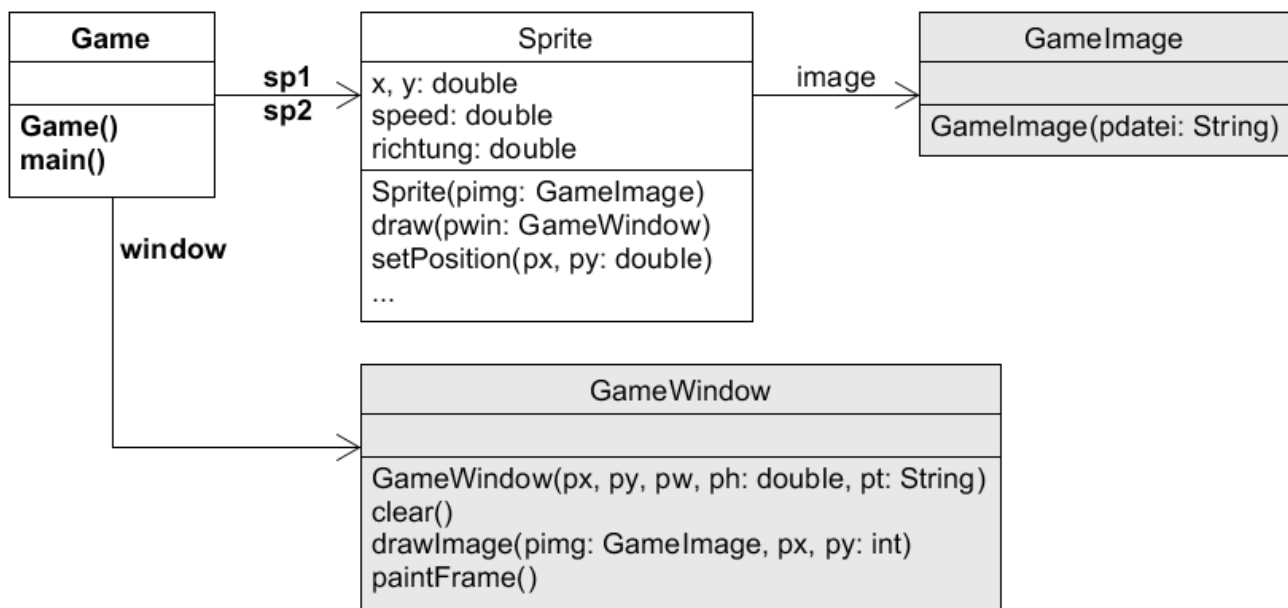


Aufgabe 1

Als Vorbereitung auf die Programmieraufgabe lies die folgende Dokumentation:

Dein BlueJ-Projekt verwendet die im folgenden abgebildeten Klassen. Im Projekt selbst sind nur die Klassen **Game** (Hauptklasse) und **Sprite** enthalten. **GameImage** und **GameWindow** werden mithilfe des „Imports“ einer Bibliothek eingebunden. Bibliotheken enthalten Klassen, die bereits fertig implementiert sind. Die Klasse **Sprite** ist in deiner Programmervorlage auch schon implementiert, du brauchst also nur die Hauptklasse (fettgedruckt) zu programmieren.



Jeder **Sprite** hat jetzt ein **GameImage**-Objekt, das ein Bild aus einer Bilddatei darstellt. Außerdem ist die Methode `draw()` hinzugefügt worden, mit der dieses **GameImage** an der aktuellen Position des Sprites (`x / y`) auf ein Fenster gezeichnet werden kann.

Das **Hauptobjekt** erzeugt zusätzlich zu den Sprites ein **GameWindow**-Objekt. Dieses stellt ein Fenster dar, auf dem Bilder gezeichnet werden können.

Die Klassen **GameWindow** und **GameImage** sind Teil der Bibliothek „`pm.gamewindow`“. Die Bibliothek ist in einer sogenannten Jar-Datei enthalten, die auf den Schulrechnern schon installiert ist.

Jede Klasse deines BlueJ-Projekts, in der die Klassen **GameWindow** und / oder **GameImage** verwendet werden, fügt man einen **Import-Befehl** hinzu. Er steht am Anfang der Datei, so wie rechts abgebildet.

```
import pm.gamewindow.*;
```

```
class Game
{
    Sprite sp1, sp2;
    GameWindow window;

    Game ()
    {
```

```
import pm.gamewindow.*;
```

```
class Sprite
{
    double x, y;
    double richtung;
    double speed;
    GameImage image;
```

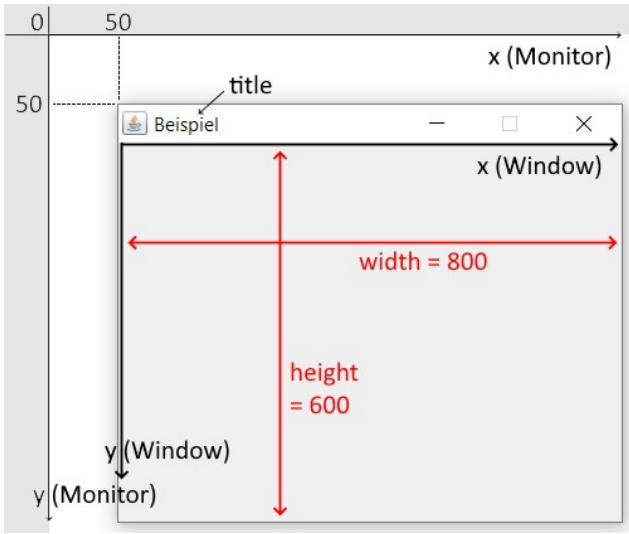
Die Klasse GameImage

Für den Anfang brauchst du nicht direkt mit GameImage-Objekten zu arbeiten. Sie werden von deinen Sprite-Objekten erzeugt und gezeichnet. Später kannst du GameImages auch unabhängig von Sprites verwenden, z.B. um ein Hintergrundbild zu zeichnen. Daher sei hier der Konstruktor der Klasse GameImage gezeigt:

Konstruktor (Beispiel): `img = new GameImage("./images/bild.png");`

Die Klasse GameWindow

Die Klasse GameWindow bietet viele Methoden, die wir an dieser Stelle jedoch noch nicht alle brauchen. Zur Lösung der Aufgaben sind nur die folgenden erforderlich:

<p>Konstruktor (Beispiel):</p> <pre>window = new GameWindow(50, 50, 800, 600, "Beispiel");</pre> <p>Erzeugt ein Fenster („Window“) mit Breite 800 und Höhe 600 mit dem Text „Beispiel“ in der Titelleiste.</p> <p>Die linke obere Ecke des GameWindows liegt an Position $x=50$ $y=50$ auf dem Monitor. Der Koordinatenursprung des Monitors ist in der linken, oberen Ecke, und seine y-Achse wird nach unten positiv!</p> <p>Ebenso liegt der Koordinatenursprung des GameWindows in seiner linken, oberen Ecke und seine y-Achse wird nach unten positiv.</p>	
<pre>void clear()</pre>	Übermalt das GameWindow in weiß.
<pre>void paintFrame()</pre>	Überträgt alles, was bisher auf das GameWindow gezeichnet wurde (z.B. Sprites) auf den Bildschirm. Vor Aufruf dieser Methode sieht man das Gezeichnete nicht!

Die Klasse Sprite

Die Klasse Sprite wurde angepasst auf die Bewegung in einem GameWindow und das Zeichnen von Bildern. Auch hier sind nur die für die Aufgabe benötigten Methoden erläutert.

Konstruktor (Beispiel): `sp = new Sprite("./images/ghost.png");` *)

Bei der Erzeugung des Sprites wird ein GameImage-Objekt erzeugt.

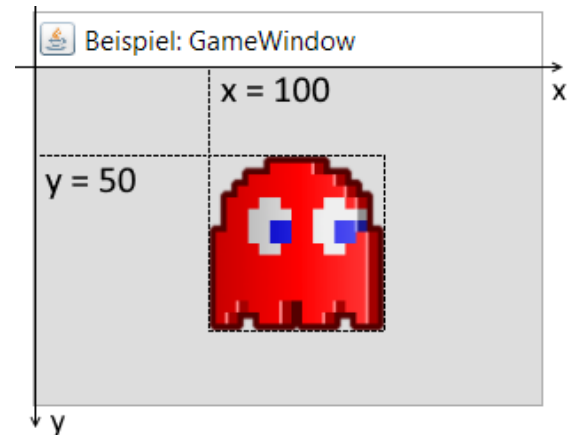
Dabei wird ein Bild aus der angegebenen Bilddatei eingelesen.

Dieses Bild wird dann mithilfe der Sprite-Methode `draw()` gezeichnet.

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

Setzt die Koordinaten des Sprites in dem GameWindow, in dem er gezeichnet wird.

- Koordinatenursprung ist die linke obere Ecke des GameWindows.
- `px / py` beziehen sich auf die linke obere Ecke des Bildes
- Die Koordinaten dürfen auch außerhalb des Fensters liegen (es wird jeweils nur der sichtbare Teil des Bildes gezeichnet).



```
void draw(GameWindow pwin)
```

Beispiel:

```
sp1.draw(window);
```

Zeichnet das Bild des Sprites an der Position `x / y`, die zuvor mit `setPosition()` gesetzt wurde. Das GameWindow-Objekt muss als Parameter übergeben werden.

*) Hinweis:

Für Sprites solltest du Bilddateien im **PNG-Format** zu benutzen. Sprites-Grafiken haben meist transparente Bereiche: Pacman ist z.B. rund. Die Ecken der Bilddatei sind durchsichtig, denn sonst würde man auf einem farbigen Hintergrund die weißen Ecken der Bilddatei sehen. Das häufig verwendete JPG-Format unterstützt jedoch keine Transparenz.

Aufgabe 2

- a) Verwende die ausgeteilte BlueJ-Vorlage.
Die Klasse Sprite ist bereits fertig (du kannst dir die Änderungen aber anschauen).
Sorge dafür, dass deine Klasse Game die Bibliothek „pm.gamewindow“ **importiert**.
Gib die **Deklaration** der Sprite-Objekte sp1 und sp2 sowie des GameWindow-Objekts an (schau dir ggf. noch einmal das Klassendiagramm auf Seite 1 an).
- b) Implementiere den **Konstruktor** der Klasse Game.
Er erzeugt die beiden Sprite-Objekte sowie das GameWindow-Objekt.
Das GameWindow soll eine Größe von 800 x 600 Pixeln haben.
Für die Sprites nutze zwei der Bilddateien im Unterordner „images“ deines BlueJ-Projekts.
Achte darauf, die Dateinamen exakt abzutippen.
- c) Die **main-Methode** soll setzt die Sprites an zwei verschiedene Positionen innerhalb des GameWindows. Dann zeichnet sie beide mit der Methode draw() auf das GameWindow und ruft paintFrame() auf, um das gezeichnete an den Bildschirm zu übertragen.
- d) **Erweitere** dein Programm um weitere Sprites mit unterschiedlichen Grafiken und zeichne sie an verschiedenen Stellen auf das GameWindow.