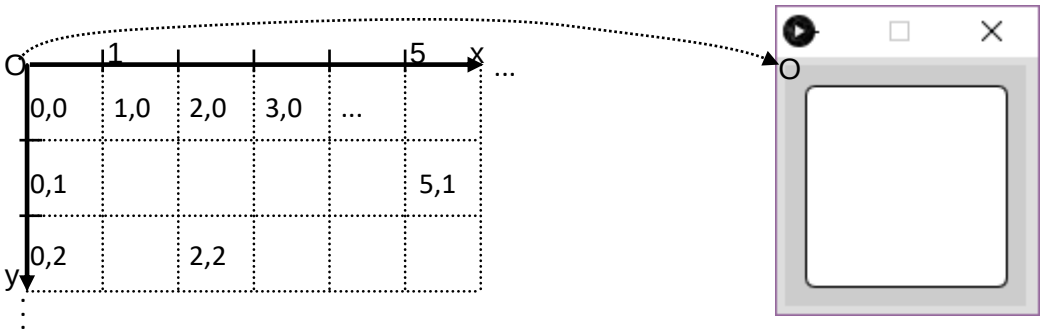
Zeichnungen erstellen mit Processing

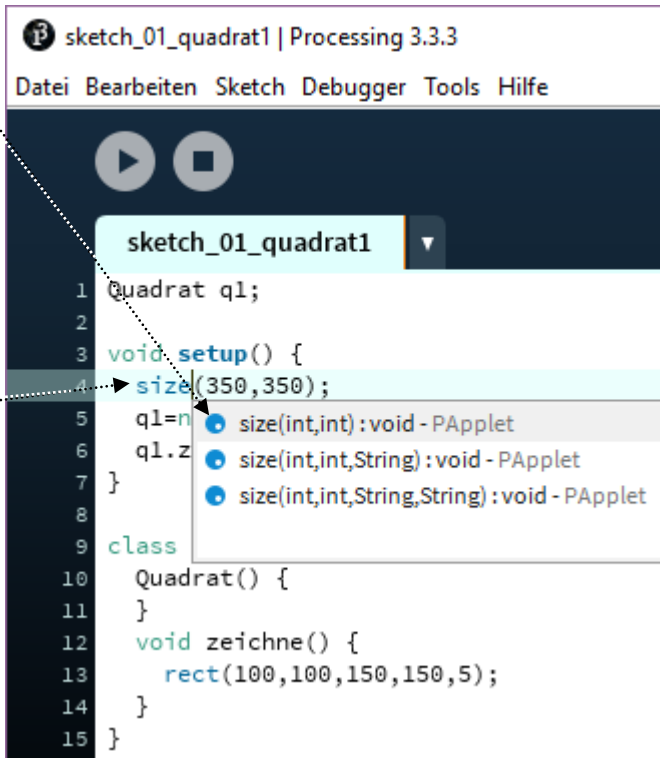
Bei dem Zeichenfenster handelt es sich um eine leere Zeichenfläche, in die geometrische Grundformen eingefügt werden können.

Koordinaten von Zeichnungelementen werden in einem Gitternetz angegeben.
Die linke obere Zelle des Zeichenfensters hat die Koordinaten (0,0). Als Trenner wird ein Komma verwendet (x,y), nicht der senkrechte Strich (x|y) wie in der Mathematik. Die Koordinaten werden in Pixel angegeben.
Die x-Koordinaten werden nach rechts hochgezählt, die y-Koordinaten nach unten:



Hinweise:

- Im Menü unter `Datei -> Einstellungen` kannst du die `Codevervollständigung` aktivieren.
  Um sie zu verwenden, gibst du den Anfangsbuchstaben der Anweisung ein und betätigst die Tastenkombination `<Strg>+<Leertaste>`.
- In den Lerninhalten sind auf Seite 6 die grundlegenden Anweisungen für Zeichnungselemente in Processing zusammengestellt, zum Beispiel:
  - size (Breite, Höhe); setzt die Größe des Zeichenfensters in Pixel. Wenn nichts anderes angegeben wird, verwendet Processing den Default-Wert (100,100).
  - rect(x, y, Breite, Höhe) zeichnet ein Rechteck mit dem Eckpunkt P(x|y) links oben. Die fünfte Zahl gibt den Radius der Abrundung an den Eckpunkten an.



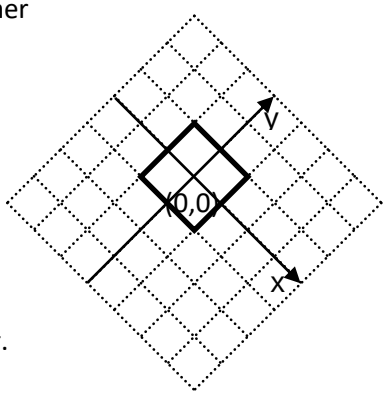
1. Einige Teile des Programms sind in der Abbildung oben durch die Codevervollständigung verdeckt. Ergänze die Anweisungen in Processing. Teste das Programm und speichere es als *quadrat1*.

q1:Quadrat
x=100 y=100 Breite=150 Höhe=150 Radius=5

Objektdiagramm des Objekts `q1`

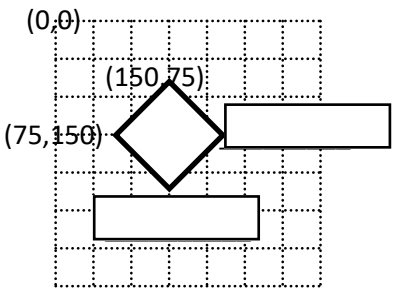
In der Folge soll das Verkehrszeichen „Vorfahrtsstraße“ wie in der Abbildung rechts gezeichnet werden. Dazu muss zuerst ein Quadrat wie in der Abbildung darunter um 45^0 gedreht werden.

Es können aber keine einzelnen Objekte gedreht werden, sondern immer nur die gesamte Zeichenfläche. Das wäre mit der Anweisung `rotate(radians(45));` möglich, bevor die Objekte gezeichnet werden. Da der Drehpunkt aber immer der Ursprung des Gittersystems ist, würden die Objekte dann außerhalb der Zeichenfläche liegen. Deshalb müsste der Ursprung zuvor mit `translate(100+150/2,100+150/2);` auf den Diagonalenschnittpunkt des Quadrats verschoben und gedreht werden. Dann wäre das Quadrat von hier aus mit `rect(-150/2,-150/2,150,150,5);` zu zeichnen.
(*vgl. .\262-materialien\vorfahrtszeichen\02-quadrat-gedreht*)



- Bei Zeichnungen mit mehreren Objekten ist die Handhabung der Anweisungen `rotate` und `translate` schwer durchschaubar. Es ist einfacher, das Quadrat als Polygon (Vieleck) zu zeichnen.

2. Berechne die fehlenden Koordinaten in dem Gitternetz rechts.



   - Ergänze das Objektdiagramm des Objekts `q1`:

q1:Quadrat
Linie=8
x1=150
y1=75
x2=75
y2=150
x3=
y3=
x4=
y4=

   - Erstelle ein Processing-Programm, das ein um 45 Grad gedrehtes Quadrat als Polygon zeichnet. Ergänze dazu in der Programmversion `quadrat1` nur die Zeichenbefehle. Du kannst auch die Vorlagendatei `sketch_v02_quadrat1` verwenden. Speichere Datei als `quadrat2`.

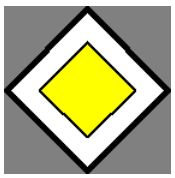
     Hinweise:
     - Mit `strokeWeight(Pixel)` kann die Linienstärke der Ränder festgelegt werden.
     - `strokeJoin(Art)` setzt die Art von Verbindungen an Eckpunkten: Ein Polygon mit abgerundeten Ecken wird mit `strokeJoin(ROUND)` gezeichnet. `BEVEL` schrägt die Ecken ab. Default ist `MITER` (engl. für Gehrung), wodurch die Ecken nicht verändert werden.
     - Ein Polygon wird mit der Anweisung `beginShape()` gezeichnet.
     - Danach werden die Koordinaten beliebig vieler Scheitelpunkte gesetzt:
       ```
       vertex(x1,y1)
       vertex(x2,y2)
       ...
       ```
     - Mit `endShape(CLOSE)` wird die Fläche vom letzten Punkt zum Anfangspunkt geschlossen.

Parameter

3. Nenne die Anweisung zum Zeichnen eines Scheitelpunkts mit den Koordinaten (150|75).

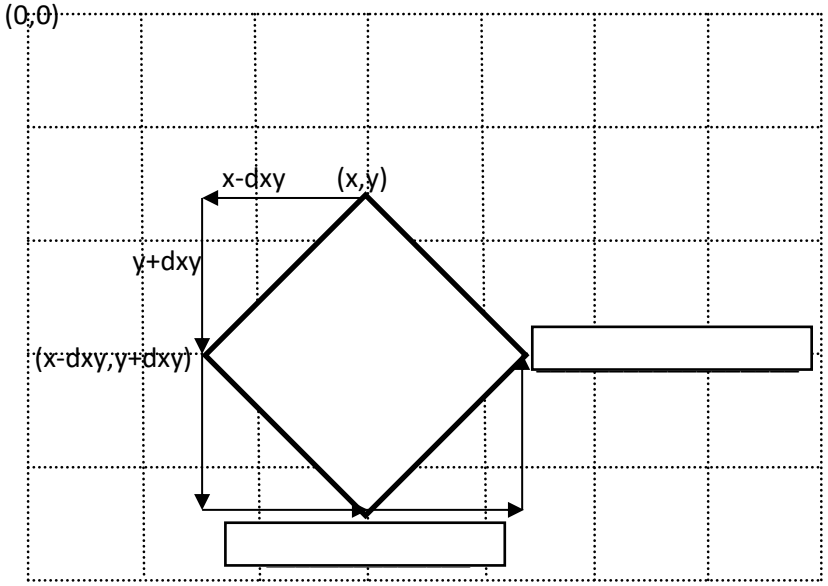
   Welche Werte müssen an die Funktion übergeben werden, um einen Punkt zeichnen zu können?

➢ Wenn Werte an eine Funktion oder Methode übergeben werden, bezeichnet man diese Werte als **Parameter**. Bei einem Methodenaufruf werden die Parameter in Klammern gesetzt.



Um das Verkehrszeichen „Vorfahrtsstraße“ zu erhalten, muss noch ein zweites, gelb gefülltes Quadrat gezeichnet werden. Es ist aber nicht sinnvoll, dieselben Anweisungen nochmals einzugeben. Vielmehr kann ein zweites Objekt mit anderen Attributwerten erstellt werden.
Dafür benötigt die Klasse `Quadrat` Attribute für die Koordinaten der Punkte und für die Linienstärke. Dann können die Werte als *Parameter* übergeben werden. Die weiteren Eckpunkte eines Quadrats können dann in Abhängigkeit des Anfangspunktes (x|y) gezeichnet werden. Man muss nur den Abstand von dem Anfangspunkt zu dem weiteren Punkt angeben. Dieses Attribut könnte man `dxy` nennen.

4. Ergänze die relativen Angaben für die Eckpunkte in der Skizze rechts.



Daraus ergibt sich das folgende Klassendiagramm:

Quadrat
Linie:int x:int y:int dxy:int
zeichne ()

Hinweis:
Beim Erstellen des Objekts müssen in dem **Konstruktor** die Attribute mit Werten belegt werden. Um zu kennzeichnen, dass Parameter übergeben werden, wird hier ein `p` vorangestellt:

```
Quadrat(int px,int py,int pdxy,int plinie) {
  x=px;
  y=py;
  dxy=pdxy;
  linie=plinie;
}
```

q1:Quadrat
Linie=
x=
y=
dxy=

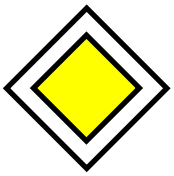
1. Ergänze das Objektdiagramm rechts oben und erstelle ein Klasse *Quadrat* mit Parametern und Konstruktor (`quadrat3`). (*vgl. .\262-materialien\vorfahrtszeichen\04_gedrehtesquadrat_parameter*)

Addierer
Summe:int s1:int s2:int
addiere () gibAus ()

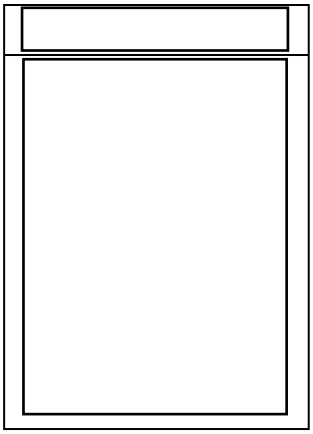
2. Zusatzaufgabe: Erstelle ein Programm mit den beiden Objekten `a1` und `a2` der Klasse `Addierer` zur Addition zweier Zahlen, die mit Parametern an den Konstruktor übergeben werden.

3. Mit der Programmversion *quadrat5.htm* soll ein Vorfahrtszeichen gezeichnet werden, indem ein zweites, gelb gefülltes, Quadrat erstellt wird.

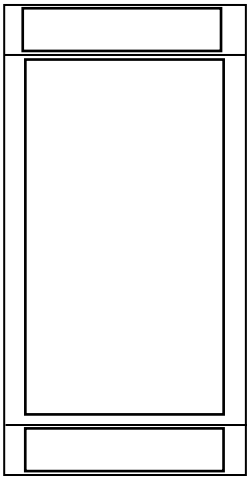
   Hinweis: Hier werden zusätzliche Attribute für die Füllfarbe benötigt. Mischfarben werden nach dem RGB Farbmodell mit den Grundfarben Rot, Grün und Blau (r,g,b) festgelegt. Die Werte für die Intensität der Grundfarben liegen im Bereich zwischen 0 und 255. (0,0,255) ergibt reines Blau, (255,255,0) ergibt Gelb.
   (vgl. Arbeitsblatt 1.4–07: Farbmodelle, Seite 1)

   Objektdiagramm:

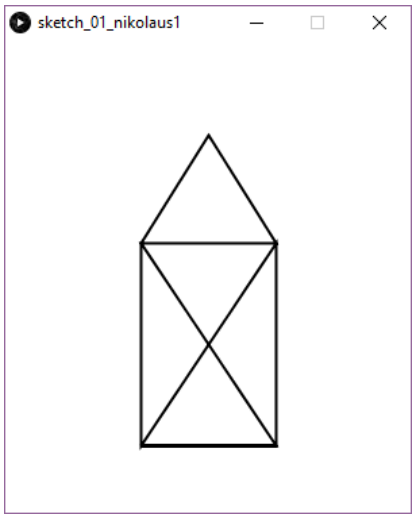
   

   Klassendiagramm:

   - Ergänze das Klassendiagramm rechts.
   - Zeichne das zweite Quadrat in die Skizze auf der vorigen Seite ein und bestimme die Werte für x, y und dxy.
     Ergänze dann das Objektdiagramm für das Objekt `q2` rechts.
   - Ergänze in der Programmversion `quadrat3` ein Objekt q2 für das gelbe Quadrat (`vorfahrt1`).

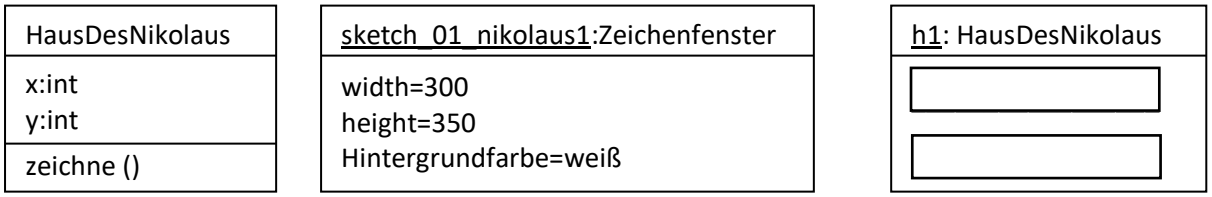
4. Zeichne mit nur einer `Shape`-Anweisung das „Haus des Nikolaus“ als Polygon in Abhängigkeit eines Startpunktes (x|y).
   Die Reihenfolge der Scheitelpunkte soll so angeordnet sein, dass man einen Stift nicht absetzen müsste, wenn man die Punkte von Hand verbinden würde.

   Hinweise zum Programmtest und zur Programmoptimierung:
   - Mit { begonnene Sequenzen müssen mit } beendet werden. Um den Überblick zu behalten, solltest du die Sequenzen immer einrücken.
   - Achte darauf, dass jede Anweisung mit einem Strichpunkt beendet wird, wenn nicht mit einer geschweiften Klammer eine Sequenz begonnen oder beendet wird.

   ➢ Plane die Figur, indem du auf einem karierten Blatt Papier eine Skizze anfertigst.

   Beachte das Klassendiagramm *HausDesNikolaus* sowie das Objektdiagramm *sketch_01_nikolaus1*. Ergänze das Objektdiagramm *h1*.

   - Speichere das Programm unter dem Dateinamen *nikolaus1*.

5. Zusatzaufgabe: Das Vorfahrtszeichen soll als Rechteck gezeichnet und mit Hilfe der Anweisungen `translate()` und `rotate()` gedreht werden (`vorfahrt2`).