5.5.A Einführung und Klassen/Mehrere Objekte einer Klasse – Arbeitsblatt

Aufgabe 1: Mehrere Objekte der Klasse `punkttyp`

In dieser Aufgabe sollen Sie Funktionen und Methoden erstellen, die mit zwei oder mehr Punkten, also Objekten vom Typ `punkttyp` zu tun haben. Das Ergebnis der Aufgabe soll in `punkttyp5.cpp` geschrieben werden.

a) Was ist der Unterschied zwischen einer Funktion und einer Methode?

- _____
- _____

b) Die Funktion `pmaxy()` soll aus mehreren Punkten den größten y-Wert finden, also den y-Wert des Punktes, der im Koordinatensystem am weitesten oben liegt:

```
1 double pmaxy(punkttyp punktarray[], unsigned int anzahl);
```

Der Parameter `punktarray` soll dabei ein Array von Objekten mit dem Typ `punkttyp` sein. Der Parameter `anzahl` soll der Funktion sagen, wie viele Objekte im Array sind. Erstellen und testen Sie diese Funktion!

c) Eine Ebene wird durch die x- und die y-Achse des kartesischen Koordinatensystems in vier Teile, sogenannte Quadranten, unterteilt (Abbildung 1). Ein Punkt liegt nun in einem dieser

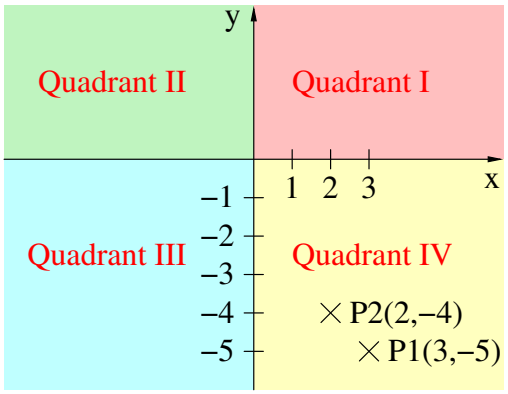

Abbildung 1: Quadranten

vier Quadranten. Die Funktion `pgleicherquadrant()` soll jetzt prüfen, ob zwei Punkte im gleichen Quadranten liegen, das heißt, ob die beiden x-Werte das gleiche Vorzeichen *und* die beiden y-Werte das gleiche Vorzeichen haben. Beispiel: P1(3,-5) und P2(2,-4) liegen im gleichen Quadranten (Nr. IV), weil beide x-Werte positiv und beide y-Werte negativ sind. Die Funktion hat folgenden Prototyp:

```
1 bool pgleicherquadrant(punkttyp p1, punkttyp p2);
```

Erstellen und testen Sie diese Funktion!

d) Nun soll die Berechnung, ob zwei Punkte im gleichen Quadranten liegen, in einer Methode der Klasse `punkttyp` stattfinden. Die Methode hat den Prototyp:

```
1 bool gleicherquadrant(punkttyp p2);
```

Erstellen und testen Sie diese Methode!

e) Zur Vorbereitung der folgenden Teilaufgabe braucht man eine Methode, die den Abstand r eines Punktes P zum Ursprung des Koordinatensystems, dem Punkt (0,0), berechnet und zurückgibt. Die Berechnung funktioniert nach dem Satz des Pythagoras mit $r^2 = x^2 + y^2$ bzw. mit $r = \sqrt{x^2 + y^2}$ (Abbildung 2). Diese Methode hat den Prototyp:

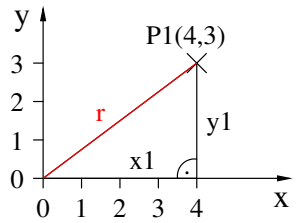

Abbildung 2: Abstand r eines Punktes p zum Ursprung

```
1 double urabstand(void);
```

Wie man sieht, hat diese Methode erstmal nur mit einem Objekt zu tun. Erstellen und testen Sie diese Methode! Denken Sie daran, für die Wurzel-Funktion `sqrt()` die Headerdatei `<math.h>` einzubinden und den `g++` mit der Option `-lm` aufzurufen!

f) Die Funktion `pmaxr()` soll aus mehreren Punkten den größten Abstand r zum Ursprung finden. Der Prototyp der Funktion ist:

```
1 double pmaxr(punkttyp punktarray[], unsigned int anzahl);
```

Erstellen und testen Sie diese Funktion!

g) Die Funktion `pabstand()` soll den Abstand a zweier Punkte P1(x1,y1) und P2(x2,y2) voneinander berechnen. Die Formel lautet (nach Pythagoras): $a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ (Abbildung 3). Der Prototyp ist:

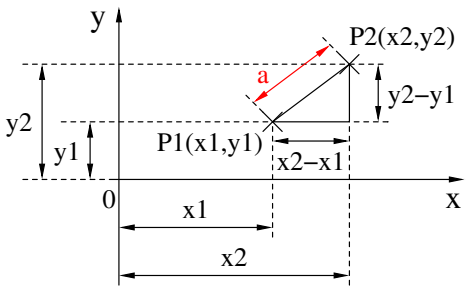

Abbildung 3: Abstand a zweier Punkte P1 und P2 voneinander

```
1 double pabstand(punkttyp p1, punkttyp p2);
```

Erstellen und testen Sie diese Funktion!

h) Nun soll der Abstand mit einer Methode der Klasse `punkttyp` berechnet werden:

```
1 double abstand(punkttyp p2);
```

Erstellen und testen Sie diese Methode!

i) Die Funktion `prwdreieck(p1, p2, p3)` soll herausfinden, ob die drei Punkte P1, P2 und P3 ein rechtwinkliges Dreieck bilden. Falls ja, muß für die Abstände a, b und c zwischen den drei Punkten eine der drei Formeln $a^2 + b^2 = c^2$, $b^2 + c^2 = a^2$ oder $c^2 + a^2 = b^2$ gelten. Die Funktion hat den Prototyp:

```
1 bool prwdreieck(punkttyp p1, punkttyp p2, punkttyp p3);
```

Erstellen und testen Sie diese Funktion!

j) Nun soll eine Methode herausfinden, ob die drei Punkte ein rechtwinkliges Dreieck bilden. Die Methode hat den Prototyp:

```
1 bool rwdreieck(punkttyp p2, punkttyp p3);
```

Erstellen und testen Sie diese Methode!

k) Die Funktion `pmittelpunkt(p1, p2)` soll die Mitte zwischen den beiden Punkten P1(x1,y1) und P2(x2,y2) berechnen und als Punkt (also als Objekt der Klasse `punkttyp`) zurückgeben. Die Koordinaten den Mittelpunkts PM berechnen sich mit $x_m = \frac{x_1+x_2}{2}$ und $y_m = \frac{y_1+y_2}{2}$ (Abbildung 4). Der Prototyp lautet:

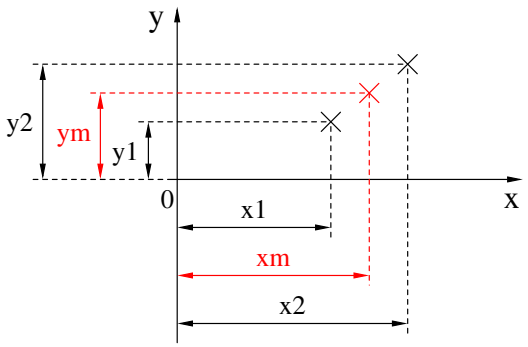

Abbildung 4: Mitte Pm zweier Punkte P1 und P2

```
1 punkttyp pmittelpunkt(punkttyp p1, punkttyp p2);
```

Erstellen und testen Sie diese Funktion!

l) Nun soll der Mittelpunkt in einer Methode der Klasse `punkttyp` berechnet werden. Der Prototyp lautet:

```
1 punkttyp mittelpunkt(punkttyp p2);
```

Erstellen und testen Sie diese Methode!