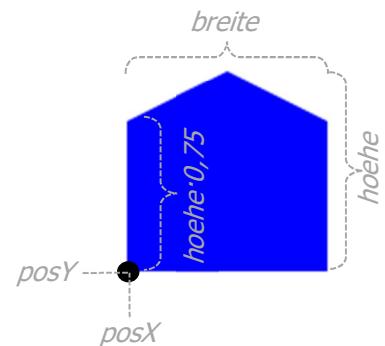


Algorithmen | Kapitel 6 Unterprogramme mit Zeichnungen

Ein Unterprogramm ist ein mehrfach wiederverwendbarer Programmblöck. Benötigt das Unterprogramm weitere Informationen (z.B. Zeichenposition, Größe, ...) verwendet man Parameter. Beim Ausführen des Unterprogramms werden die Parametervariablen mit den angegebenen Werten belegt.

Diese Unterprogramm `zeichneHaus` zeichnet ein Haus:

```
// Unterprogramm zum zeichnen eines Hauses
// posX: untere linke Ecke des Hauses (x-Wert)
// posY: untere linke Ecke des Hauses (y-Wert)
// breite: Breite des gesamten Hauses
// hoehe: Höhe des gesamten Hauses. 25% davon ist das Dach.
void zeichneHaus(int posX, int posY, int breite, int hoehe) {
    rect(posX, posY - hoehe*0.75, breite, hoehe*0.75);
    triangle(posX, posY - hoehe*0.75,
              posX + breite/2, posY - hoehe,
              posX + breite, posY - hoehe*0.75);
}
```



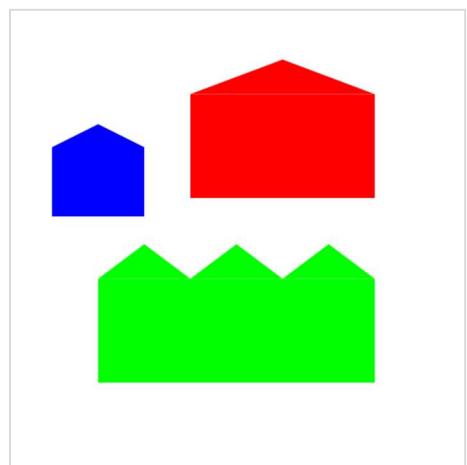
Im `draw`-Unterprogramm kann nun die neue `zeichneHaus`-Anweisung mehrfach verwendet werden, um mehrere Häuser zu zeichnen:

```
void draw() {
    background(200);
    noStroke();

    fill(0, 0, 255);
    zeichneHaus(50, 220, 100, 100);

    fill(255, 0, 0);
    zeichneHaus(200, 200, 200, 150);

    fill(0, 255, 0);
    zeichneHaus(100, 400, 100, 150);
    zeichneHaus(200, 400, 100, 150);
    zeichneHaus(300, 400, 100, 150);
}
```



Syntax von einem Unterprogramm (*ohne Rückgabewert*)

```
void bezeichnung( <Parameter1>, <Parameter2>, ... ) {
    // ...
}
```

Die Definition eines Unterprogramm besteht aus einem Bezeichner und der Parameterliste in der runden Klammer. Jeder Parameter wird mit Datentyp und Bezeichner angegeben. Ein Unterprogramm kann auch keine Parameter benötigen.

Bekannte Unterprogramme



Du kennst bereits die Unterprogramme `setup` und `draw` mit ihren Sonderrollen. Processing kennt noch mehr solche Unterprogramme, die in bestimmten Momenten automatisch ausgeführt werden.

Mehr dazu findest du auf dem Cheat-Sheet:

Unterprogramme zur Nutzerinteraktion in Processing

<code>void mousePressed()</code> { ... }	<code>void keyPressed()</code> { ... }
Maustaste wird (herunter) gedrückt	Taste wird (herunter) gedrückt
<code>void mouseReleased()</code> { ... }	<code>void keyReleased()</code> { ... }
Maustaste wird losgelassen	Taste wird losgelassen
<code>void mouseClicked()</code> { ... }	<code>void keyTyped()</code> { ... }

```
void setup() {
    // Wird einmalig zu Beginn
    // des Sketches ausgeführt
}
```

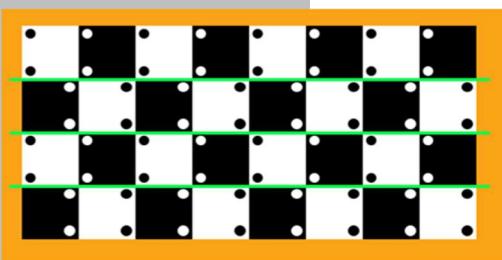
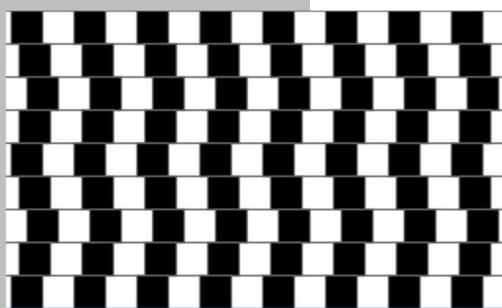
```
void draw() {
    // Wird wiederholt
    // immer wieder ausgeführt
}
```

1 Erste Unterprogramme

b)



- Verwende das Unterprogramm `zeichneHaus` um einige Häuser in verschiedenen Farben zu zeichnen.
 - Erstelle ein Unterprogramm `zeichneStrasse` mit den Parameter `int posY` und `int hoehe` um eine (horizontale) Straße – wie auf dem Bild – zu zeichnen.
- Erstelle mit deinen Unterprogrammen das Bild links – oder ein ähnliches Bild.
- Ergänze für die Straße eine gestrichelte Mittellinie. Definiere dazu zunächst ein Unterprogramm `void gestrichelteLinie(int posY)`. Ergänze dann für das Unterprogramm `zeichneStrasse` den Parameter `boolean zeichneMittellinie`, der entscheidet, ob die Mittellinie gezeichnet wird.
 - Ergänze beim Unterprogramm `zeichneHaus` zwei Parameter für die Dach- und Wandfarbe.



2  Die optische Täuschung besteht aus parallelen Linien zwischen denen sich weiße und schwarze Quadrate befinden. Gehe so vor, um die Zeichnung zu erstellen:

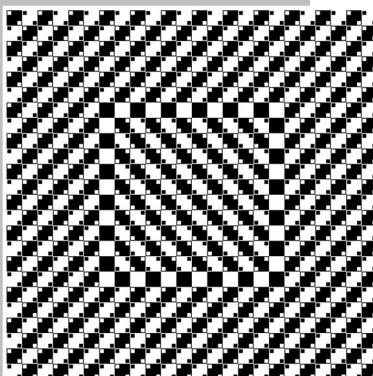
- Erstelle das Unterprogramm
`void zeichneStreifen(int posY, int hoehe, int abstandLinks)` um einen (horizontalen) Streifen zu zeichnen.
- Rufe das Unterprogramm mehrfach auf, um das Bild zu erstellen.
Profivariante: Rufe das Unterprogramm selbst mit einer Schleife auf.

3  Erstelle für diese optische Täuschung zunächst dieses Unterprogramm:

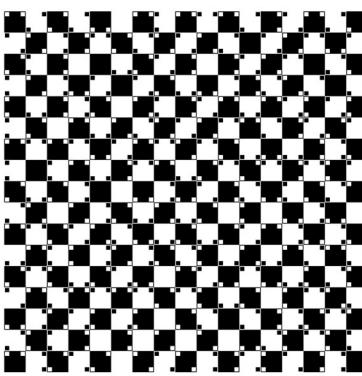
```
void zeichneQuadrat(int posX, int posY,  
                      boolean schwarz, boolean kreiseLinks)
```

Der Parameter `schwarz` gibt an, ob das Quadrat schwarz ist. Zeichne alle Quadrate 50 Pixel groß.

Vervollständige dann dein Programm, um die Zeichnung zu erstellen.



Feste Anordnungen



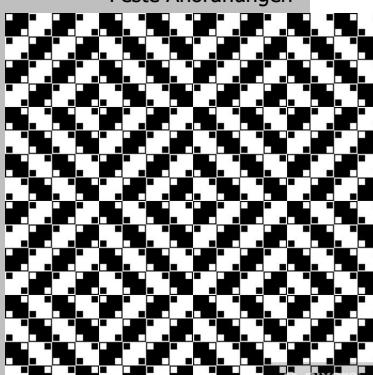
Zufällige Anordnung

4  Erstelle für diese optische Täuschungen zunächst ein Unterprogramm, um eines der Quadrate zu zeichnen. Überlege dir geeignete Parameter.

Erstelle eine der Täuschungen.

Tipp: Für einen zufälligen Wahrheitswert kannst du diese Anweisung verwenden:

```
boolean zufall = (int(random(2)) == 0);
```



5  Erstelle ein Sketch für eine weitere optische Täuschung. Setze dabei geschickt Unterprogramm ein.

Du findest die weiteren Täuschungen bei den Online-Lösungen.