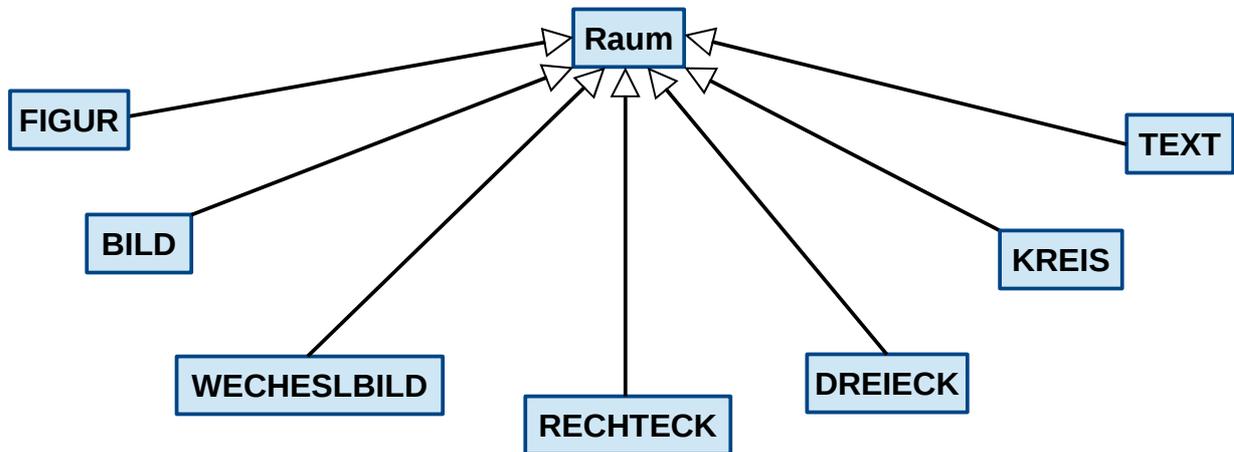


Die Super-Klasse Raum

Alle grafischen Klassen, die dir vorgegeben sind, **erben von einer Klasse Raum**.

Wundere dich also nicht, wenn du bei manchen Methoden den Datentyp `Raum` für den Übergabeparameter vorfindest. Das bedeutet nur, dass du hier jedes im Programmfenster sichtbare Objekt einsetzen kannst.



Dieses Prinzip, dass jede Sub-Klasse überall dort „willkommen“ ist, wo eigentlich eine Super-Klasse von ihr steht, nennt man **Polymorphie**.

Im englisch-sprachigen Raum nennt man das auch eine **is-a-Beziehung** und meint damit, dass z.B. jedes Bild-Objekt allgemein auch als ein Grafik-Objekt (`Raum`) angesehen werden kann, weil die Klasse `BILD` von der Klasse `Raum` erbt.

**Weil all diese Klassen von der Klasse `Raum` erben
verfügen sie auch über einige geerbte Methoden.
Einige von ihnen werden auf der Rückseite erklärt ...**

Gemeinsame Methoden aller Klassen

Die Super-Klasse *Raum*, von der alle Grafik-Klassen erben, bringt eine Menge an Methoden mit, die damit für jedes Grafik-Objekt zur Verfügung stehen.

```
berechneAbstandX( Raum r )
```

```
berechneAbstandY( Raum r )
```

Berechnet den horizontalen (*x*) bzw. vertikalen (*y*) Abstand in Pixel vom Mittelpunkt dieses Grafik-Objekts zum Mittelpunkt des übergebenen Grafik-Objekts.

```
beruehrt( Raum r )
```

Gibt *true* zurück, wenn dieses Grafik-Objekt das übergebene Grafik-Objekt berührt, *false* wenn nicht.

```
nenneM_x()
```

```
nenneM_y()
```

Gibt die *x*-Koordinate bzw. die *y*-Koordinate des Mittelpunkts dieses Grafik-Objekts zurück.

```
nenneSichtbar()
```

```
setzeSichtbar( boolean sichtbar)
```

Gibt *true* zurück, wenn dieses Grafik-Objekt gerade sichtbar ist.
Kann mit Übergabe-Wert *false* ein Grafik-Objekt unsichtbar machen und mit *true* wieder sichtbar.

```
setzeMittelpunkt( int x , int y )
```

Verschiebt dieses Grafikobjekt so dass sein Mittelpunkt die übergebenen Koordinaten hat.

```
verschiebenUm( int dx , int dy )
```

Verschiebt dieses Grafik-Objekt auf dem Bildschirm um *dx* Pixel in *x*-Richtung und um *dy* Pixel in *y*-Richtung. Da die *x*-Achse nach rechts zeigt, verschieben **positive *x*-Werte nach rechts**. Da die *y*-Achse nach unten zeigt, verschieben **positive *y*-Werte nach unten**.

Sieh dir von jeder Klasse die Klassen-Karte genau an .

(in BlueJ Rechts-Klick auf Klasse → Klassenkarte)

Es gibt noch viele weitere interessante Methoden ...