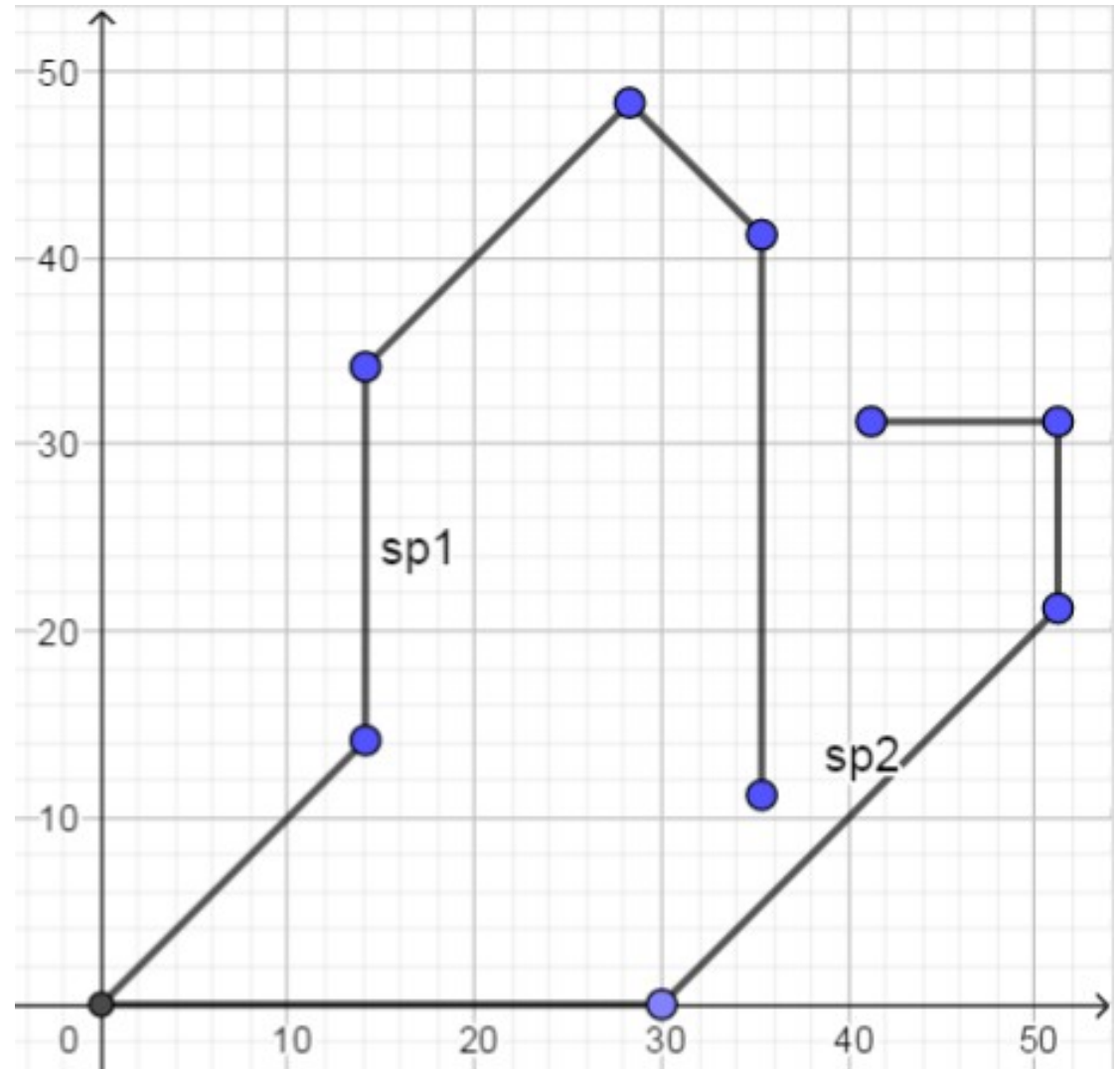




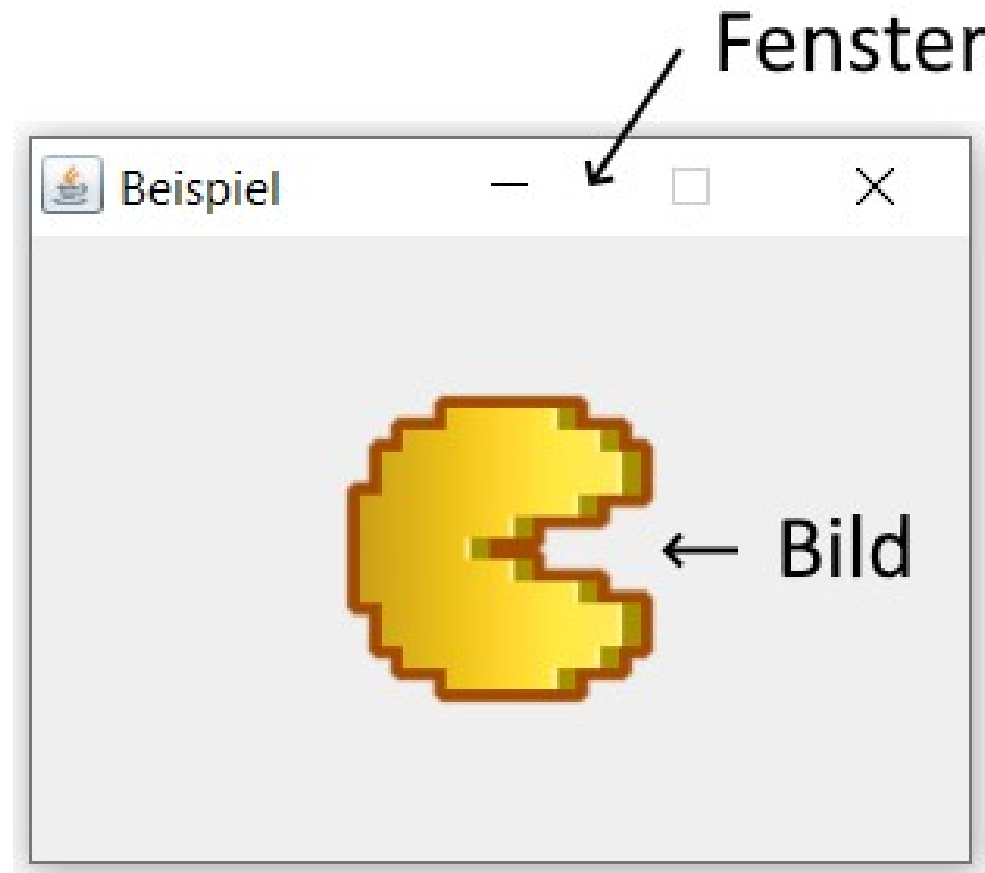
**Spiele programmieren
mit der
GameWindow
-Bibliothek**

Bisher...

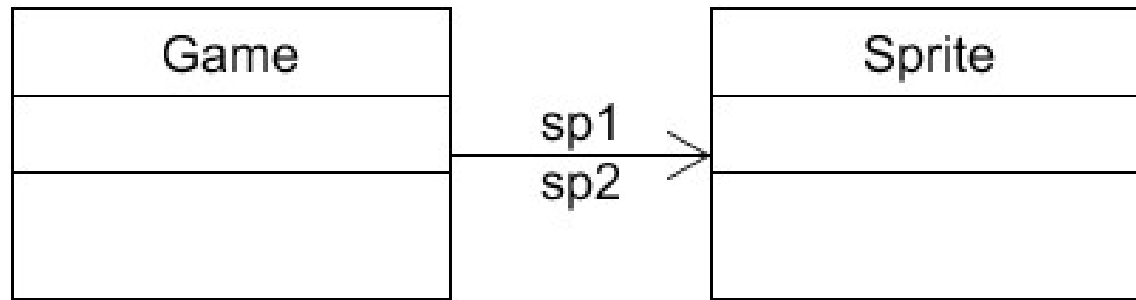
„Sprites“ bewegen
sich als Punkte
in einem
Koordinatensystem



Neu: Sprites mit Grafik



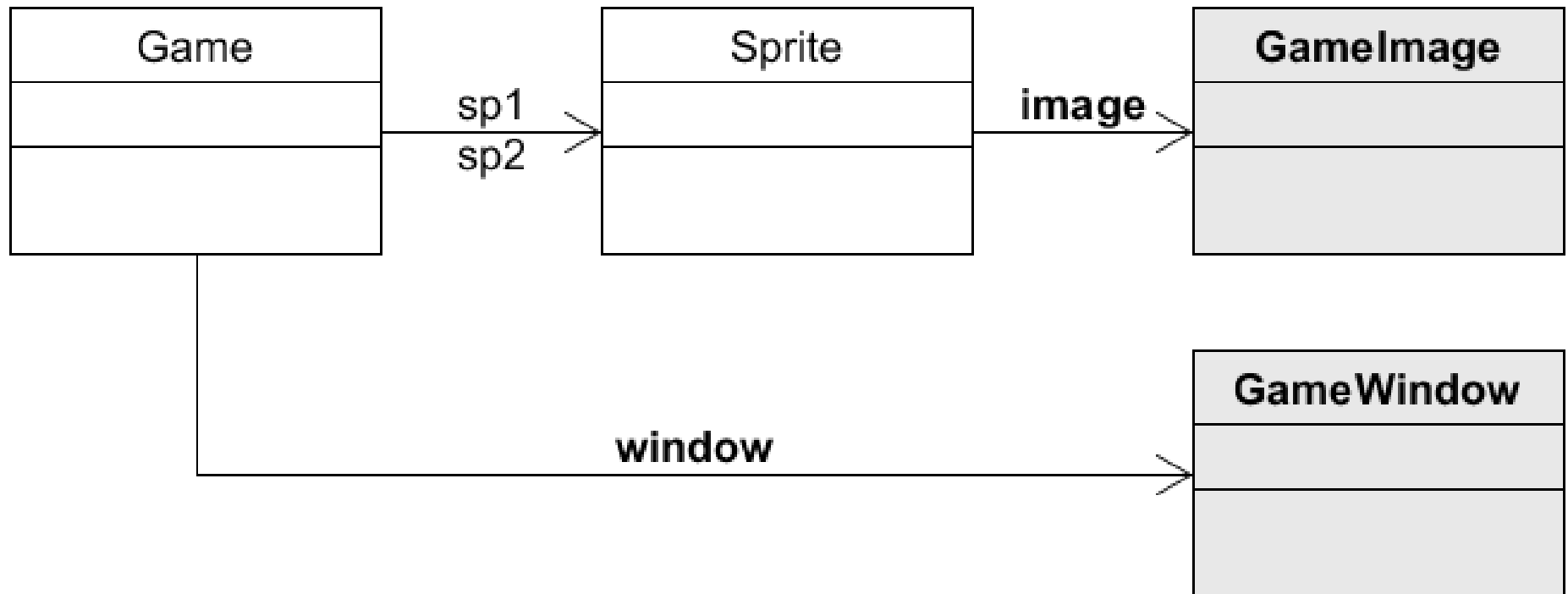
Bisher...



Klasse Game
Klasse Sprite

→ **Hauptklasse**
→ bewegte **Figuren**

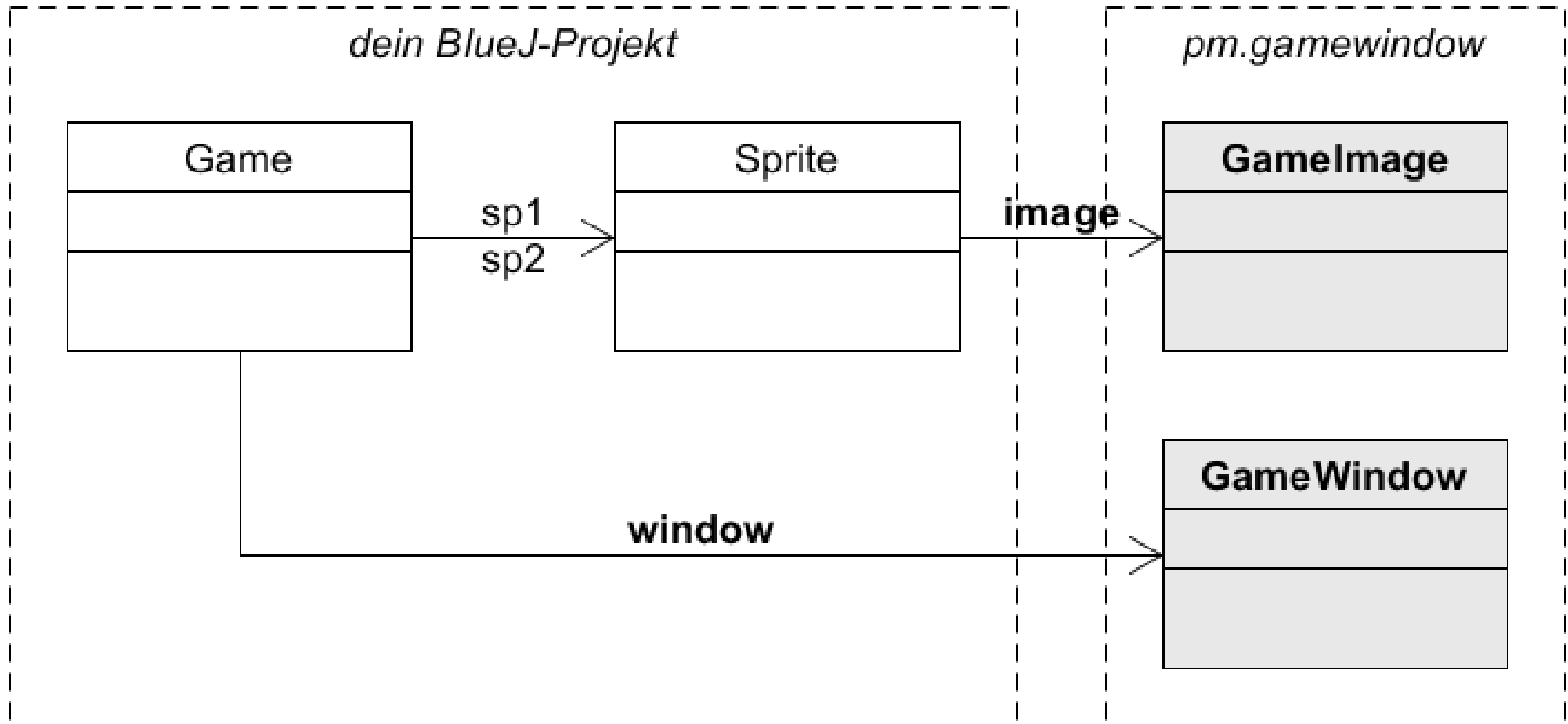
Neu: Sprites mit Grafik



Klasse GameImage → **Bild** (aus Bilddatei geladen)

Klasse GameWindow → **Fenster** zum Zeichnen

Neu: Sprites mit Grafik



GameImage und GameWindow sind
Teil der **Bibliothek** „pm.gamewindow“
→ schon fertig programmiert

Klassen importieren



Dein BlueJ-Projekt enthält weiterhin nur die Klassen, die du selbst programmierst.

Klassen importieren

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

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

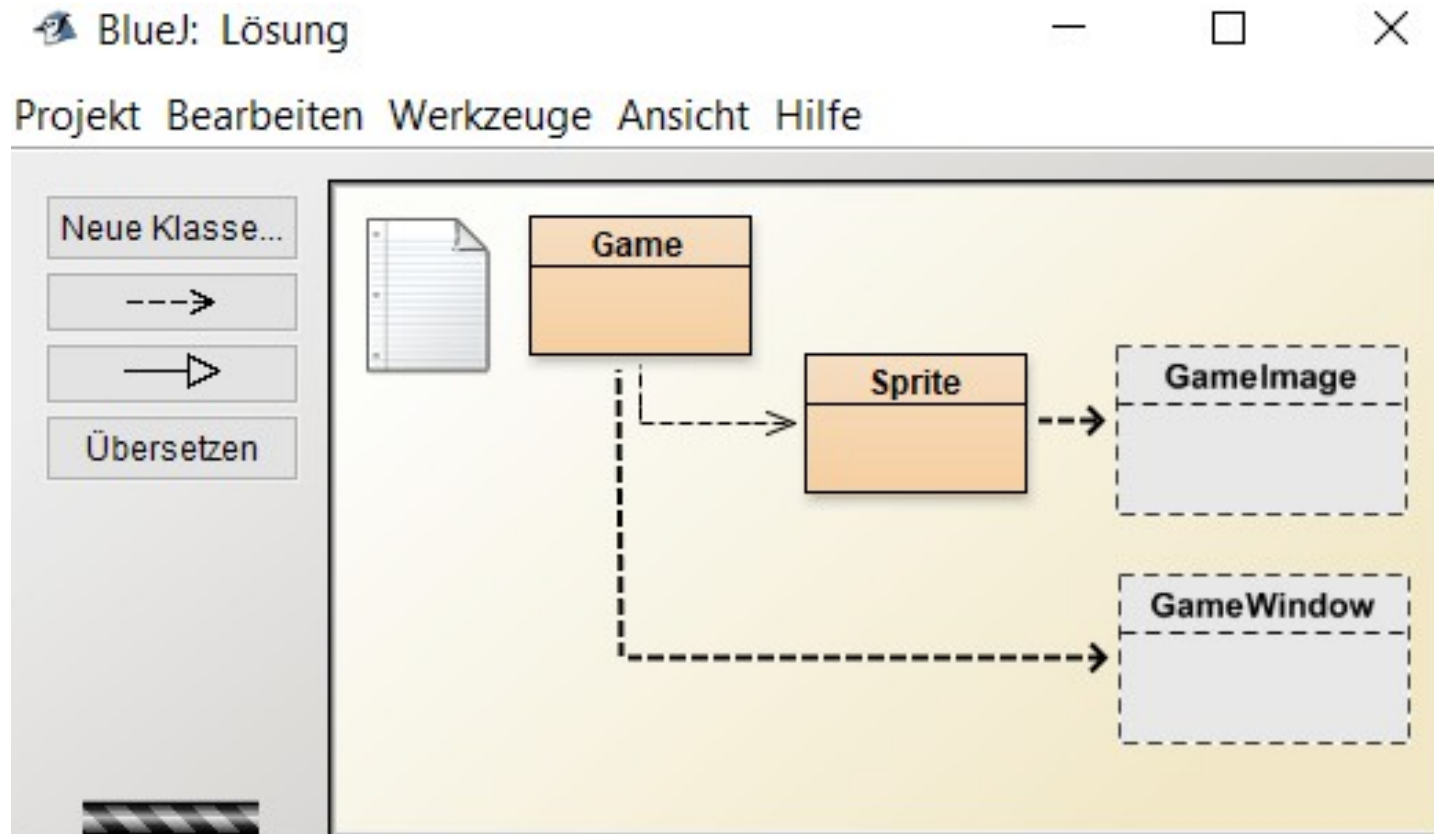
    Game()
    {
```

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

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

Mit der „import“-Anweisung kannst du
alle Klassen einer Bibliothek
im Quellcode deiner Klassen benutzen.

Klassen importieren

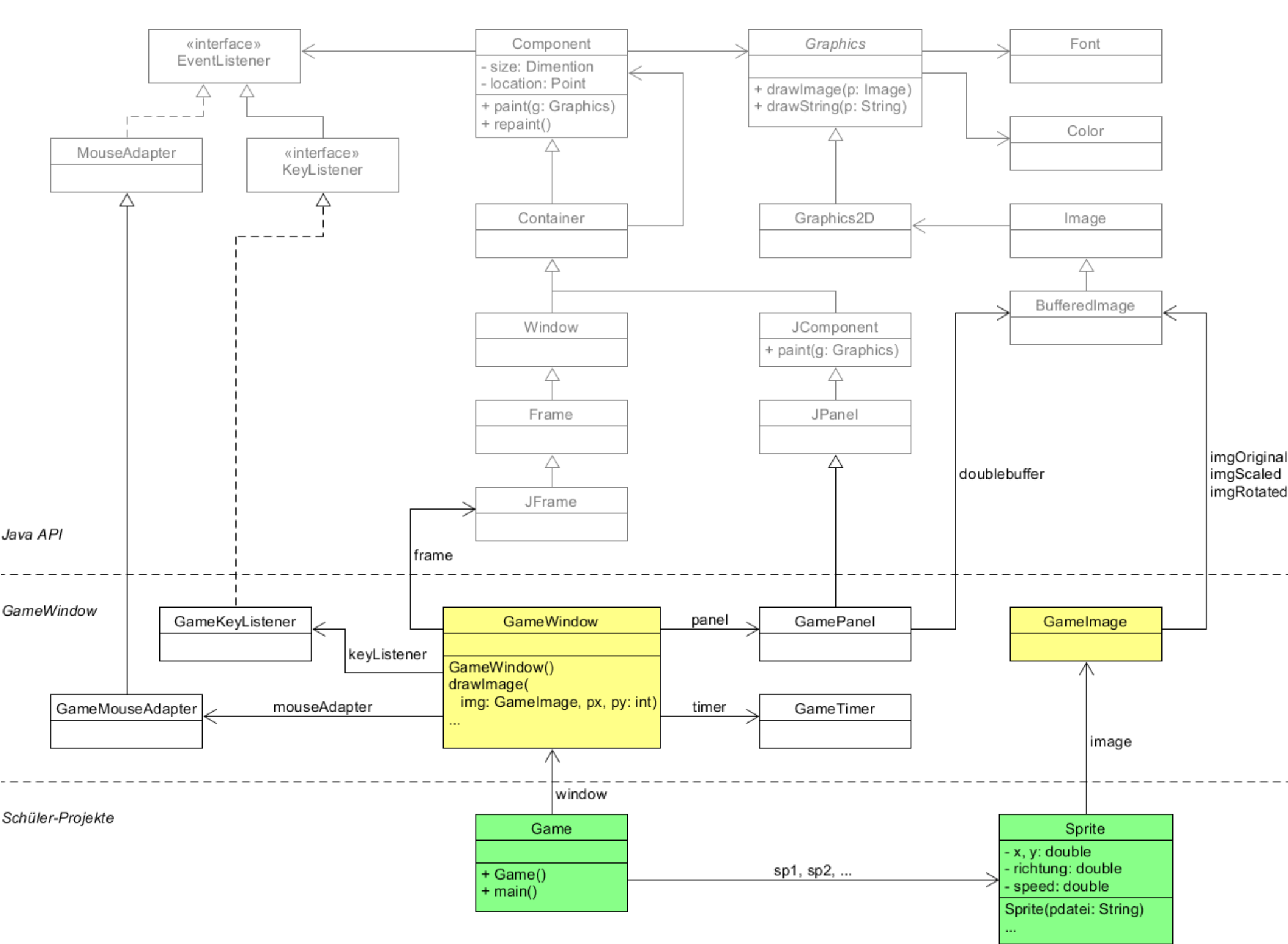


Die importierten Klassen musst du dir „dazudenken“ - sie werden nicht in BlueJ angezeigt!

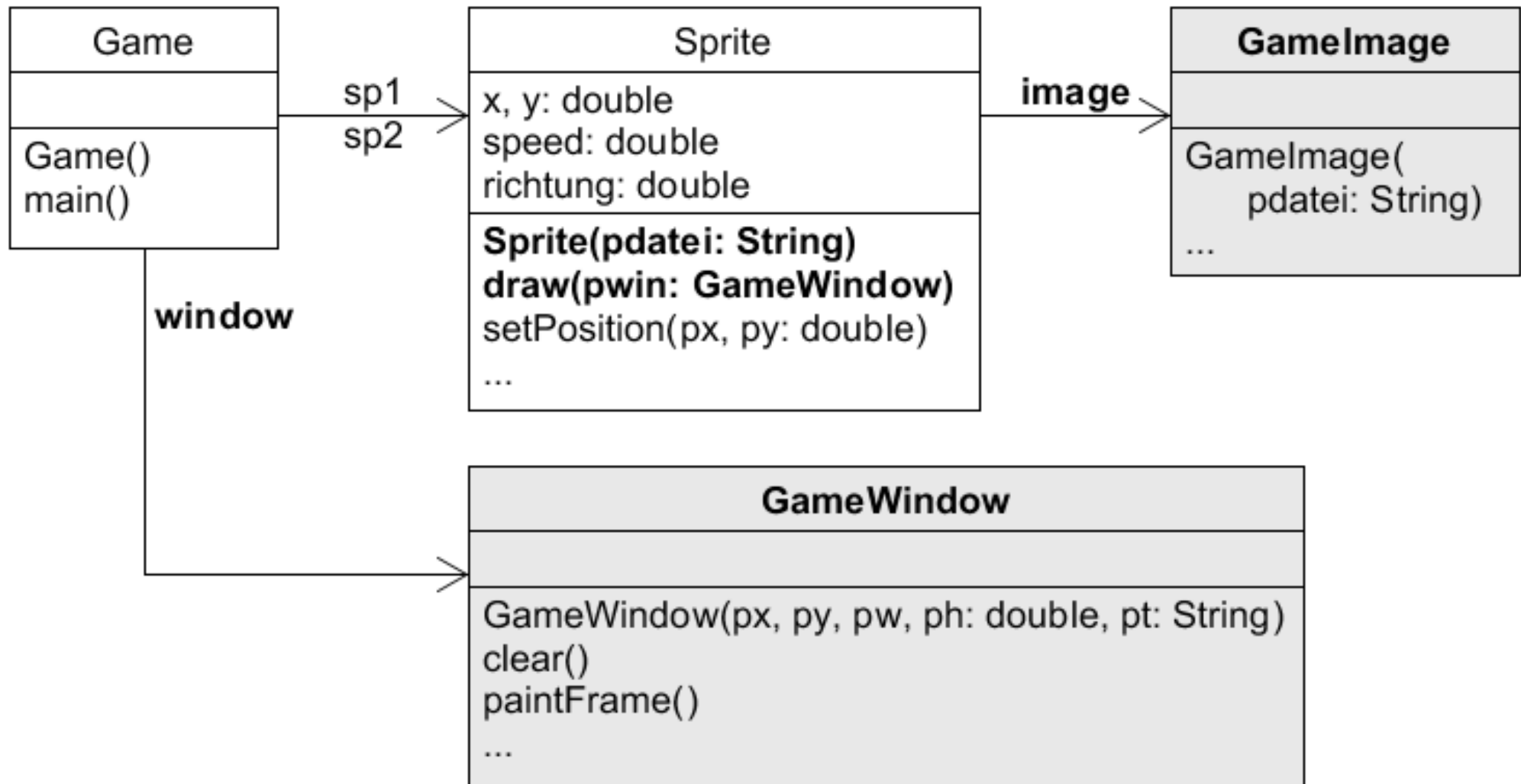
Bibliotheken ...

... ermöglichen **einfachen Zugriff** auf viele, bereits existierende Klassen.

→ ein Beispiel, wie OOP **Arbeitsteilung** und **Wiederverwendung** von Quellcode ermöglicht



Projekt mit GameWindow im Detail



Klasse GameWindow

Konstruktor:

```
GameWindow(  
    int px, int py, int pw, int ph, String ptitle )
```

Beispiel:

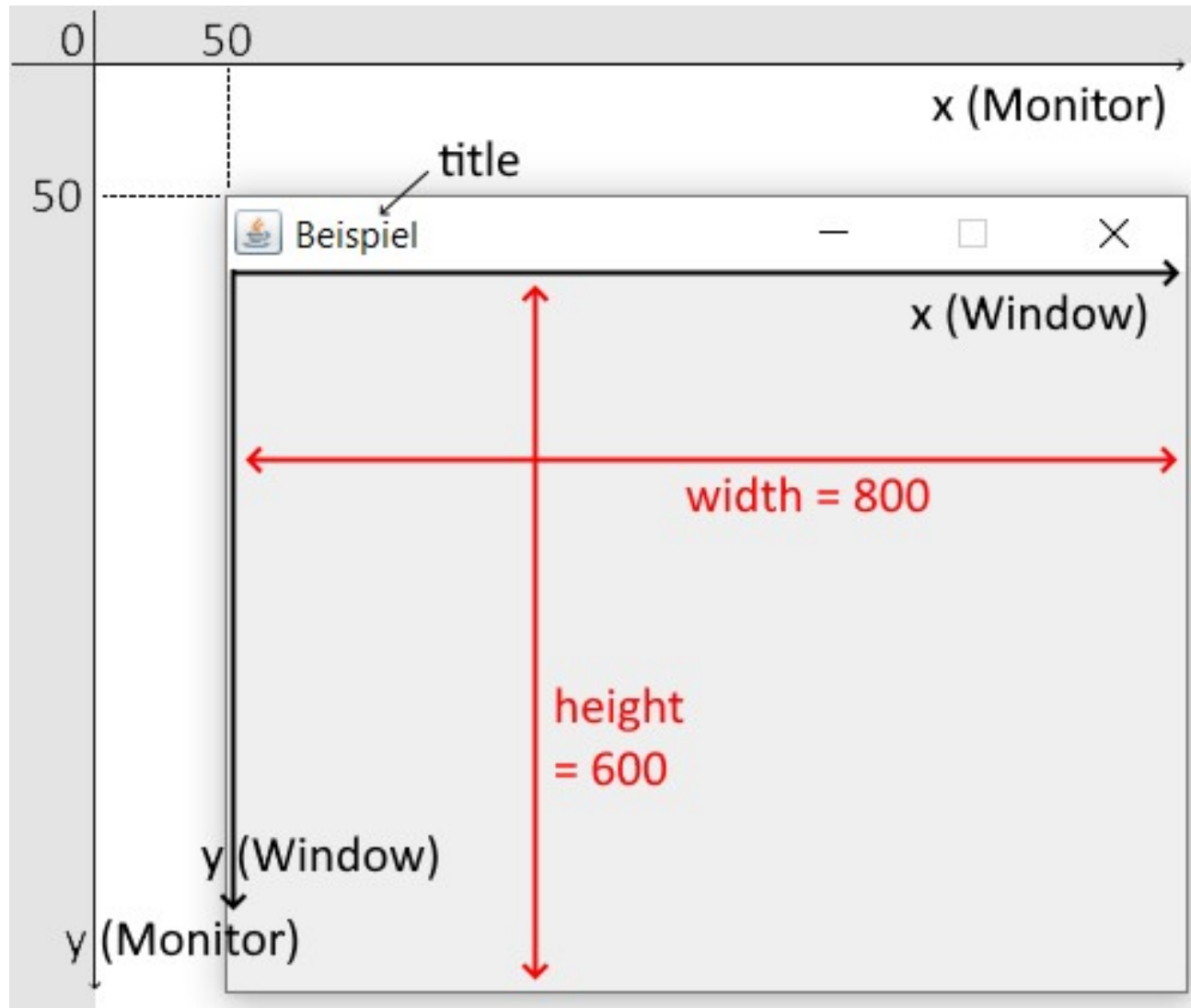
```
window = new GameWindow(  
    50, 50, 800, 600, "Beispiel" );
```

→ $x = 50$, $y = 50$

→ Breite 800, Höhe 600

→ Titel „Beispiel“

Klasse GameWindow



... new GameWindow(50, 50, 800, 600, "Beispiel");

Klasse Sprite

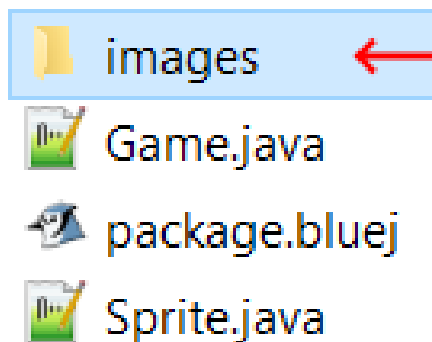
Konstruktor:

`Sprite(String pdatei)`

Beispiel:

```
sp1 = new Sprite("./images/pacman.png");
```

→ lädt Bilddatei für den Sprite



← Unterordner des BlueJ-Projekts
enthält die Bilddatei

Klasse Sprite

Methode zum Zeichnen:

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

Beispiel:

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

→ zeichnet das Bild des Sprites
an der aktuellen Position (x / y)
auf das GameWindow „window“.

Vollständiges Beispiel

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

    Game()
    {
        sp1 = new Sprite("images/ghost1.png");
        sp2 = new Sprite("images/ghost2.png");

        window = new GameWindow
            (50, 50, 800, 600, "Beispiel");
    }
}
```

Hauptklasse

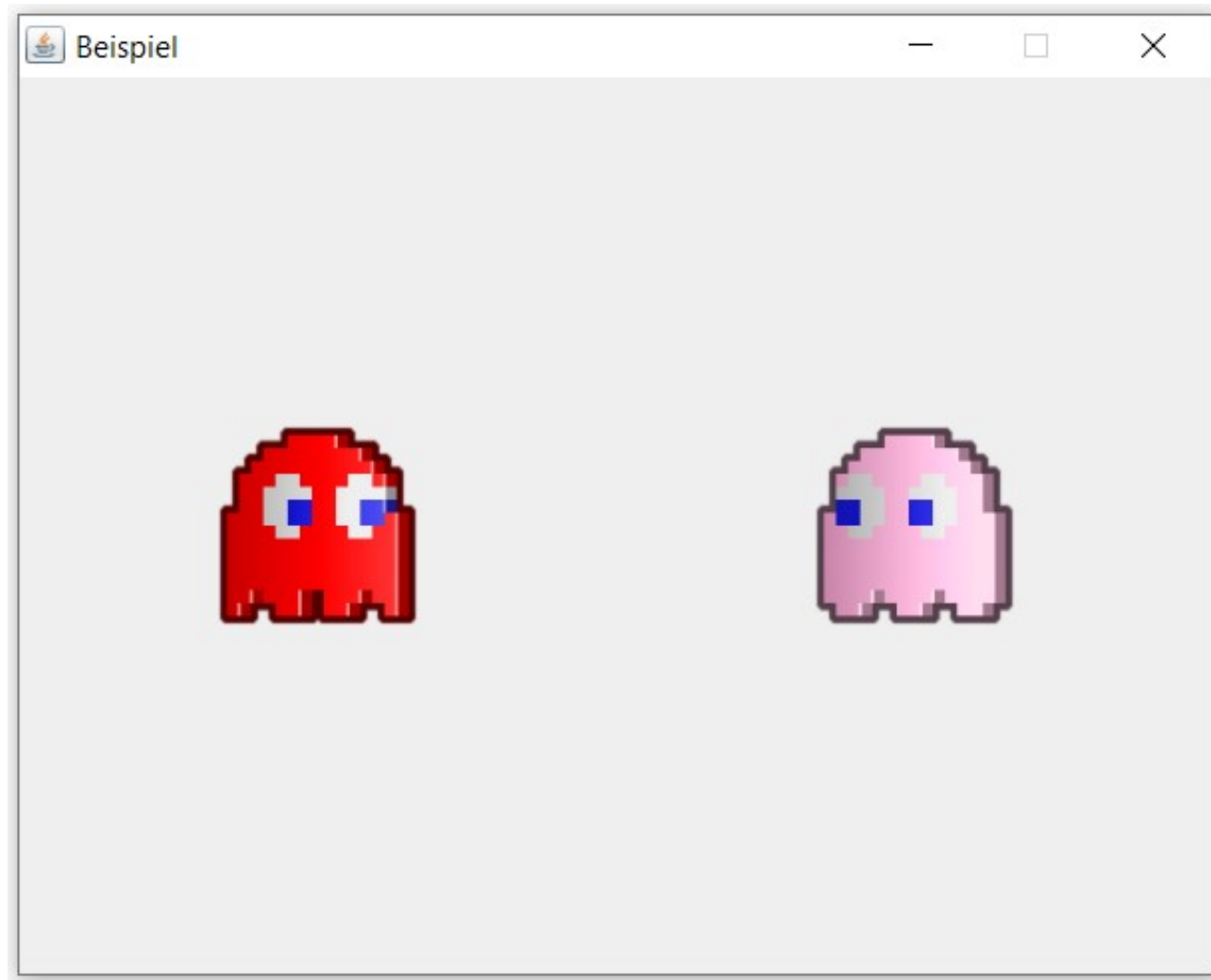
```
void main()  
{  
    sp1.setPosition(100, 250);  
    sp1.setPosition(600, 250);  
  
    sp1.draw(window);  
    sp2.draw(window);  
  
    window.paintFrame();  
}
```

Hauptklasse

```
void main()  
{  
    sp1.setPosition(100, 250);  
    sp1.setPosition(600, 250);  
  
    sp1.draw(window);  
    sp2.draw(window);  
  
    window.paintFrame();  
}
```

Erläuterung: Zeichnet die Bilder der Sprites an den gesetzten Positionen auf das GameWindow. paintFrame() überträgt an den Bildschirm.

Ergebnis



Bibliothek → BlueJ

Bibliotheken:
in **Jar-Dateien**

Für BlueJ:
Jar-Datei kopieren nach



`C:\Programme\BlueJ\lib\userlib\`

→ Import in allen BlueJ-Projekten möglich

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
Freigegeben unter CC BY-NC-SA 4.0, März 2021



Grafiken:

- Pacman-Grafiken: strategywiki.org, Lizenz CC BY-SA 3.0