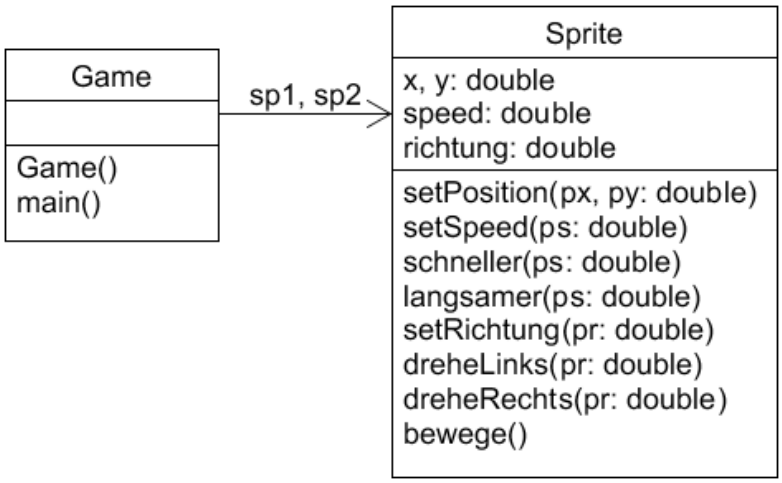
Aufgabe 1



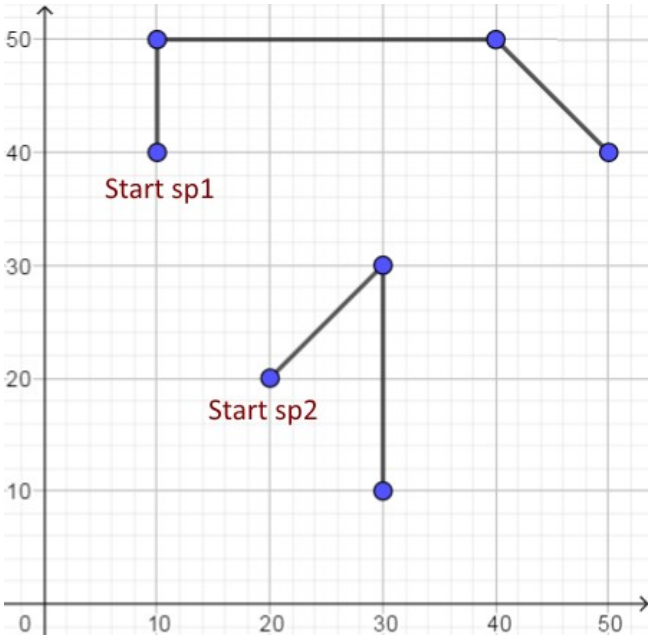
Das Sprite-Modell wird um set-Methoden erweitert:

setPosition (px, py) setzt den Sprite direkt auf die Position, die beim Methodenaufruf für die Parameter px und py eingesetzt wird.

setSpeed() und setRichtung() setzen die Attribute speed und richtung jeweils auf den Wert des Parameters.

Die anderen Methoden arbeiten wie gehabt.

***Auftrag**:*
Implementiere die Klassen Sprite und Game
Verwende dazu die ausgeteilte Vorlage.



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

Beide sollen gehen mit setPosition() auf ihre Startposition. Für den restlichen Weg sollen sie jeweils mit setSpeed() und setRichtung() ihre Attribute einstellen und dann mit bewege() jeweils einen Schritt weitergehen (schneller, langsamer, dreheLinks/Rechts darfst du bei Bedarf auch einsetzen).

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

Tipp:
Die Strecken im 45°-Winkel sind jeweils ca. 14,14 Pixel lang.

Aufgabe 2

Für eine Firma soll eine Software zur Verwaltung der Gehälter der Mitarbeiter entwickelt werden. Für jeden Mitarbeiter soll dessen **Name** und sein **monatliches Gehalt** gespeichert werden, und außerdem, wieviel Gehalt der Mitarbeiter bisher **insgesamt** verdient hat.
Das Gesamtgehalt wächst also jeden Monat um das monatliche Gehalt.

Die Klasse für die Mitarbeiter-Objekte soll Methoden bereitstellen, den Namen und das monatliche Gehalt auf einen bestimmten Wert zu setzen, und außerdem eine Methode, um das monatliche Gehalt zu erhöhen.
Darüberhinaus soll es eine Methode zur Zahlung des monatlichen Gehalts geben. Dabei wird das monatliche Gehalt zum bereits verdienten Gesamtgehalt dazuaddiert.

Die Hauptklasse soll zwei Mitarbeiter-Objekte erzeugen.
Die main-Methode setzt Namen und monatliche Gehälter für beide Mitarbeiter. Dann simuliert sie den Ablauf einiger Monate, in denen jeweils das Gehalt ausgezahlt wird.
Zwischendurch soll sich das Gehalt auch mal erhöhen.

a) Stelle das Modell als **Klassendiagramm** dar.
Überlege dabei, für welche Methoden **Parameter** sinnvoll bzw. notwendig sind.

b) **Implementiere** deine Klassen mit BlueJ.

c) Zeichne ein **Objektdiagramm** der Situation nach Ausführen der main-Methode.

Aufgabe 3

Erweitere das Klassendiagramm und den Quellcode aus Aufgabe 2 um die Lohnsteuer:

Da der **Steuersatz** nicht immer gleich ist, soll der Steuersatz für jeden Mitarbeiter gespeichert werden, in Prozent: Ein Wert von 10 entspricht also 10%.
Das Attribut für das monatliche Gehalt soll dem Bruttogehalt entsprechen, also was die Firma insgesamt zahlen muss. Für das monatliche Nettogehalt braucht es kein Attribut, da es sich aus dem Bruttogehalt ergibt, indem die Steuern abgezogen werden:

$$\text{Nettogehalt} = \text{Bruttogehalt} - (\text{Bruttogehalt} \cdot \text{Steuersatz} / 100)$$

Zusätzlich zu dem Attribut für das bisher ausgezahlte Gesamt (Brutto-) Gehalt soll es auch je ein Attribut für das **bisher ausgezahlte Nettobehalt** und die **bisher gezahlten Steuern** geben.

Für den Steuersatz braucht es eine weitere set-Methode. Für die beiden anderen neuen Attribute wird keine zusätzliche Methode benötigt. Stattdessen wird die Methode zur monatlichen Zahlung des Gehalts erweitert. Sie erhöht dann auch das Gesamt-Nettogehalt und die Gesamt-Steuern.

In der main-Methode brauchst du nur zu Beginn für die Mitarbeiter einen Steuersatz zu setzen.