

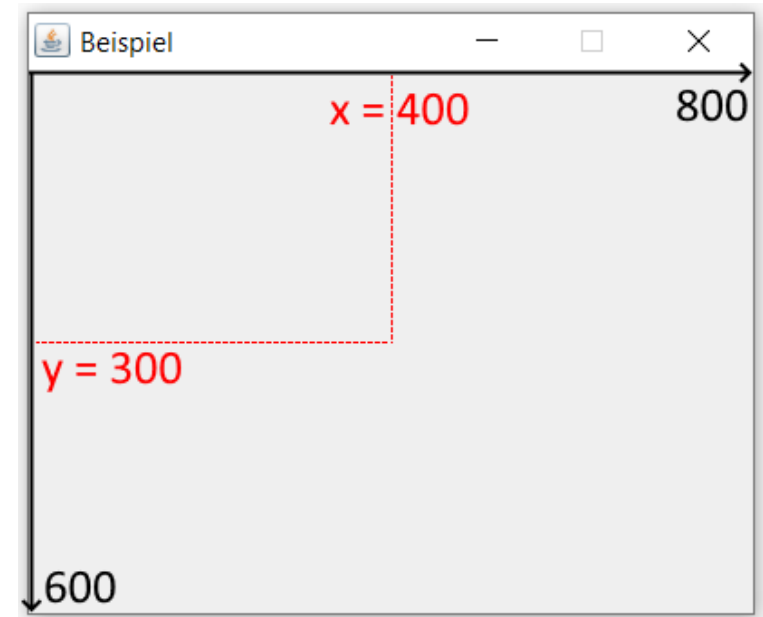
# **OOP mit Java**

## **Daten abfragen mit Get-Methoden**

# Beispiel: Sprite in die Mitte setzen

```
class Game
{
    GameWindow window;
    GameImage img;
    Sprite sp;

    Game ()
    {
        window = new GameWindow
            (0, 0, 800, 600);
        img = new GameImage("./ima...
        sp = new Sprite(img);
    }
}
```



# Beispiel: Sprite in die Mitte setzen

```
void aufgabe()  
{  
    sp.setPosition(400, 300);  
    window.clear();  
    sp.draw(window);  
    window.paintFrame();  
    ...  
}
```

# Beispiel: Sprite in die Mitte setzen

```
void aufgabe()  
{  
    sp.setPosition(400, 300);  
    ...  
}
```

Funktioniert nur, wenn das GameWindow eine Breite von 800 / Höhe von 600 hat.

Wenn man den Quellcode ändert ...

# Beispiel: Sprite in die Mitte setzen

```
class Game
{
    GameWindow window;
    GameImage img;
    Sprite sp;

    Game()
    {
        window = new GameWindow
            (0, 0, 1200, 900);
        img = new GameImage("./ima...
        sp = new Sprite(img);
    }
}
```

# Beispiel: Sprite in die Mitte setzen

```
void aufgabe()  
{  
    sp.setPosition(400, 300);  
    ...  
}
```

... wäre dieser Code fehlerhaft.

# Beispiel: Sprite in die Mitte setzen

```
void aufgabe()  
{  
    sp.setPosition(400, 300);  
    ...  
}
```

Auch möglich:

Der Benutzer kann das Fenster größer /  
kleiner ziehen

→ im Quellcode ist gar nicht bekannt,  
wie groß das Fenster ist.

# Beispiel: Sprite in die Mitte setzen

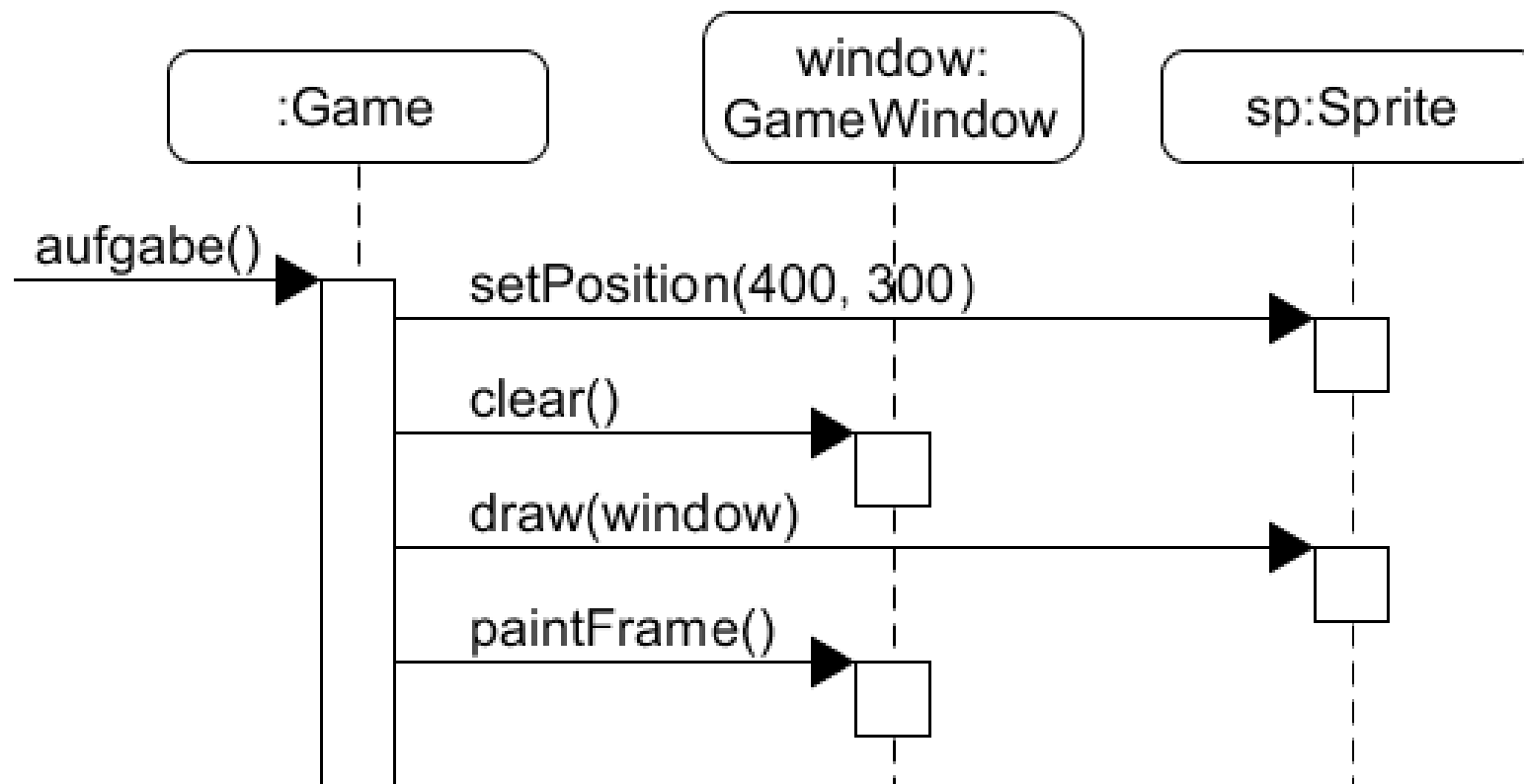
```
void aufgabe()  
{  
    sp.setPosition(400, 300);  
    ...  
}
```

Stattdessen wünschenswert:

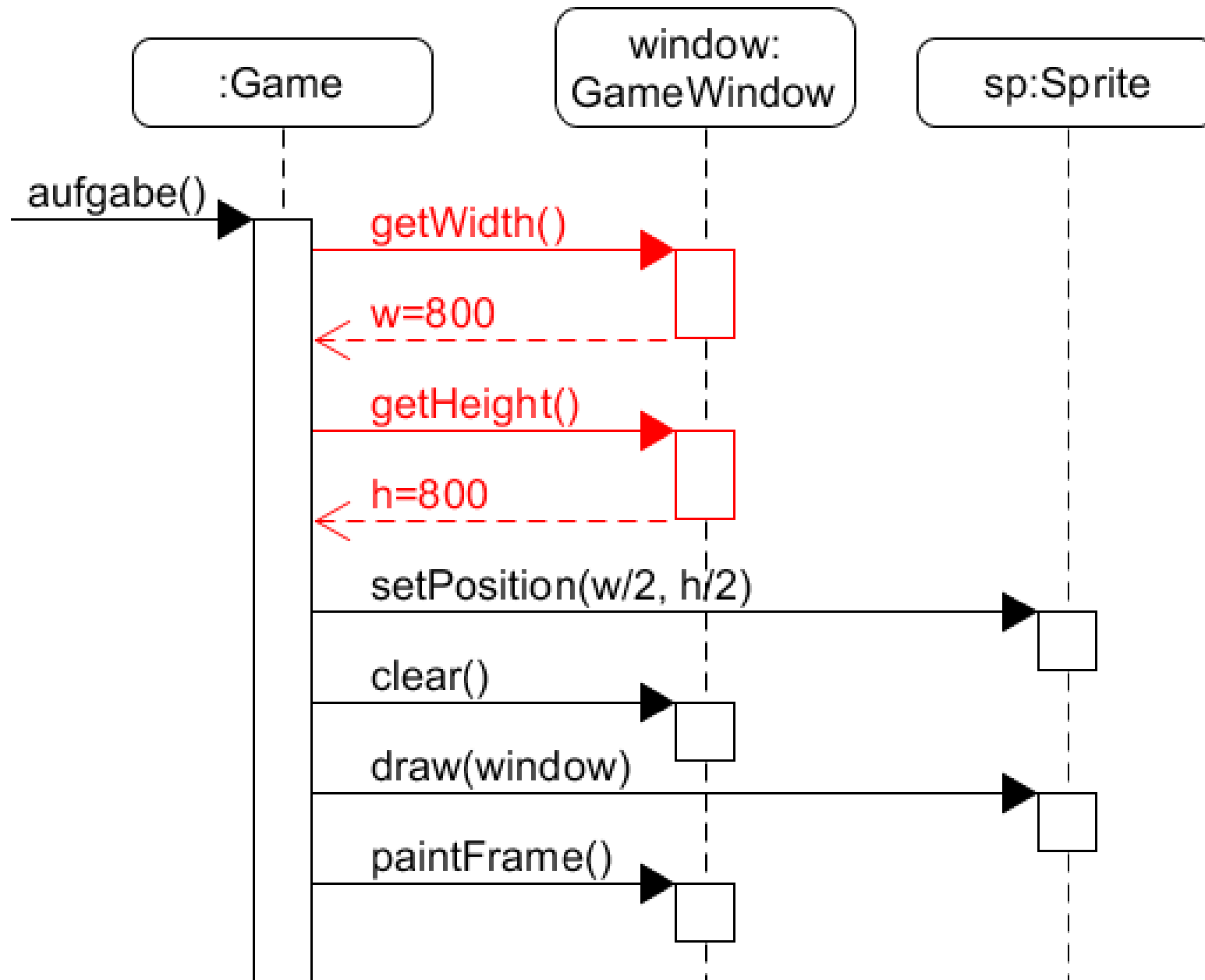
Die Breite / Höhe des Fensters vom  
window-Objekt **holen** / erfragen.



# Bisher: Befehle an Objekte senden



# Neu: Daten von Objekten holen / abfragen



# Daten mit get-Methoden „holen“

```
void aufgabe()  
{  
    int w, h;  
    w = window.getWidth();  
    h = window.getHeight();  
  
    sp.setPosition(w/2, h/2);  
    ...  
}
```

# Daten mit get-Methoden „holen“

```
void aufgabe()  
{  
    int w, h;  
    w = window.getWidth();  
    h = window.getHeight();  
  
    sp.setPosition(w/2, h/2);  
    ...  
}
```

getWidth() „holt“ die Breite  
vom Objekt window ...

# Daten mit get-Methoden „holen“

```
void aufgabe()  
{  
    int w, h;  
    w = window.getWidth();  
    h = window.getHeight();  
  
    sp.setPosition(w/2, h/2);  
    ...  
}
```

... und die Zahl, die zurückgeliefert wird, wird zunächst in Variable w gespeichert, um dann weiter verwendet zu werden.

# Get-Methoden direkt einsetzen

```
void aufgabe()  
{  
    sp.setPosition(  
        window.getWidth() /2,  
        window.getHeight() /2) ;  
    ...  
}
```

Alternativ kann man die get-Methoden auch direkt als Parameter einsetzen.

# Get-Methoden allgemein

```
void methode()  
{  
    Datentyp var;  
    var = objekt.getWert();  
    ...  
    (var irgendwo einsetzen)  
    ...  
}
```

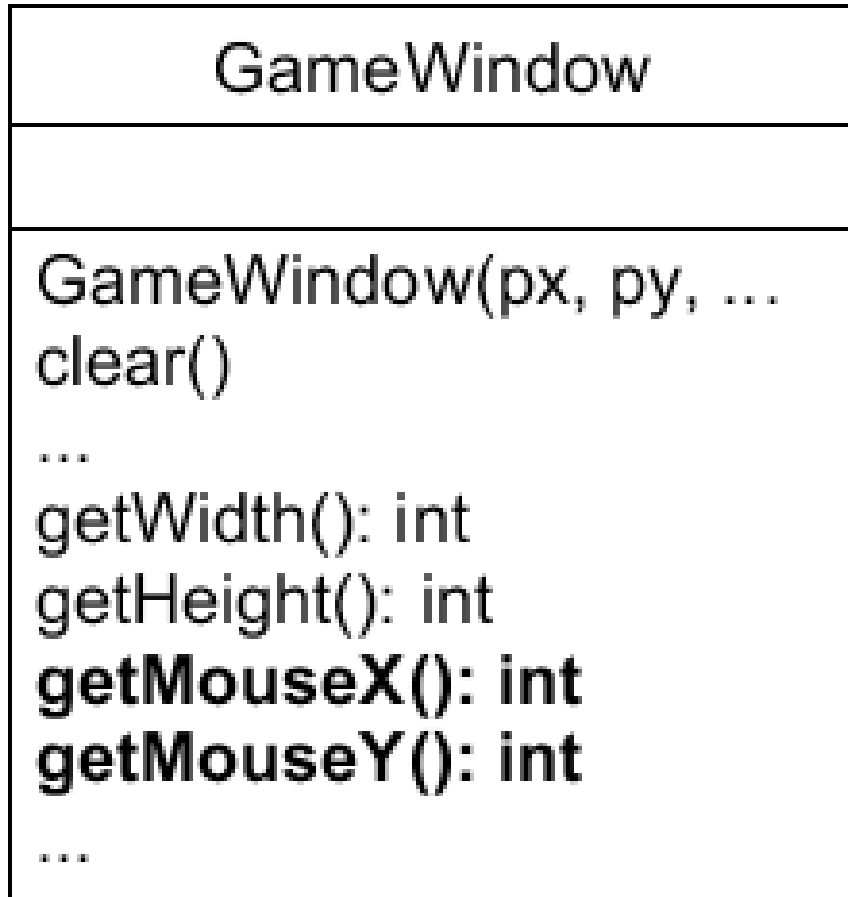
# GameWindow: Weitere Get-Methoden

GameWindow
GameWindow(px, py, ... clear() ... getWidth(): int getHeight(): int <b>getMouseX(): int</b> <b>getMouseY(): int</b> ...

getMouseX() und  
getMouseY() „holen“  
die aktuelle Position  
des Mauszeigers im  
GameWindow



# Get-Methdoden im Klassendiagramm



Allgemein:  
methode(): Datentyp

Der Datentyp gibt an,  
welche Art von Wert  
mit der Methode  
abgefragt wird.

# Autor / Quellen

Autor:

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

