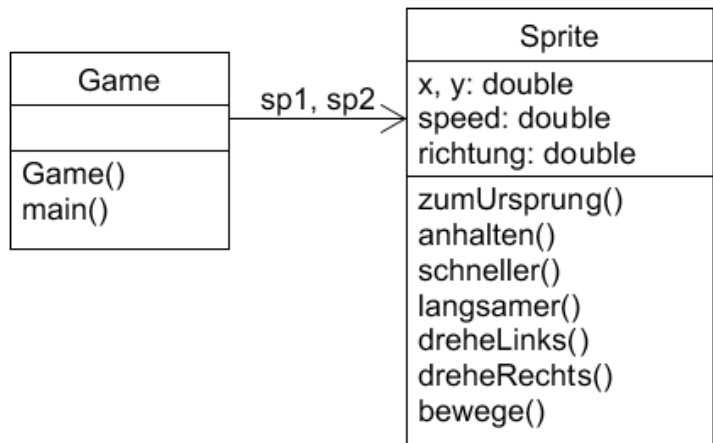


## Aufgabe 1

Erinnere dich an das Sprite-Modell aus der vorigen Unterrichtseinheit:

Sprite-Objekte haben eine Position im Koordinatensystem (x und y), eine Richtung (also einen Winkel), in der sie sich bewegen, und eine Geschwindigkeit (speed), die angibt, wie weit sie sich in einem Bewegungsschritt bewegen.



Die Methode `schneller()` erhöht `speed` um 10, `langsamer()` verringert `speed` entsprechend um 10. `anhalten()` setzt `speed` auf 0.

`dreheLinks()` ändert die Richtung um  $15^\circ$  nach links, `dreheRechts()`  $15^\circ$  nach rechts. Die Richtung  $0^\circ$  entspricht dabei der Richtung der x-Achse.

`bewege()` verschiebt die x- und y-Koordinate in der aktuellen Richtung um die Länge von `speed`. `zumUrsprung()` setzt die Koordinaten beide auf 0.

**Auftrag:** Implementiere die Klasse `Game` in dem ausgeteilten BlueJ-Projekt.

Der **Konstruktor** erzeugt die beiden Sprite-Objekte `sp1` und `sp2`.

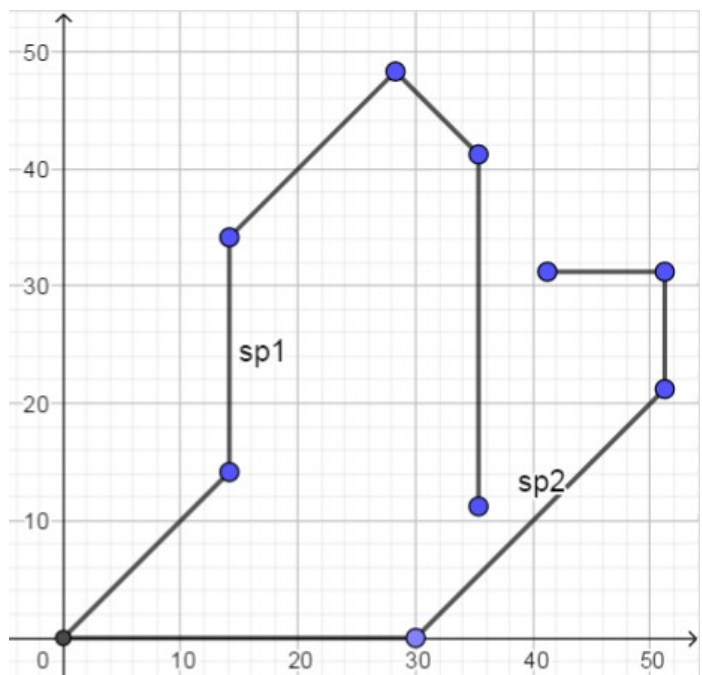
Die **main**-Methode bewegt die beiden Sprites, so dass sie die abgebildeten Wege „ablaufen“.

Beide beginnen ihren Weg im Ursprung. Die Strecken haben jeweils eine Länge von 10, 20 oder 30.

Die Sprites bewegen sich abwechselnd: `sp1` bewegt sich zu seinem ersten Punkt, dann `sp2` zu seinem ersten, dann ist `sp1` wieder dran mit dem zweiten Punkt, usw.

Wenn du dein Projekt übersetzt hast, führe die `main`-Methode mit BlueJ einmal aus.

Prüfe dann mit dem Objektinspektor, ob die x- und y-Werte der Sprite-Objekte mit den Zielpositionen im abgebildeten Koordinatensystem übereinstimmen.



## Aufgabe 2

Eine Spedition verwaltet ihre LKWs mit einer Software.

Für die Software ist relevant, welchen Kilometerstand die LKWs aktuell haben und wie viel Kraftstoff gerade im Tank vorhanden ist.

Für LKWs soll es eine Methode fahren() geben, die den Kilometerstand um 100 km erhöht, und die Kraftstoffmenge entsprechend verringert (der Verbrauch sei hier einheitlich 20 l / 100 km).

Außerdem gibt es eine Methode tanken(), bei der sich die Kraftstoffmenge um 50 l erhöht.

Es soll eine Hauptklasse für die Software geben. Bei Programmstart soll der Konstruktor zwei LKW-Objekte erzeugen. Die main-Methode soll die Software testen: für beide LKWs werden die zur Verfügung stehenden Methoden einige Male aufgerufen.

- a) Stelle das Modell als **Klassendiagramm** dar.
- b) **Implementiere** deine Klassen mit BlueJ.
- c) Zeichne ein **Objektdiagramm** der Situation nach Ausführen der main-Methode.