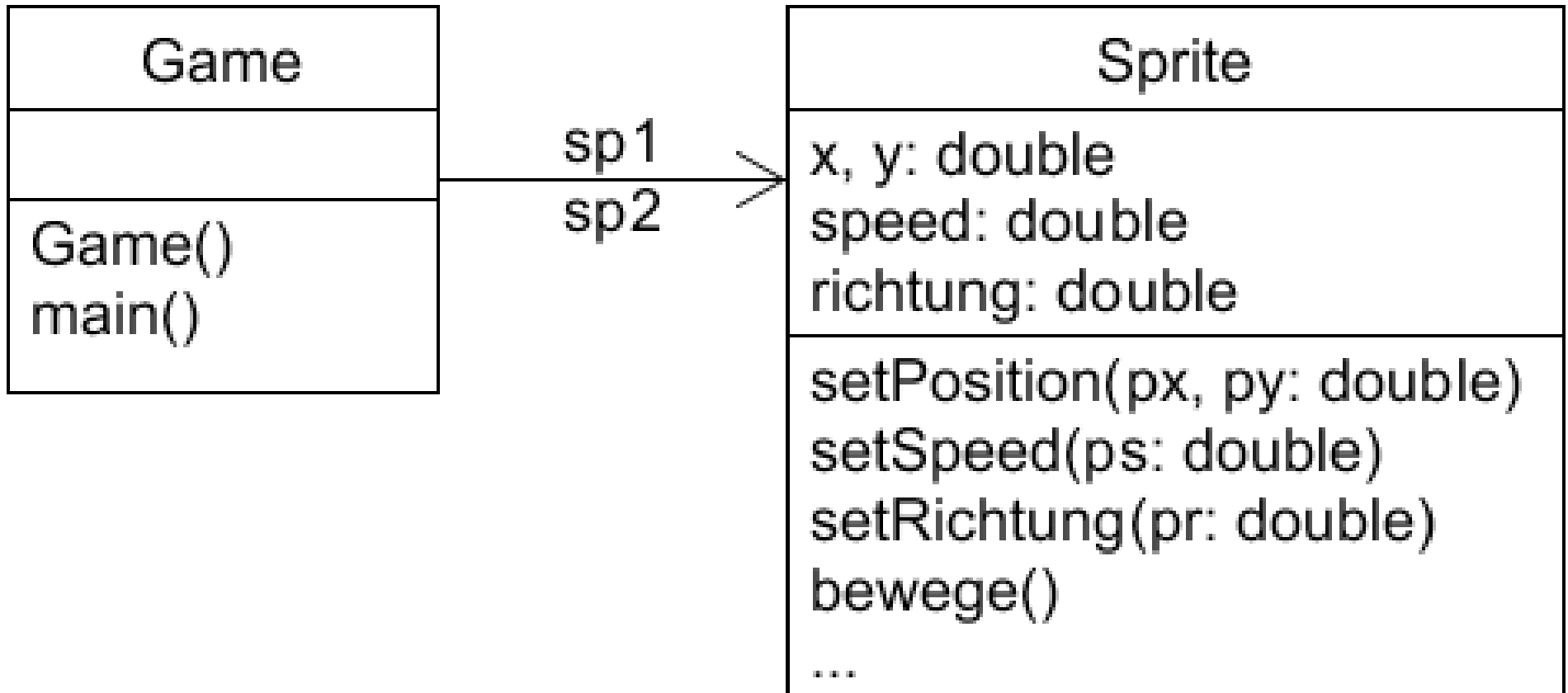




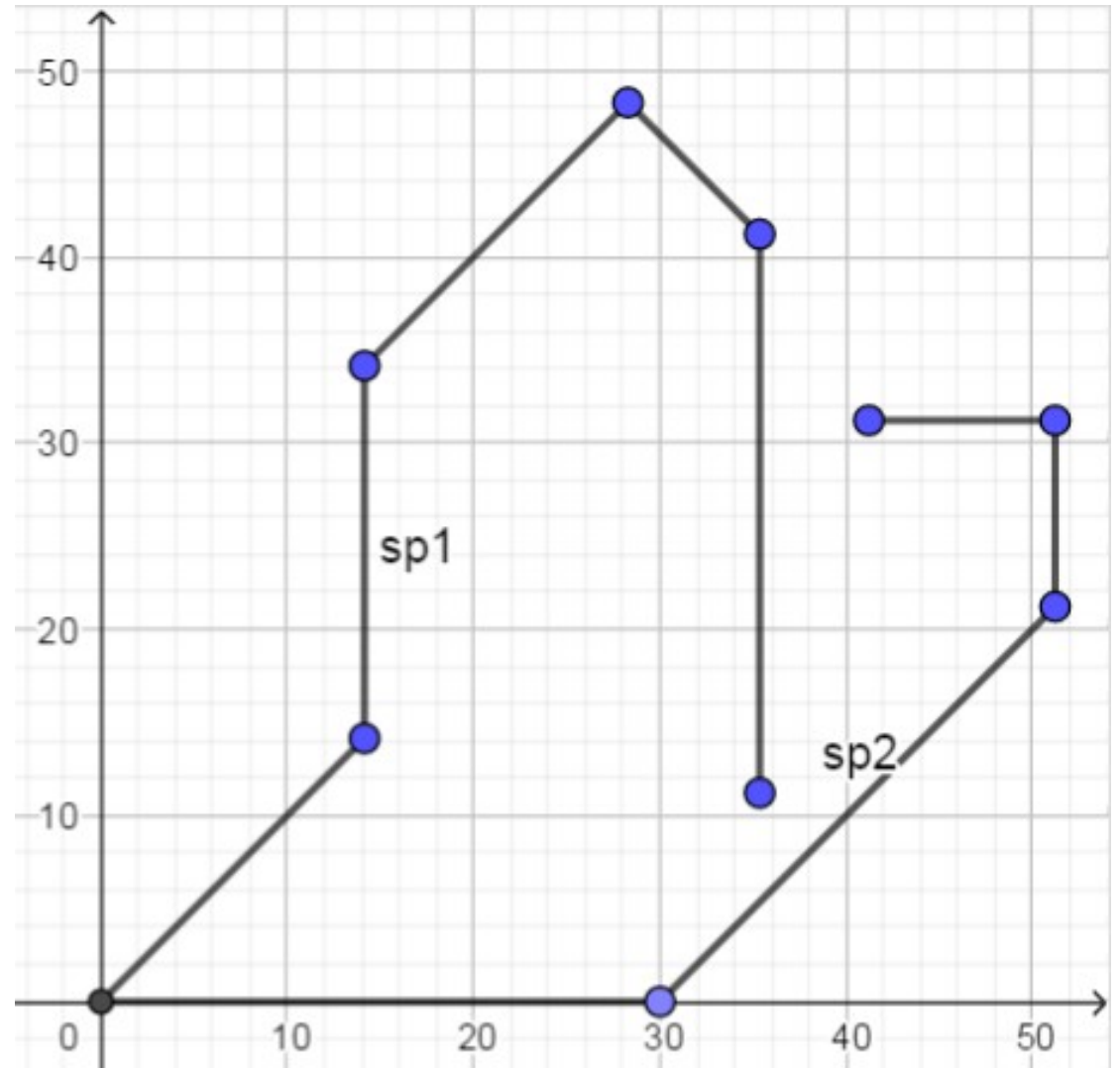
**Spiele programmieren
mit der
GameWindow
-Bibliothek**

Bisher...

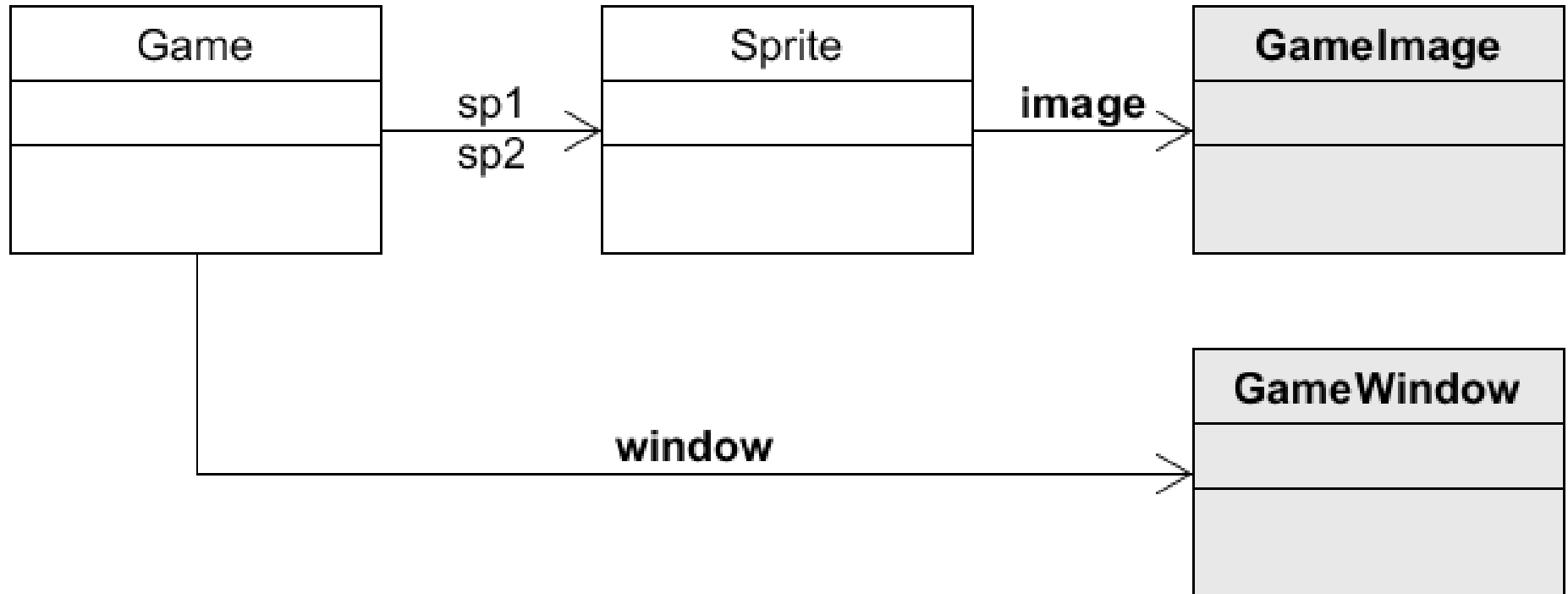


Bisher...

„Sprites“ bewegen
sich als Punkte
in einem
Koordinatensystem



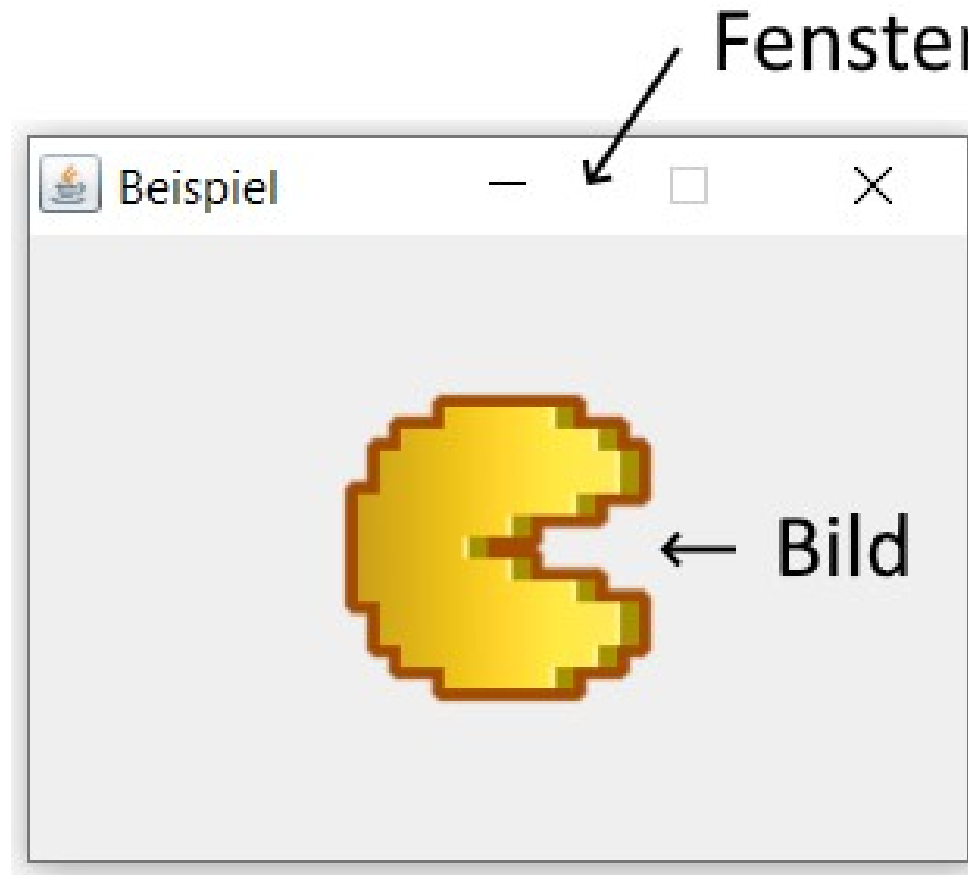
Neu: Sprites mit Grafik



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

Klasse GameWindow → **Fenster** zum Zeichnen

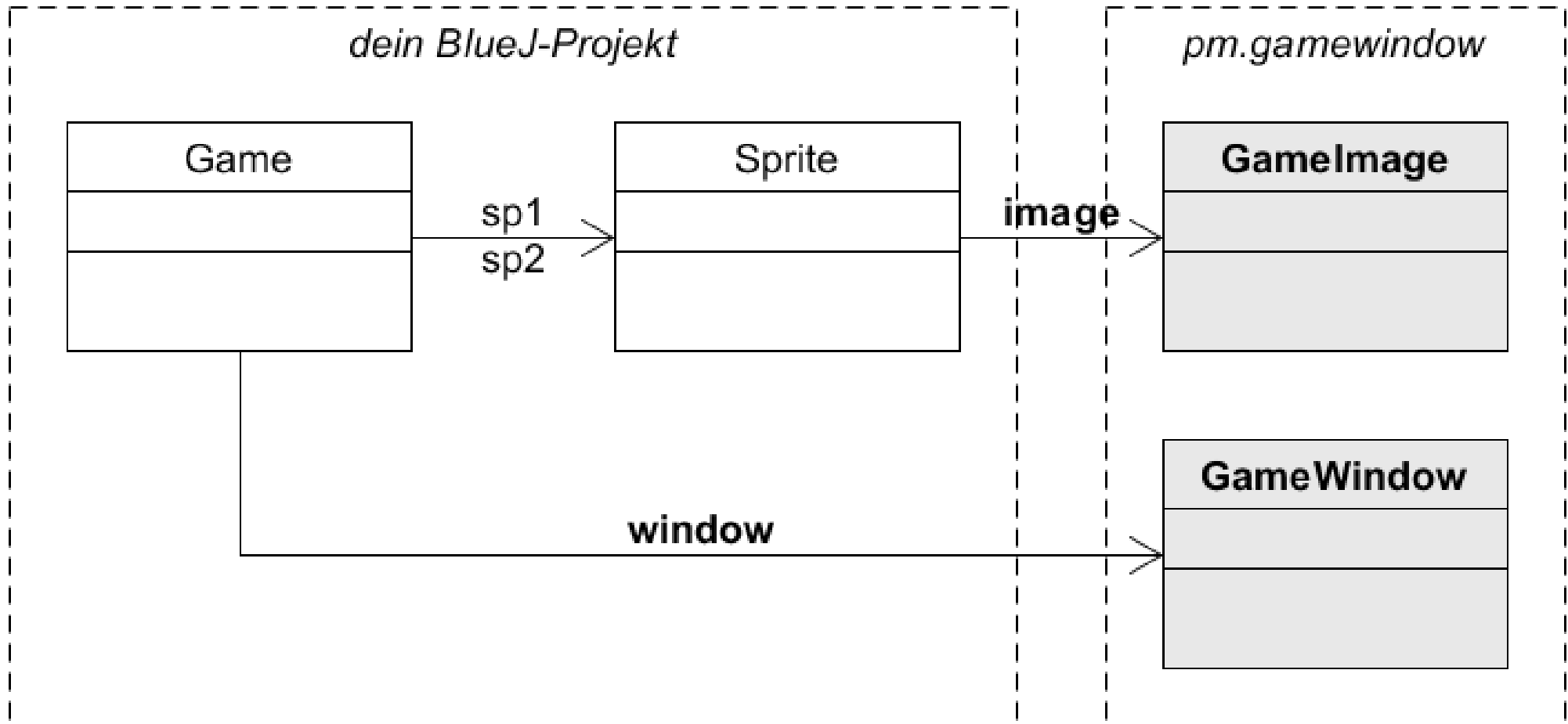
Neu: Sprites mit Grafik



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;
```

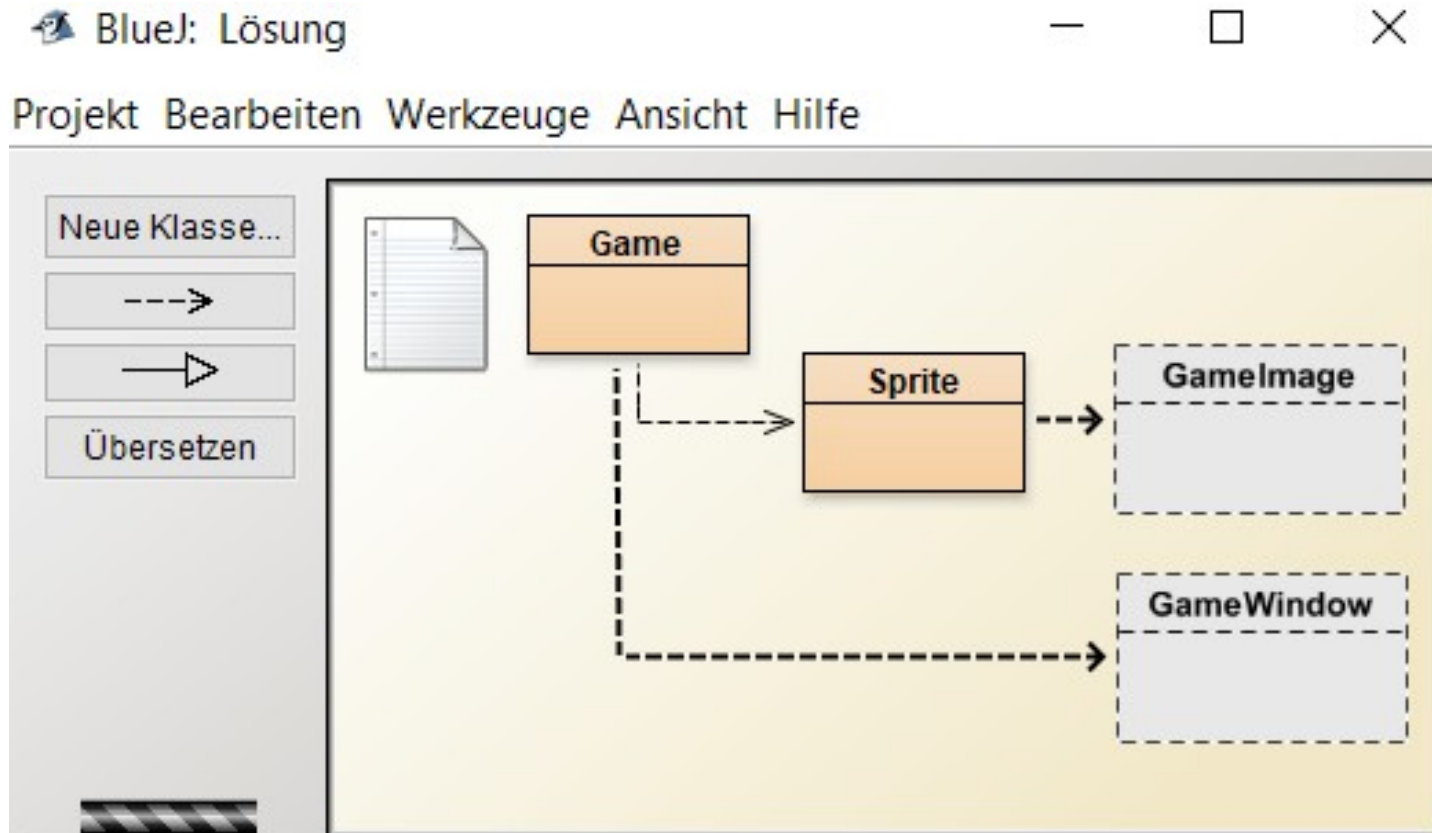
```
    double richtung;
```

```
    double speed;
```

```
    GameImage image;
```

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

Klassen importieren

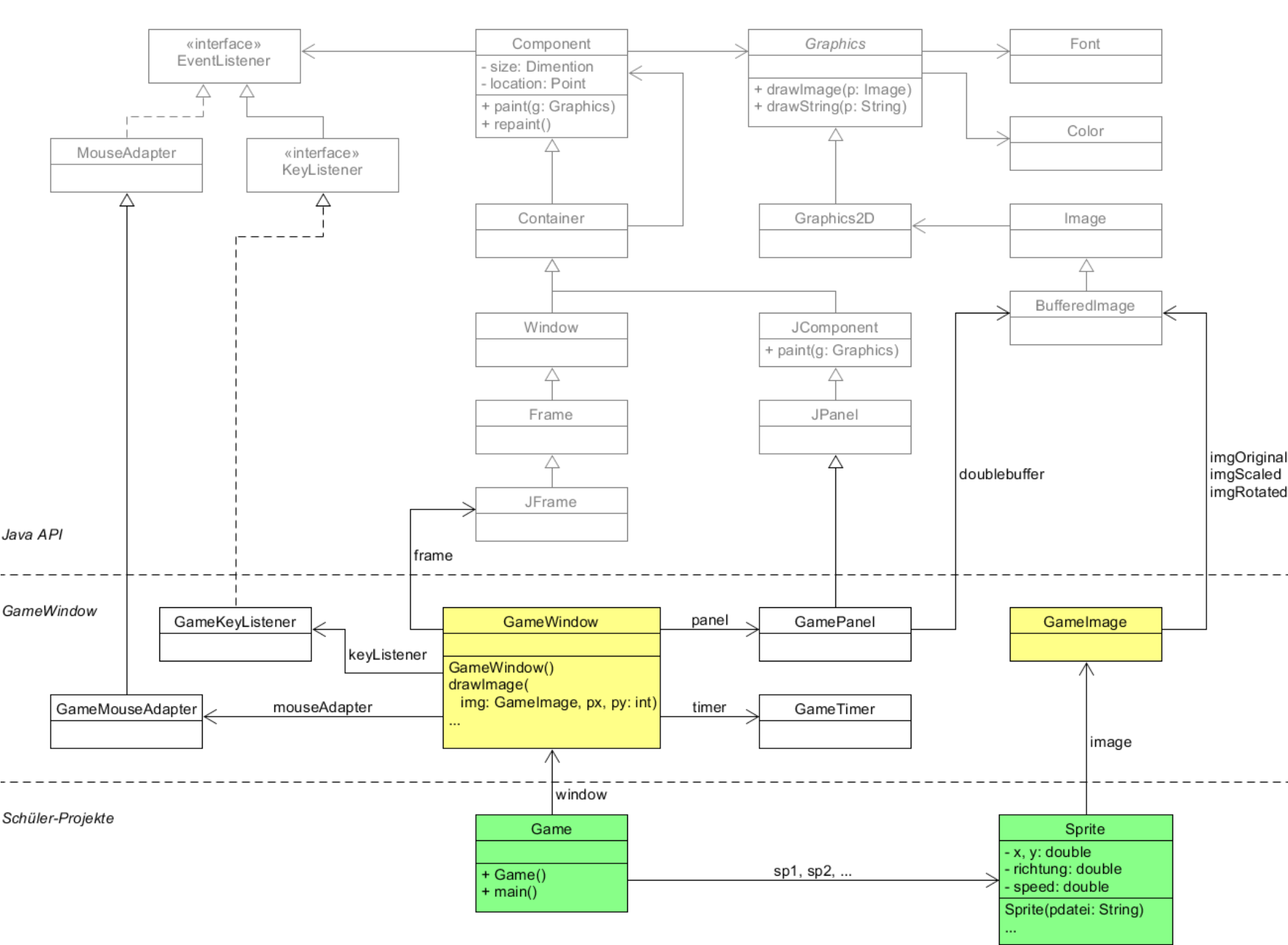


Die importierten Klassen musst du dir „dazudenken“ - sie werden nicht in BlueJ angezeigt!

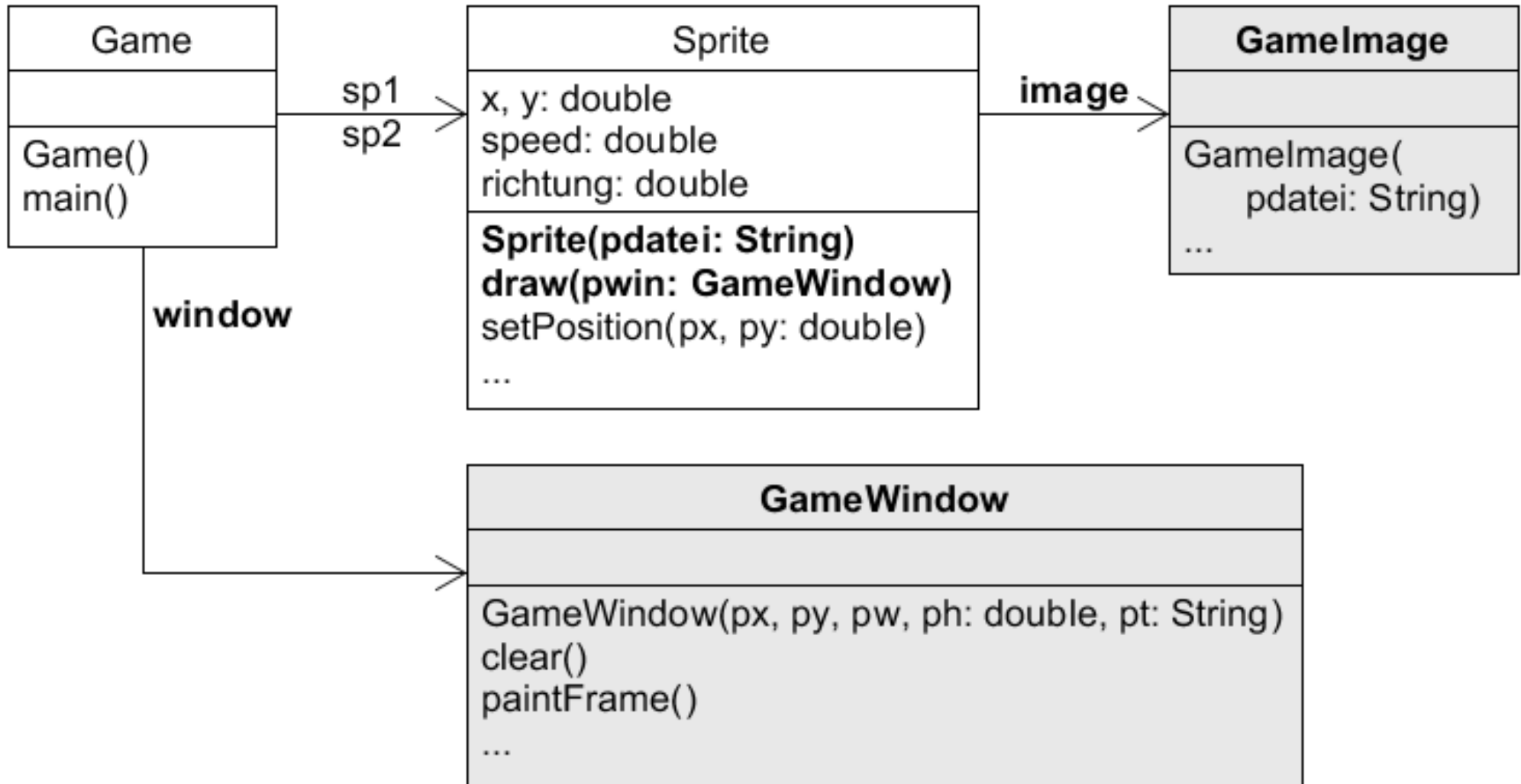
Bibliotheken ...

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

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



Projekt mit GameWindow im Detail



Klasse GameImage

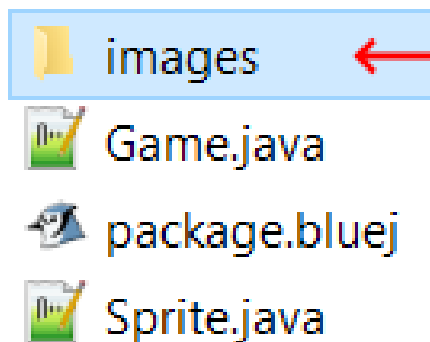
Konstruktor: GameImage(String pdatei)

→ lädt ein Bild aus einer Bilddatei

Der **Parameter** muss den vollständigen Pfad zur Bilddatei enthalten.

Beispiel:

```
image = new GameImage("./images/ghost.png");
```



← Unterordner des BlueJ-Projekts
enthält die Bilddatei

Klasse `GameWindow`

Konstruktor:

```
GameWindow(int px, int py, int pwidth, int pheight,  
           String ptitle)
```

`px / py` → Position des Fensters auf dem Bildschirm

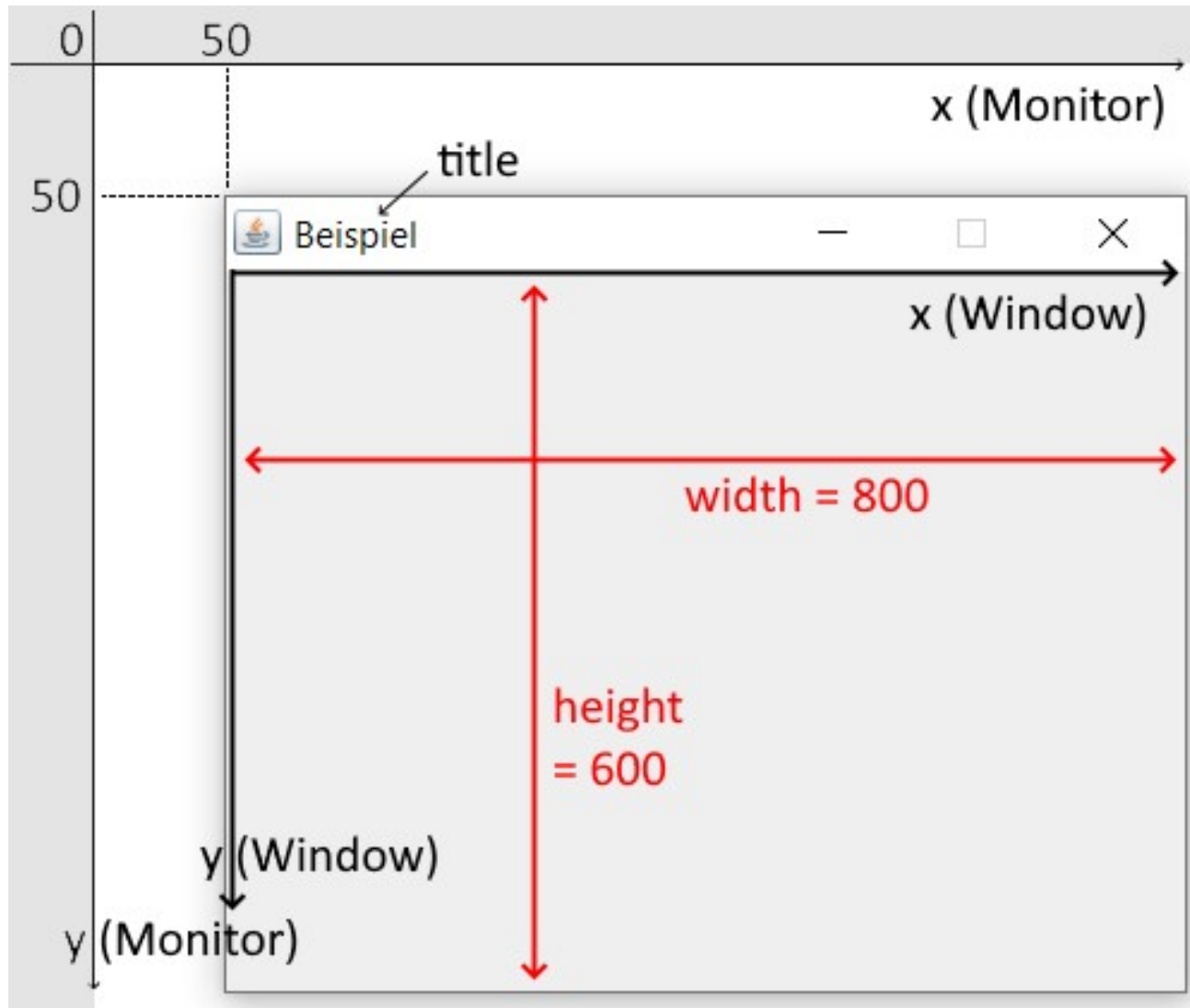
`pwidth / pheight` → Breite und Höhe in Pixeln

`ptitle` → Text für die Titelleiste

Beispiel:

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

Klasse GameWindow



Koordinaten-
Ursprung
des Monitors
und des
GameWindow
liegt jeweils
in der
linken, oberen
Ecke!

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

Klasse Sprite

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

    // Konstruktor: erzeugt GameImage-Objekt
    Sprite(String pdatei)
    {
        image = new GameImage(pdatei);
    }

    ...
}
```


Klasse Sprite

```
...  
/**  
 * Zeichnet das GameImage des Sprites  
 * an den Koordinaten x / y  
 * auf das GameWindow,  
 * das als Parameter „pwin“ übergeben wird.  
 */  
void draw(GameWindow pwin)  
{  
    pwin.drawImage(image, x, y);  
}  
...
```

Hauptklasse

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

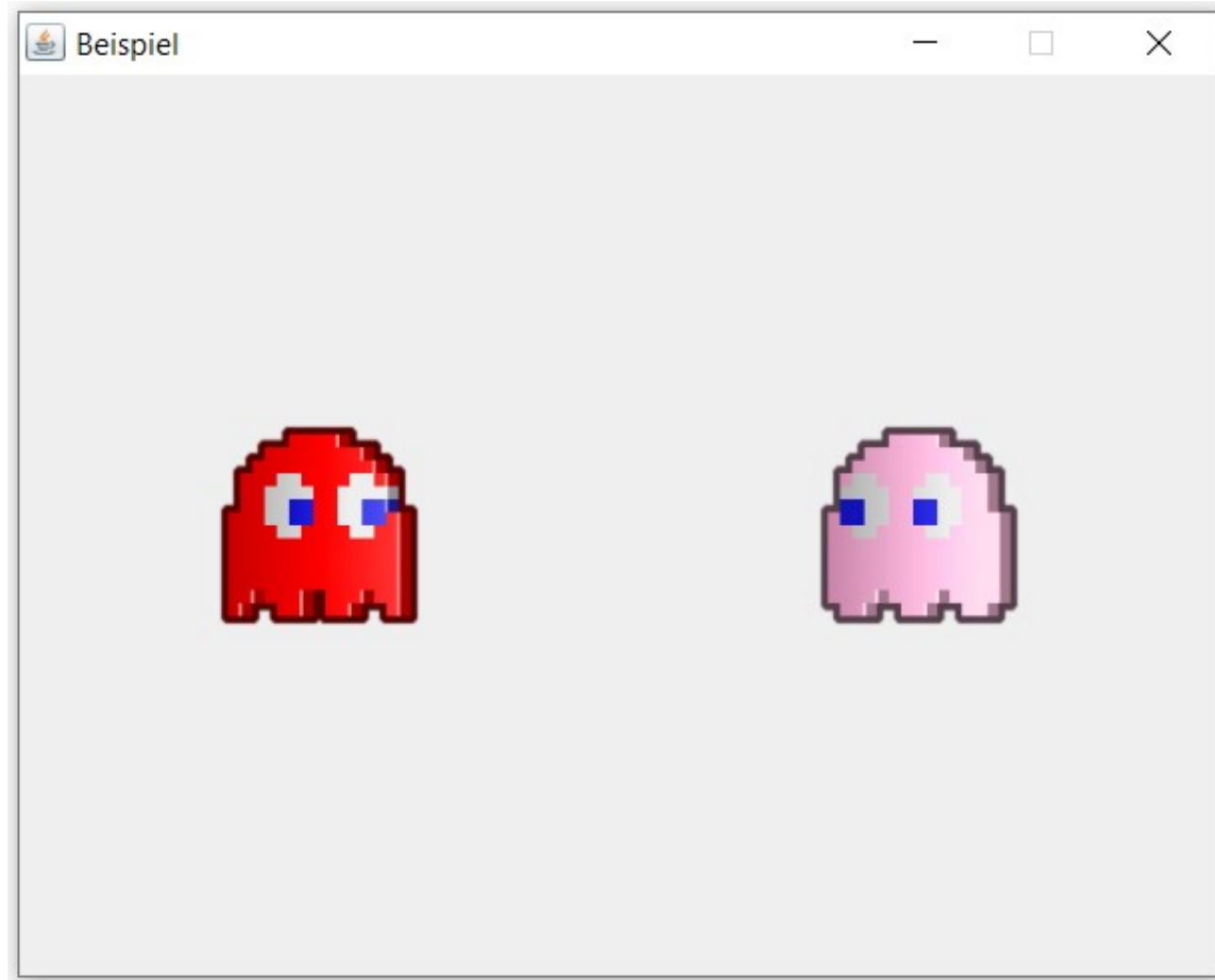
    Game()
    {
        sp1 = new Sprite("images/ghost1.png");
        sp2 = new Sprite("images/ghost2.png");
        window = new GameWindow
            (50, 50, 800, 600, "Beispiel");
    }
    ...
}
```

Hauptklasse

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

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