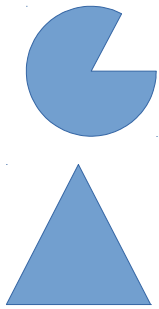


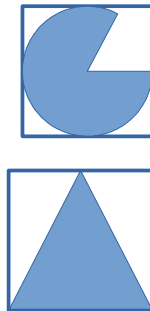
Aufgabe 2 : Treffen sie sich?

Bei vielen Computerspielen ist es wichtig zu wissen, ob sich zwei Objekte berühren. Das nennt man Kollisionserkennung (collision detection). Diese Kollisionserkennung kann je nach Komplexität der Körper sehr rechenaufwändig werden. Um den Berechnungsaufwand zu reduzieren, werden um Körper so genannte Hüllkörper (bounding box) gelegt. Bei diesen Hüllkörpern, die den komplexen Körper komplett einschließen handelt es sich um einfache geometrische Körper (z. B. im 2-dimensionalen ein Rechteck, im 3-dimensionalen ein Quader). Bei diesen Hüllkörpern ist die Kollisionserkennung leichter zu berechnen. Erst wenn sich zwei Hüllkörper berühren, wird die komplexe Berechnung gestartet, ob die Körper sich wirklich berühren.

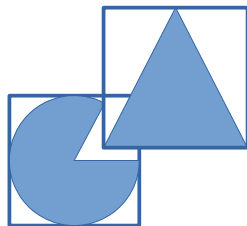
2 Komplexe Körper



2 Komplexe Körper in Hüllkörpern

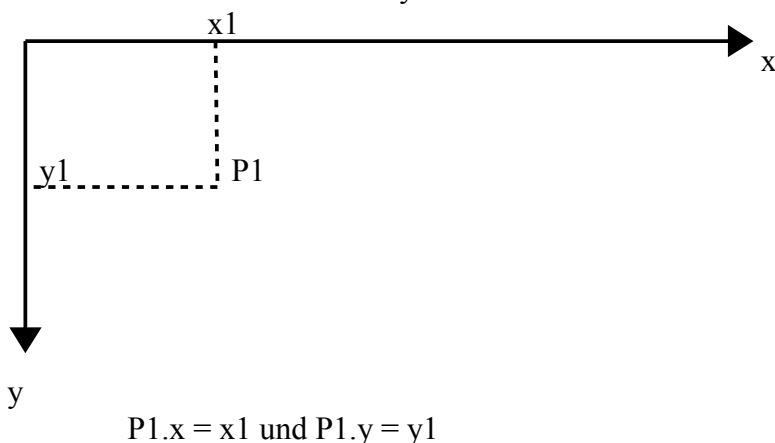


die Hüllkörper berühren sich, aber nicht die komplexen Körper

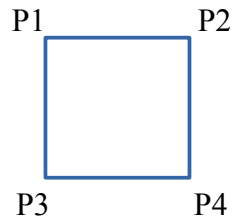


Das Koordinatensystem

In der Computergrafik ist der Ursprung ($x = 0$ [Rechtswert] und $y = 0$ [Untenwert]) des Koordinatensystems häufig links oben (in der Mathematik links unten). Dann wird von für die x-Koordinat nach rechts und für y-Koordinate nach unten hochgezählt.



Ein 2-dimensionaler Hüllkörper wird durch 2 Punkte (P1 und P4 mit jeweils X- und Y-Koordinaten beschrieben).



P2 und P3 lassen sich daraus berechnen:

$$\begin{array}{ll} P2.x = P4.x & \text{und} \quad P2.y = P1.y \\ P3.x = P1.x & \text{und} \quad P3.y = P4.y \end{array}$$

Zusammengesetzte Datentypen

Mit dem Befehl `typedef` kann man neue Datentypen aus anderen Datentypen erzeugen. Mit dem Datentyp `struct` kann man zusammengesetzte Variablen erzeugen.

```
typedef struct
{
    int x;
    int y;
} Punkt_t;
```

```
typedef struct
{
    Punkt_t p1;
    Punkt_t p2;
} Box_t;
```

```
Box_t box; /* hier wird eine Variable vom Typ Box erzeugt */
```

Der Zugriff auf einen Teil von `Box` erfolgt über den Punktoperator:

```
box.p1.x = 4; /* Zuweisung des Wertes 4 auf den x-Wert des Punktes p1 */
box.p2.y = 3; /* Zuweisung des Wertes 3 auf den y-Wert des Punktes p2 */
```

[Infos zu Strukturen](#)

Aufgabe:

Schreibe eine Funktion die eine 1 zurückgibt, wenn sich 2 Hüllkörper berühren. Wenn nicht dann eine 0.

Funktionssignatur: `int berührenSich(Box_t box1, Box_t box2)`