

## Erste Begegnung mit der Programmiersprache Java

### Aufträge:

1. Öffne die Datei *wasmacheich* aus dem Tauschverzeichnis in der Entwicklungsumgebung.
2. **Stopp! Noch nicht starten!** Lies dir die Zeilen zunächst einmal durch und stelle eine Vermutung auf, was passieren wird, wenn du das Programm startest.
3. Starte jetzt das Programm und vergleiche das Ergebnis mit deiner Vermutung. Hattest du Recht?
4. Um die folgenden Fragen zu beantworten, kannst du die Zahlen im Programmtext ändern, um zu sehen, was passiert. Merke dir aber immer, wie es vorher ausgesehen hat, damit du notfalls deine Änderungen rückgängig machen kannst.
5. Was bewirkt die Anweisung `size( b, h )` in der `setup`-Methode?
6. Was bewirkt die Anweisung `line( x1, y1, x2, y2 )`?
7. Was bewirkt die Anweisung `ellipse( x, y, b, h )`?
8. Was bewirkt die Anweisung `rect( x, y, b, h )`?
9. Die Leinwand, auf der gezeichnet wird, hat ein unsichtbares Koordinatensystem. Wo liegt der Ursprung (0|0)? Gibt es negative Koordinaten? In welche Richtung gehen x- und y-Achse hier? Was ist im Mathematikunterricht anders?

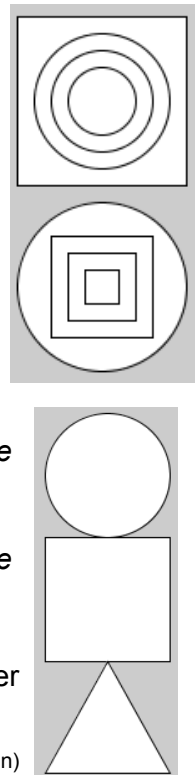
Es ist dir bestimmt aufgefallen, dass jede Anweisung mit einem **Semikolon** (Strichpunkt) abgeschlossen wurde. Wenn du selbst programmierst, darfst du das **nicht vergessen!**

Im Folgenden solltest du dir ein leeres kariertes Blatt Papier bereitlegen, auf dem du zu allem, was du programmieren möchtest, eine kleine Skizze machen kannst. Das erleichtert deine Arbeit sehr!

10. Jetzt darfst du dein erstes eigenes kleines Programm schreiben. Öffne dazu die Datei *figuren* aus dem Tauschverzeichnis.
11. Füge Programmzeilen so ein, dass das Bild C (Abb. rechts) aus der Gruppenarbeit gezeichnet wird.
12. Speichere dein Ergebnis ab.
13. \*\*\* Füge Programmzeilen so ein, dass Bild A (Abb. rechts) gezeichnet wird. Du kannst zum Zeichnen eines Dreiecks die Methode `triangle(x1,y1,x2,y2,x3,y3)` nutzen. (Es werden die Koordinaten der Eckpunkte des Dreiecks übergeben.)
14. \*\*\*\*\* Überlege dir selbst etwas, was du zeichnen lassen willst und programmiere deine Überlegungen in einer neuen Datei (Datei – Neu). Mache dir vorher eine Skizze auf Papier! Experimentiere auch mit Farbe.

Nutze für deine Zeichnungen die Informationen zum Zeichnen in deiner Entwicklungsumgebung.

Bilder: Screenshots von Ausführung des Programms „figuren“ (Eisenmann)

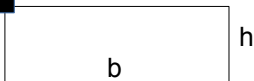


## Methoden zum Zeichnen

### Figuren

Punkt  $P(x | y)$

Strecke PQ mit  $P(x_1 | y_1)$  und  $Q(x_2 | y_2)$

Rechteck  $P(x | y)$   h

Ellipse mit Mittelpunkt  $M(x | y)$ , Breite b und Höhe h

Dreieck ABC mit  $A(x_1 | y_1)$ ,  $B(x_2 | y_2)$  und  $C(x_3 | y_3)$

Viereck ABCD mit  $A(x_1 | y_1)$ ,  $B(x_2 | y_2)$ ,  $C(x_3 | y_3)$  und  $D(x_4 | y_4)$

### Farbe

*Füllfarbe in Graustufen:*

Schwarz

Weiß

Graustufen

*Füllfarbe bunt:*

Eingabe einer RGB-Farbe (Werte von Rot, Grün, Blau zwischen 0 und 255)

Auswahl/Eingabe einer RGB-Farbe (Tools – Farbauswahl, Farbe wählen und Hexcode kopieren)

*keine Füllung*

Linienfarbe (wie oben)

keine Linie

Hintergrundfarbe einstellen

### Linienstärke

wird mit Pixelanzahl eingegeben

### Textausgabe

Text an Position  $(x | y)$  ausgeben

Schriftart wählen (Schriftart muss im Setup definiert werden)

```
point(x, y);
```

```
line(x1, y1, x2, y2);
```

```
rect(x, y, b, h);
```

```
ellipse(x, y, b, h);
```

```
triangle(x1, y1, x2, y2, x3, y3);
```

```
quad(x1, y1, x2, y2, x3, y3, x4, y4);
```

#### **Achtung:**

Zuerst Linien- und Füllfarbe ändern, dann zeichnen lassen!

```
fill(0);
```

```
fill(255);
```

```
fill(a); 0 < a < 255
```

```
fill(255, 0, 0); - rot
```

```
fill(#FF0000); - rot
```

```
noFill();
```

```
stroke(255); stroke(0, 255, 0);
```

```
noStroke();
```

```
background(122, 50, 113);
```

```
strokeWeight(3);
```

```
text("Hallo", x, y);
```

```
setFont(kleineSchrift);
```