

# Grundlagen der Programmierung

## Mathematik und Malen

### Aufgabe 1: Die Klassen `Math` und `StrictMath`

Die Klassen `Math` und `StrictMath` sind in Java schon vorhanden und stellen einige numerischen Operationen als Klassenmethoden (bzw. Funktionen) zur Verfügung. Die Klasse `Math` werden Sie im Laufe des Praktikums immer wieder benötigen.

- a) Informieren Sie sich bitte in der Java API (Application Programming Interface) über die Klassen `Math` bzw. `StrictMath` und stellen Sie fest worin diese sich unterscheiden. Schauen Sie sich insbesondere die Klassenmethoden `sqr` und `hypot` an. Sie finden die API zu den Klassen `Math` und `StrictMath` auf den Seiten von Oracle (  
<https://docs.oracle.com/javase>).

- b) Berechnen Sie mit Hilfe der Klassenmethoden aus der Klasse `Math` die Länge der Hypotenuse in einem rechtwinkligen Dreieck, wenn die Katheten 100.0 und 75.0 lang sind und lassen Sie sich das Ergebnis mit Hilfe von `System.out.println` ausgeben. Ergänzen Sie die dazu nötigen Anweisungen in der `main`-Methode

Hinweis: Wenn die Längen der Katheten  $a$  und  $b$  eines rechtwinkligen Dreiecks bekannt sind, dann kann mit Hilfe des Satzes der Pythagoras die Länge der Hypotenuse  $c$  berechnet werden:  
 $a^2 + b^2 = c^2$ .

## Aufgabe 2: Geometrische Berechnungen

**Hinweis:** Auch wenn in dieser Aufgabe viele mathematische Formeln stehen, müssen Sie keine einzige Rechnung selbst durchführen! Sie müssen lediglich die angegebenen Gleichungen in die Programmiersprache Java übersetzen.

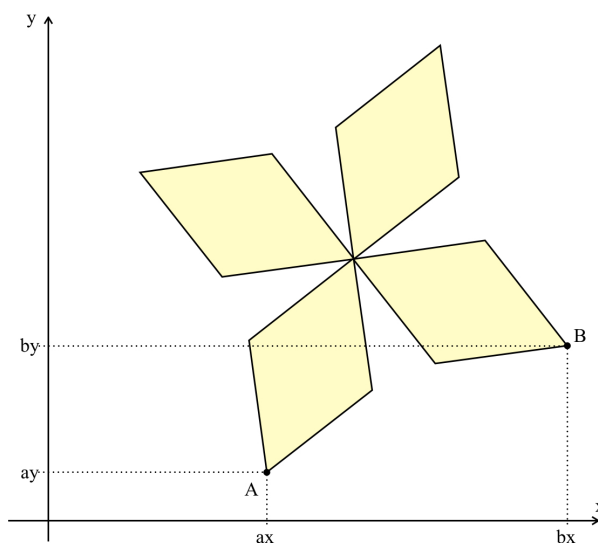


Figure 1: Blume

In dieser Aufgabe sollen Sie Klassenmethoden (bzw. Funktionen) schreiben, die geometrische Berechnungen durchführen. Diese Klassenmethoden werden wir in der nächsten Aufgabe zum Zeichnen von Blumen (siehe Abbildung 1) und im nächsten Praktikum zum Zeichnen von Bäumen verwenden. Wie Sie in der Abbildung 1 sehen, besteht eine Blume aus vier Rauten, wobei diese Rauten aus je zwei gleichseitige Dreiecken zusammengesetzt sind. Um die Blume zu malen, werden die Koordinaten aller Eckpunkte der Blume benötigt. Die Klassenmethoden, die Sie in den folgenden Teilaufgaben schreiben sollen, müssen die Funktionalität zur Berechnung dieser Koordinaten zur Verfügung stellen. Die Klassenmethoden müssen außerdem so angelegt sein, dass alle Koordinaten ausgehend von den Parametern  $A = (a_x, a_y)$  und  $B = (b_x, b_y)$  berechenbar sind!

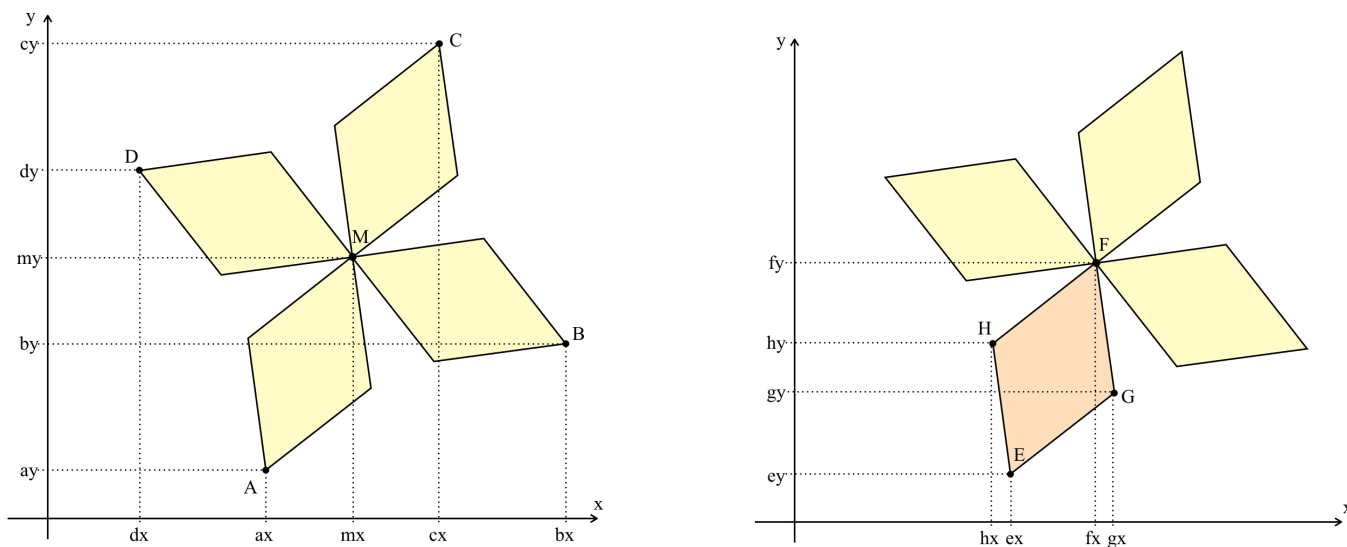


Figure 2: Links: Blume mit den Punkten  $A$ ,  $B$ ,  $C$ ,  $D$  und  $M$ . Rechts: Raute mit den Punkten  $E$ ,  $F$ ,  $G$  und  $H$ .

Da sich die Blume aus vier Rauten zusammensetzt, kann die Berechnung aller nötigen Koordinaten durch drei Schritte erfolgen. Zuerst können die Punkte berechnet werden, zwischen denen die vier Rauten liegen:  $A$ ,  $B$ ,  $C$ ,  $D$  und  $M$ .  $A$  und  $B$  sind ja schon bekannt, d.h. es müssen nur noch die Koordinaten der Punkte  $C$ ,  $D$  und  $M$  bestimmt werden (siehe linkes Bild in der Abbildung 2). Im zweiten Schritt kann die Funktionalität erstellt werden, die aus den Koordinaten der Punkte  $E$  und  $F$ , die Koordinaten der Punkte  $G$  und  $H$  berechnet (siehe rechtes Bild in der Abbildung 2). Im letzten Schritt kann dann diese Funktionalität verwendet werden, um alle Koordinaten zu berechnen, die zum Malen einer Blume erforderlich sind.

- a) In dieser Teilaufgabe sollen sie sechs Klassenmethoden schreiben, die die Koordinaten der Punkte  $C = (c_x, c_y)$ ,  $D = (d_x, d_y)$  und  $M = (m_x, m_y)$  berechnen.

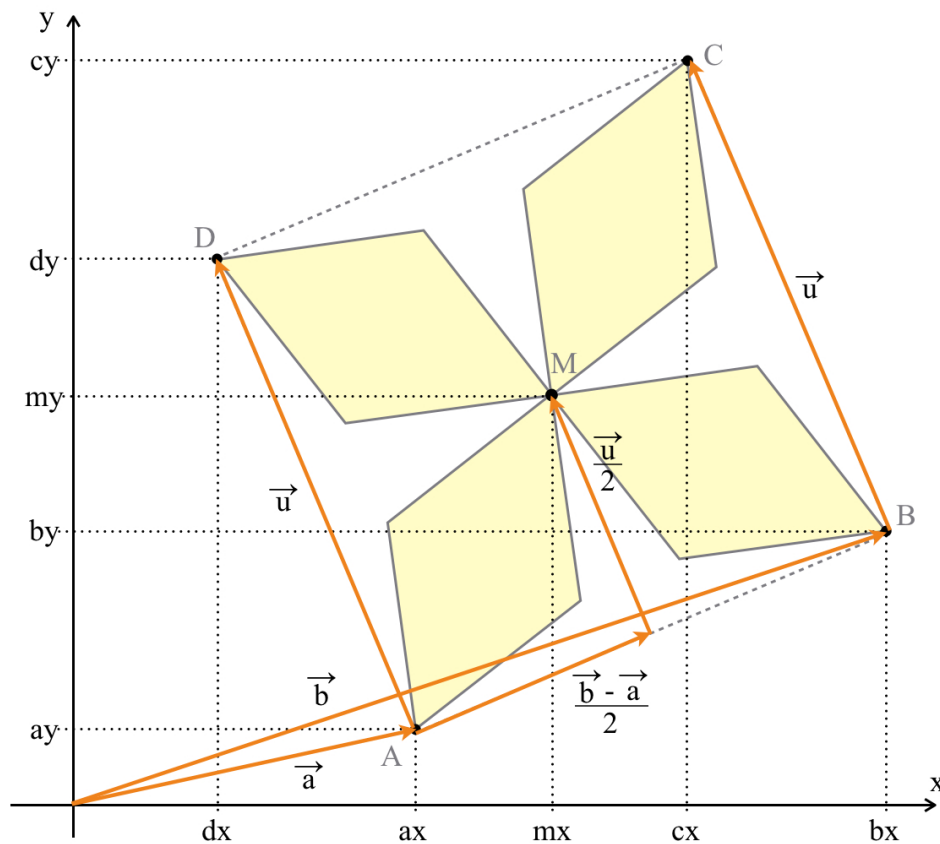


Figure 3: Berechnung der Koordinaten der Punkte  $C$ ,  $D$  und  $M$  aus den Punkten  $A$  und  $B$

Wenn man die Eigenschaft ausnützt, dass der Vektor  $\vec{u}$  senkrecht auf dem Vektor  $\vec{b} - \vec{a}$  steht und auch noch gleich lang ist, dann kann die Berechnung der Koordinaten der Punkte  $C$ ,  $D$  und  $M$  entsprechend der Abbildung 3 mit Hilfe der folgenden Vektoren erfolgen:

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad \vec{b} = \begin{pmatrix} b_x \\ b_y \end{pmatrix} \quad \vec{u} = \begin{pmatrix} a_y - b_y \\ b_x - a_x \end{pmatrix}$$

Diese Vektoren können jetzt verwendet werden, um die Koordinaten der Punkte  $C$ ,  $D$  und  $M$

entsprechend der Abbildung 3 zu berechnen:

$$\begin{aligned}\vec{c} &= \vec{b} + \vec{u} = \begin{pmatrix} b_x + a_y - b_y \\ b_y + b_x - a_x \end{pmatrix} \\ \vec{d} &= \vec{a} + \vec{u} = \begin{pmatrix} a_x + a_y - b_y \\ a_y + b_x - a_x \end{pmatrix} \\ \vec{m} &= \vec{a} + \frac{1}{2}(\vec{b} - \vec{a}) + \frac{1}{2}\vec{u} = \begin{pmatrix} a_x + \frac{1}{2}(b_x - a_x) + \frac{1}{2}(a_y - b_y) \\ a_y + \frac{1}{2}(b_y - a_y) + \frac{1}{2}(b_x - a_x) \end{pmatrix}\end{aligned}$$

Erstellen Sie auf Basis dieser drei Gleichungen sechs Klassenmethoden, die die Koordinaten der Punkte  $C$ ,  $D$  und  $M$  berechnen. Die Signatur der sechs Klassenmethoden sollten wie folgt lauten:

```
public static double cx (double ax, double ay, double bx, double by)
public static double cy (double ax, double ay, double bx, double by)
public static double dx (double ax, double ay, double bx, double by)
public static double dy (double ax, double ay, double bx, double by)
public static double mx (double ax, double ay, double bx, double by)
public static double my (double ax, double ay, double bx, double by)
```

**Tipp:** Sie müssen nur noch die obigen Gleichungen in die Programmiersprache Java übersetzen. Zum Beispiel ist der Programmcode der Klassenmethode `mx`:

```
public static double mx (double ax, double ay, double bx, double by) {
    return ax + (bx - ax)/2.0 + (ay - by)/2.0;
}
```

- b) Als nächstes benötigen wir die Funktionalität, die die Koordinaten der Punkte  $G = (g_x, g_y)$  und  $H = (h_x, h_y)$  aus den Koordinaten der Punkte  $E = (e_x, e_y)$  und  $F = (f_x, f_y)$  entsprechend der Abbildung 4 berechnet.

Auch hier können wieder einige geometrische Eigenschaften ausgenutzt werden, um die Berechnung durchzuführen. Zum einen kann ausgenutzt werden, dass sich die Raute in unserem Fall durch zwei gleichseitige Dreiecke zusammensetzt. Zum anderen kann die Strecke  $t$  mit Hilfe des Satzes des Pythagoras berechnet werden. Mit diesen beiden Überlegungen kann man die Strecken  $s$  und  $t$  wie folgt berechnen:

$$\begin{aligned}t &= \sqrt{(f_y - e_y)^2 + (f_x - e_x)^2} \\ s &= \frac{t}{\sqrt{3}}\end{aligned}$$

Erstellen Sie auf Basis dieser zwei Gleichungen zwei Klassenmethoden, die die Werte  $t$  und  $s$  berechnen:

```
public static double t (double ex, double ey, double fx, double fy)
public static double s (double ex, double ey, double fx, double fy)
```

**Tipp:** Verwenden Sie im Rumpf der Klassenmethode `s`, die Klassenmethode `t`.

Um die Koordinaten der Punkte  $G$  und  $H$  zu berechnen, müssen wir jetzt noch die Eigenschaft nutzen, dass der Vektor  $\vec{v}$  senkrecht auf dem Vektor  $\vec{f} - \vec{e}$  steht und  $\frac{s}{2t}$  lang ist, d.h.:

$$\vec{v} = \frac{s}{2t} \begin{pmatrix} f_y - e_y \\ e_x - f_x \end{pmatrix}$$


$$\begin{aligned}\vec{g} &= \vec{e} + \frac{1}{2}(\vec{f} - \vec{e}) + \vec{v} = \begin{pmatrix} e_x + \frac{1}{2}(f_x - e_x) + \frac{s}{2t}(f_y - e_y) \\ e_y + \frac{1}{2}(f_y - e_y) + \frac{s}{2t}(e_x - f_x) \end{pmatrix} \\ \vec{h} &= \vec{e} + \frac{1}{2}(\vec{f} - \vec{e}) - \vec{v} = \begin{pmatrix} e_x + \frac{1}{2}(f_x - e_x) - \frac{s}{2t}(f_y - e_y) \\ e_y + \frac{1}{2}(f_y - e_y) - \frac{s}{2t}(e_x - f_x) \end{pmatrix}\end{aligned}$$

**Hinweis:** Falls Sie wollen, dürfen Sie natürlich noch weitere als die hier vorgeschlagenen Klassenmethoden zur Berechnung definieren. Sie müssen aber nicht.

5

## Aufgabe 3: Blumen malen

In dieser Aufgabe sollen Sie Blumen malen. Zum Malen wird Ihnen die Klasse `SimpleGraphicPanel` in der jar-Datei `SimpleGraphicPanel.jar` zur Verfügung gestellt. Lesen Sie dazu die API zur Klasse `SimpleGraphicPanel` (siehe `SimpleGraphicPanel.pdf` auf dem Dokumentenserver).

**Hinweis:** Die jar-Datei können Sie in Netbeans einfach über das Fenster "Project Properties - <Projektname>" in Ihr Projekt <Projektname> einbinden.

Das Fenster "Project Properties - <Projektname>" können Sie öffnen, indem Sie mit der rechten Maustaste in der Projektliste auf das Projekt <Projektname> klicken und in dem sich öffnenden Menü `Properties` auswählen. Im Fenster "Project Properties - <Projektname>" selektieren Sie auf der linken Seite `Libraries` und wählen dann mit Hilfe des Buttons "Add JAR/Folder" die Datei `SimpleGraphicPanel.jar` aus. Anschließend schließen Sie das Fenster "Project Properties - <Projektname>".

Damit Sie die Klasse `SimpleGraphicPanel` und `Color` verwenden können, müssen Ihre ersten beiden Zeilen Ihrer Java-Datei folgende Import Anweisungen enthalten:

```
import edu.unibw.etti.SimpleGraphicPanel;
import java.awt.Color;
```

**Hinweis:** Die Klasse `Color` stellt verschiedene Farben zur Verfügung und kann ebenso wie ein primitiver Datentyp verwendet werden, d.h. Sie können z.B. eine Klassenmethode schreiben, in der eine Farbe als Parameter übergeben wird. Es gibt schon einige vordefinierte Werte für diesen Datentyp, z.B. `Color.RED` entspricht der Farbe Rot.

- a) Schreiben Sie eine Klassenmethode

`public static boolean maleRaute(double ex, double ey, double fx, double fy)`, die eine Raute malt, in dem Sie die Klassenmethode `drawQuadrilateral` der Klasse `SimpleGraphicPanel` verwendet. Zur Berechnung der Ecken der Raute verwenden Sie Ihre Klassenmethoden von der letzten Aufgabe. Ihre Klassenmethode soll zurückgeben, ob das Zeichnen der Raute erfolgreich war.

Testen Sie Ihre Klassenmethode.

- b) Schreiben Sie eine Klassenmethode

`public static boolean maleBlume(double ax, double ay, double bx, double by)`, die eine Blume malt, in dem Sie die Klassenmethode `maleRaute` verwendet. Zur Berechnung der Positionen der Raute verwenden Sie Ihre Klassenmethoden von der letzten Aufgabe. Ihre Klassenmethode soll zurückgeben, ob das Zeichnen der Blume erfolgreich war. Eine Blume wurde erfolgreich gezeichnet, wenn auch nur eine Raute erfolgreich gemalt wurde.

Testen Sie Ihre Klassenmethode.

- c) Erweitern Sie Ihre obigen Klassenmethode, so dass die Farbe als Parameter beim Malen einer Blume übergeben wird und die Blume dann in dieser Farbe gemalt wird

```
public static boolean maleBlume(Color color,
                                   double ax, double ay, double bx, double by).
```

Testen Sie Ihre Klassenmethode.