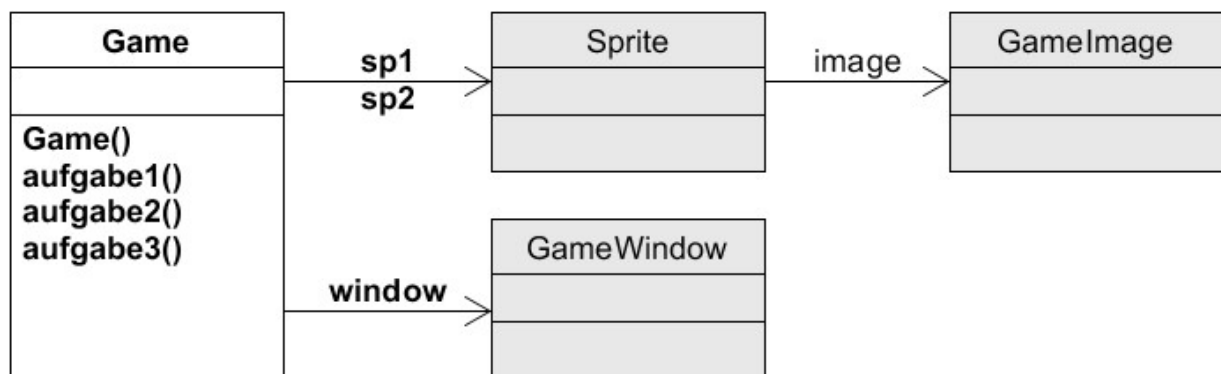


## Hinweise zu den Aufgaben

Verwende die ausgeteilte Vorlage, die bereits die Klasse `Sprite` enthält. Die Klassen `GameWindow` und `GameImage` kannst du über einen Import verwenden.

→ Du brauchst **nur die Klasse `Game`** zu implementieren.

Statt der sonst üblichen `main`-Methode enthält `Game` die Methoden `aufgabe1()`, `aufgabe2()` usw. Diese übernehmen jeweils die Aufgabe, die sonst die `main`-Methode hat.



Zur Implementierung nutze die folgende Dokumentation:

<code>import pm.gamewindow.*;</code>	Importiert das <code>GameWindow</code> -Package in deine Datei.
--------------------------------------	-----------------------------------------------------------------

## Die Klasse `GameWindow`

<b>Konstruktor</b> (Beispiel): <code>window = new GameWindow(50, 50, 800, 600, "Beispiel");</code> Erzeugt ein <code>GameWindow</code> an der Bildschirm-Position <code>x=50 y=50</code> mit Breite 800 und Höhe 600.	
<code>void clear()</code>	Übermalt das Fenster in weiß.
<code>void paintFrame()</code>	Überträgt das bisher gezeichnete an den Bildschirm.

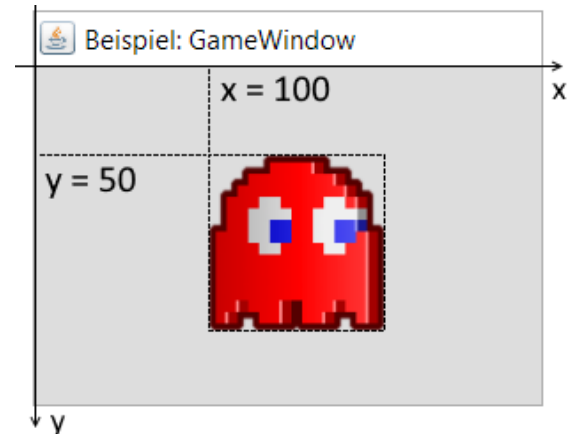
## Die Klasse Sprite

**Konstruktor** (Beispiel): `sp = new Sprite("../images/ghost.png");`  
 Bei der Erzeugung des Sprites wird ein `GameImage`-Objekt erzeugt.  
 Dabei wird ein Bild aus der angegebenen Bilddatei eingelesen.  
 Dieses Bild wird dann mithilfe der `Sprite`-Methode `draw()` gezeichnet (siehe weiter unten).

`void setPosition(`  
     `double px, double py)`

Setzt die Koordinaten des Sprites bezogen auf das `GameWindow`, in dem gezeichnet wird.

- Koordinatenursprung ist die linke obere Ecke des `GameWindows`.
- `px / py` beziehen sich auf die linke obere Ecke des Bildes
- Das Bild kann außerhalb des Fensters liegen (es wird nur der sichtbare Teil gezeichnet)

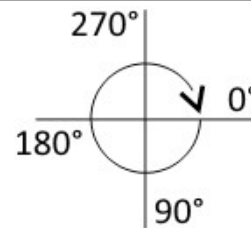


`void setRichtung(double pr)`

Setzt die Richtung (in Grad), in die der Sprite sich mit `bewege()` bewegt.

0° entspricht der x-Achse.

Auch negative Werte sind möglich.



`void dreheLinks(double pr)`

`void dreheRechts(double pr)`

Dreht die aktuelle Richtung um `pr` Grad nach links bzw. nach rechts.

`void setSpeed(double ps)`

Setzt die Distanz (in Pixeln), die der Sprite sich mit einer Ausführung von `bewege()` bewegt.

`void bewege()`

Ändert die aktuelle Position `x/y` des Sprites um `speed` Pixel in der aktuellen Richtung.

`void draw(GameWindow pwin)`

Beispiel: `sp1.draw(window);`

Zeichnet das Bild des Sprites an der Position `x / y`. Das `GameWindow`-Objekt wird als Parameter übergeben.

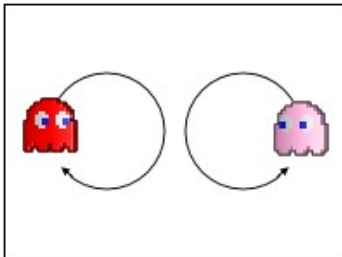
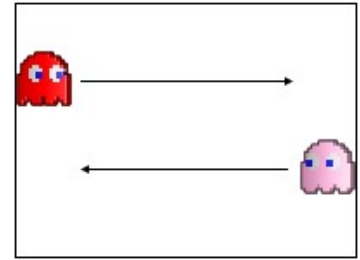
Bearbeite die Klasse Game:

## Aufgabe 1

Implementiere die Methode **aufgabe1()**.

Die Sprites beginnen am linken bzw. rechten Bildrand und bewegen sich gleichzeitig im Zeitraum von 2 Sekunden einmal von links nach rechts (bzw. umgekehrt) über das Fenster.

Nutze eine **while-Schleife** entsprechend der Folien.



## Aufgabe 2

Implementiere die Methode **aufgabe2()**.

Die Sprites beginnen links bzw. rechts und bewegen sich im Kreis, so dass sie wieder am Ausgangspunkt ankommen.

Die Sprites bewegen sich gleichzeitig und symmetrisch zueinander.

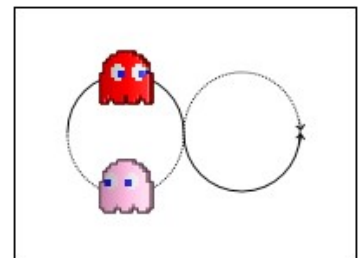
## Aufgabe 3

Implementiere die Methode **aufgabe3()**.

Beide Sprites beginnen links an der gleichen Position.

Beide beschreiben gleichzeitig eine Schlangenlinie (der eine beginnt nach oben, der andere nach unten).

Hilfestellung: teile die Bewegung in mehrere Halbkreise ein. Für jeden Halbkreis verwende eine while-Schleife.



## Aufgabe 4

Schreibe zur Klasse Game und zu ihren Methoden je einen **Kommentar**.