

nach rechts und die y -Achse für die zweite Koordinate vertikal von oben nach unten. Alle Koordinaten für die Positionierung und Maße von grafischen Elementen werden somit in Pixel angegeben.

Mit Hilfe ihrer Methoden

- **public int** getHeight()
liefert die Höhe (in Pixel) der Komponente.
- **public int** getWidth()
liefert die Breite (in Pixel) der Komponente.

die wir bereits in Abschnitt 14.1 erwähnt haben, kann jede Komponente abfragen, welcher Bereich des Koordinatensystems ihr gerade zur Verfügung steht. Beträgt die Höhe h Pixel und die Breite b Pixel, so sind in x -Richtung die Pixel 0 bis $b-1$ und in y -Richtung die Pixel 0 bis $h-1$ verfügbar. Handelt es sich um eine Komponente, die Randlelemente wie z. B. eine Titelzeile, eine Menüleiste oder einen Rahmen beinhaltet, so ist zu beachten, dass nicht der gesamte Bereich nutzbar ist. Die abzweigenden Randbereiche können mittels der Methode

- **public Insets** getInsets()

ermittelt werden. Das zurückgelieferte Objekt der Klasse `Insets` besitzt vier Instanzvariablen `left`, `right`, `top` und `bottom` vom Typ `int`, in denen jeweils die Anzahl der links, rechts, oben und unten abzweigenden Pixel gespeichert ist.

16.1.3 Die abstrakte Klasse `Graphics`

In dieser Klasse werden zahlreiche Methoden bereitgestellt, die es ermöglichen, innerhalb des Grafikkordinatensystems einer Komponente zu zeichnen. Einige davon wollen wir hier kurz vorstellen.

- **void** drawLine(**int** x1, **int** y1, **int** x2, **int** y2)
zeichnet eine Linie vom Pixel (x_1, y_1) bis zum Pixel (x_2, y_2) .
- **void** drawPolyline(**int**[] x, **int**[] y, **int** n)
zeichnet eine Sequenz von Linien, die jeweils die Punkte $(x[i], y[i])$ und $(x[i+1], y[i+1])$ für $i = 0, 1, 2, \dots, n-2$ verbinden.
- **void** drawRect(**int** x, **int** y, **int** w, **int** h)
zeichnet ein Rechteck mit linker oberer Ecke (x, y) und rechter unterer Ecke $(x+w, y+h)$.
- **void** drawPolygon(**int**[] x, **int**[] y, **int** n)
zeichnet ein geschlossenes Polygon, das jeweils die Punkte $(x[i], y[i])$ und $(x[i+1], y[i+1])$ für $i = 0, 1, 2, \dots, n-2$ sowie die Punkte $(x[n-1], y[n-1])$ und $(x[0], y[0])$ verbindet.
- **void** drawOval(**int** x, **int** y, **int** w, **int** h)
zeichnet ein Oval, das in ein Rechteck mit linker oberer Ecke (x, y) und rechter unterer Ecke $(x+w, y+h)$ passt.