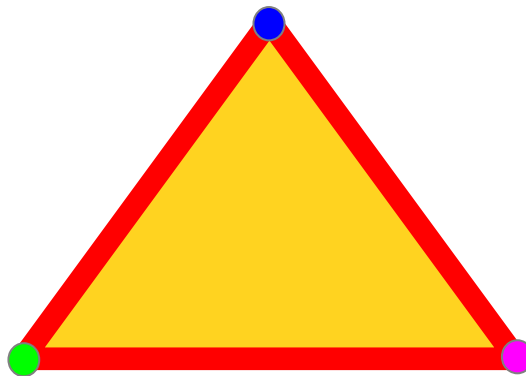


## Objektorientierte Programmierung

### Übungsaufgabe u05d Aggregation, Freundfunktionen (Erweiterung der Aufgabe u03d)



#### Lernziele:

- o **Strukturiertes planvolles schrittweises Vorgehen**
- o Aggregation von Klassen
- o Vorgabewerte bei Konstruktoren
- o Innere Konstruktorenkaskade für die Aggregationsbestandteile
- o Funktionen / Freundfunktionen

#### Aufgabenstellung:

Erstellen Sie eine Klasse **cDreieck**, die es ermöglicht, Dreiecke als Aggregation aus **drei Instanzen** einer ebenfalls zu erstellenden Klasse **cPunkt** zu erzeugen. Ein Punkt ist ein Objekt im 2-dimensionalen realen Koordinatensystem.

Gehen Sie wie folgt vor:

Erstellen Sie die Klasse **cPunkt** mit geeigneten Attributen für die Koordinaten eines Punktes.

Erstellen Sie für die Klasse **cPunkt** einen universellen Konstruktor, der den Punkt mit Vorgabewerten für den Nullpunkt erzeugt, falls keine Werte angegeben sind.

Bei der Konstruktion soll erreicht werden, dass die x- und y-Koordinaten jeweils im Bereich von  $\geq -10.0$  und  $\leq +10.0$  sind; Werte außerhalb des Bereichs müssen korrigiert werden. Erstellen Sie dafür ggf. eine private Hilfsmethode oder implementieren Sie es direkt im Konstruktor.

Erstellen Sie eine öffentliche Methode **void ausgabe()**, die die beiden Koordinatenwerte des Punktes ausgibt.

Erstellen Sie die Klasse **cDreieck** mit 3 Punkten als Attributen (Aggregation):

Erstellen Sie einen universellen Konstruktor, der das Dreieck **aus 3 Punkten** erzeugt.

Der Konstruktor soll mit Vorgabewerten ausgestattet sein, die das Dreieck aus den Punkten (0.0, 1.0), (1.0, 0.0), (0.0, 0.0) erzeugen.

Erstellen Sie eine private Methode **double umfangD()**, die den Umfang des Dreiecks berechnet und als Rückgabewert zurückgibt.

Erstellen Sie eine **öffentliche** Methode **double flaecheD()**, die die Fläche des Dreiecks berechnet und als Rückgabewert zurückgibt.

Erstellen Sie eine öffentliche Methode **void ausgabe()**, die die Punktwerte, den Umfang und den Flächeninhalt des Dreiecks auf dem Bildschirm ausgibt. Dabei soll die Ausgabefunktion der Klasse **cPunkt** verwendet werden.

Erstellen Sie eine **globale** Freundfunktion **int dreieckVergleich (cDreieck d1, cDreieck d2)** zum Vergleich zweier Dreiecke anhand der Flächengröße, die folgende Werte zurückgibt:

**1** wenn d1 größer ist d2

**0** wenn beide Dreiecke gleich groß sind

**-1** wenn d1 kleiner ist als d2

Erstellen Sie ein Hauptprogramm:

Instanzieren Sie 3 Dreiecke als ein Array aus cDreieck-Objekten.

Initialisieren Sie dabei 2 der 3 Dreiecke mit den folgenden Werten über eine Initialisierungsliste:

- o Dreieck aus 3 Punkten mit den Koordinaten (14.3, 2.41), (1.24, 16.3), (4.73, 6.42)
- o Dreieck aus 3 Punkten mit den Koordinaten (0.5, 1.0), (1.5, 0.0), (0.5, 0.0)

Geben Sie in einer Schleife die Werte aller 3 Dreiecke aus.

Vergleichen Sie die ersten beiden Dreiecke miteinander und geben Sie das Ergebnis aus.

Vergleichen Sie die letzten beiden Dreiecke miteinander und geben Sie das Ergebnis aus.

---

#### Hinweise:

Denken Sie an das Geheimnisprinzip: Wer kennt den Abstand zwischen zwei Punkten? Können das die Punkte selbst wissen oder braucht es dafür ein Dreieck?

Berechnung der Länge einer Strecke zwischen zwei Punkten A(a1, a2) und B(b1, b2) nach Pythagoras:

$$\overline{AB} = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}.$$

Berechnung der Dreiecksfläche nach dem Satz des Heron, wenn alle drei Seitenlängen a, b, c eines Dreiecks bekannt sind:

$$F = \sqrt{s(s - a)(s - b)(s - c)}$$

Dabei ist  $s = \frac{a+b+c}{2}$  der halbe Umfang des Dreiecks.