebung**(double dx, double dy);

Der zuvor eingestellte Bereich wird um *dx* nach rechts und um *dy* nach oben verschoben.

double **randLinks**(), **randUnten**(), **randRechts**(), **randOben**();

Berechnet die Koordinaten der Ränder im aktuellen Koordinatensystem.

int **breite**(), **höhe**();

Berechnet die Breite und die Höhe des Bildes in Pixeln.

double **pixelBreite**(), **pixelHöhe**();

Berechnet die Breite und die Höhe eines Pixels in Einheiten des aktuellen Koordinatensystems.

double **yFaktor**();

Berechnet um welchen Faktor ein Pixel in den Einheiten des aktuellen Koordinatensystems höher ist als breit.

void **punkt**(double x, double y);

Ein Punkt wird gezeichnet.

void **linie**(double x1, double y1, double x2, double y2);

Eine Linie wird gezeichnet.

void **rechteckVoll**(double x1, double y1, double x2, double y2);

void **rechteckLeer**(double x1, double y1, double x2, double y2);

Ein Rechteck wird gezeichnet.

void **ellipseVoll**(double x1, double y1, double x2, double y2);

void **ellipseLeer**(double x1, double y1, double x2, double y2);

Eine Ellipse wird gezeichnet.

void **kreisLeer**(double mx, double my, double Radiusx);

void **kreisVoll**(double mx, double my, double Radiusx);

Ein Kreis wird gezeichnet.

void **bogenLeer**(double x1, double y1, double x2, double y2, double start, double winkel);

void **bogenVoll**(double x1, double y1, double x2, double y2, double start, double winkel);

Ein Bogen oder ein Sektor wird als Teil der Ellipse gezeichnet, die in das Rechteck mit den angegebenen Eckpunkten hineinpasst. *start* und *winkel* sind Winkel im Gradmaß.

void **bogenLeerRad**(double x1, double y1, double x2, double y2, double start, double winkel);

void **bogenVollRad**(double x1, double y1, double x2, double y2, double start, double winkel);

Bogenmaß-Version von *bogenLeer* und *bogenVoll*.

void **ecke**(double x, double y);

Eine Ecke eines Vielecks wird definiert.

void **vieleckVoll**();

void **vieleckLeer**();

Das aus den zuvor definierten Eckpunkten bestehende Vieleck wird gezeichnet.

void **stil**(double x);

Die Dicke einer Linien wird festgelegt.

void **farbe**(Color farbe), **farbe**(int farbcode), **farbe**(int r, int g, int b), **farbe**(String farbname);

In *zeichnen()* wird die aktuelle Zeichenfarbe festgelegt, außerhalb von *zeichnen()* die Hintergrundfarbe.

void **font**(String name, int style, int size);

Stellt die Schriftart ein. Z.B.: `B.font("Arial", Font.BOLD, 20)`. *style* ist gleich `Font.PLAIN`, `Font.BOLD` oder `Font.ITALIC`.

void **transparent**(boolean b);

Stellt ein, ob der Hintergrund unter einem Text gelöscht wird ($b = false$) oder nicht ($b = true$, Standard).

double **zeilenabstand**();

Berechnet den Zeilenabstand der aktuellen Schriftart in Einheiten des aktuellen Koordinatensystems.

void **text**(T, double x, double y);

void **text**(T);

void **text**(T, double x1, double y1, double x2, double y2);

void **text**(T, double x1, double y1, double x2, double y2, int position);

T kann eine Zahl oder ein String sein und wird an der angegebenen Position auf den Bildschirm geschrieben. Wird keine Position angegeben, wird hinter dem zuletzt ausgegebenen Text weitergeschrieben.

Durch zwei Eckpunkte kann auch ein Rechteck angegeben werden, in dem der Text zentriert wird. Durch die zusätzliche Angabe einer der Konstanten `B.LINKS`, `B.OBEN`, `B.RECHTS` oder `B.UNTEN` für *position* wird der Text in diesem Rechteck an den entsprechenden Rand geschoben.

void **doubleFormat**(String format);

z.B. "000,00". Legt das Format für die Ausgabe von Dezimalzahlen mit dem Befehl *text* fest.

Image **ladeBild**(String filename);

Image **ladeBild**(String filename, Color transparent);

Lädt ein Bild aus einer Datei. Es kann eine Farbe angegeben werden, die nicht gezeichnet wird (die entsprechenden Bildteile erscheinen transparent).

Image **definiereBild**(Color farbe, daten[]);

daten[] ist vom Typ `long` oder `int` und enthält zeilenweise die Pixeldaten für das zu definierende Bild.

boolean **bild**(Image img, double x, double y);

boolean **bild**(Image img, double x, double y, int position);

boolean **bild**(Image img, double x1, double y1, double x2, double y2);

boolean **bild**(double winkel, boolean spiegeln, double strecken, Image img, double x, double y);

boolean **bild**(double winkel, boolean spiegeln, double strecken, Image img, double x, double y, int position);

boolean **bild**(double winkel, boolean spiegeln, Image img, double x1, double y1, double x2, double y2);

Ein Bild wird gezeichnet und eventuell gespiegelt, um den Faktor *strecken* gestreckt und um *winkel*^o gedreht. *winkel* und / oder *spiegeln* und *strecken* können weggelassen werden.

Es kann ein Punkt (*x/ y*) gemeinsam mit *position* wie beim Befehl *text* angegeben werden, oder ein Rechteck. Dann wird das Bild gestreckt bzw. gestaucht, dass es in Rechteck passt.

boolean **bildRad**(...

Alle Methoden mit *winkel* haben eine Bogenmaß-Version.