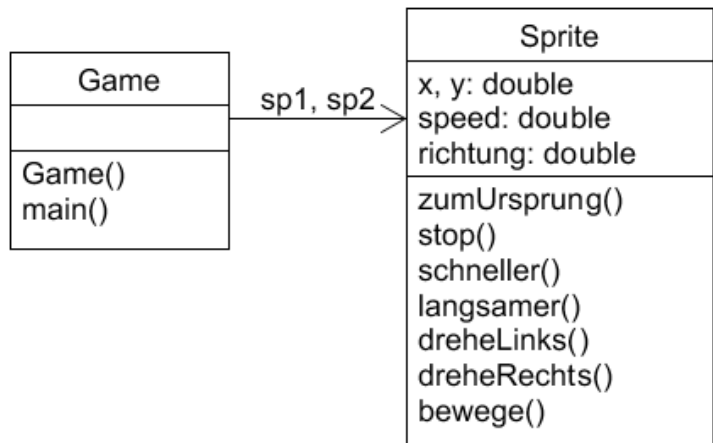


## Aufgabe 1

Erinnere dich an das Sprite-Modell vom letzten Arbeitsblatt:

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. dreheLinks() ändert die Richtung um 15° nach links, dreheRechts() 15° nach rechts.

(Die Richtung 0° entspricht dabei der Richtung der x-Achse.)

bewege() verschiebt die x- und y-Koordinate in der aktuellen Richtung um die Länge von speed.

Ein BlueJ-Projekt mit der Klasse Sprite wird ausgeteilt.

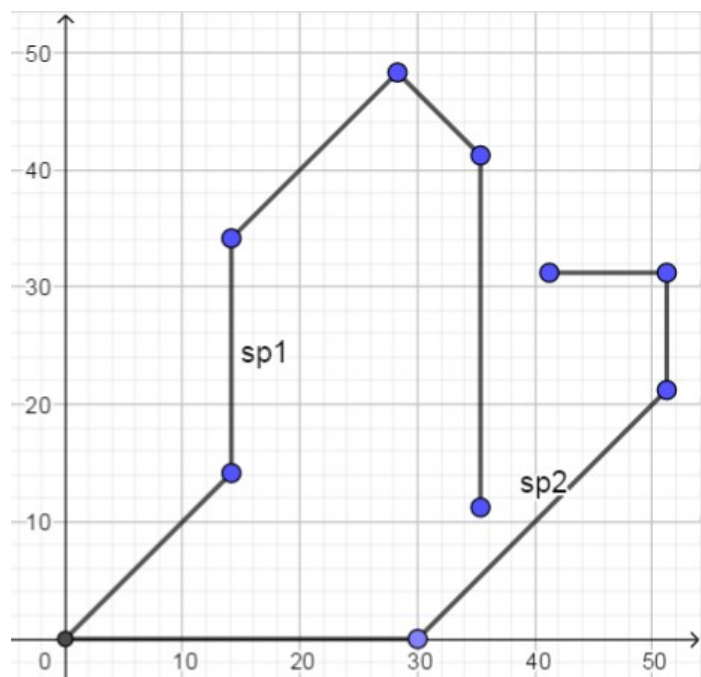
### Auftrag: Implementiere die Klasse Game

Der **Konstruktor** soll die beiden Sprite-Objekte für die Referenzen sp1 und sp2 erzeugen.

Die **main**-Methode soll die beiden Sprites bewegen, so dass sie die abgebildeten Wege „ablaufen“. Beide beginnen ihren Weg im Punkt (0 / 0). Alle Strecken sind Vielfache von 10.

Die Sprites sollen sich dabei abwechselnd bewegen: sp1 bewegt sich zu seinem ersten Punkt, sp2 zu seinem ersten, dann ist sp1 wieder dran mit dem zweiten Punkt, und so weiter.

Wenn du dein Projekt übersetzt hast, rufe die main-Methode einmal auf. Prüfe mit dem Objektinspektor, ob beide Sprites sich an ihrer Zielposition befinden.



## 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.