

# **Array von Referenzen**

# Array „einfacher“ Werte

<i>Index</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Wert	7	12	25	28	33	41

Zugriff auf Elemente über „Index“

Anzahl Elemente wird bei Erzeugung festgelegt: „statische Länge“

# Array im Quellcode

```
public class Beispiel
{
    private int[] liste;

    public Beispiel()
    {
        liste = new int[10];
    }
}
```

Deklaration  
mit [ ]

Erzeugung im  
Konstruktor  
mit Anzahl der  
Elemente

# Array im Quellcode

```
public class Beispiel  
{
```

```
    ...
```

```
    public void main()  
    {
```

```
        liste[3] = 199;
```

```
        Console.println(liste[3]);
```

```
        ...
```

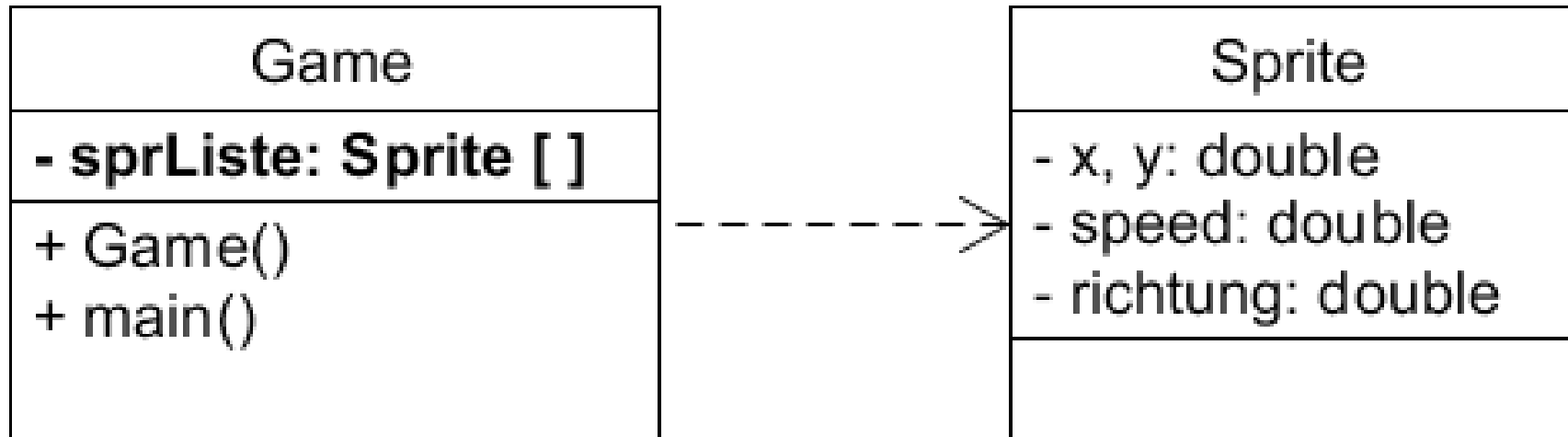
```
    }
```

Element ändern  
oder verwenden  
mit Index

# Array im Klassendiagramm



# Array von Referenzen



Ein Objekt der Klasse Game „hat“ Referenzen der Klasse Sprite.

Die Assoziation wird nicht (immer) dargestellt.

# Array von Referenzen

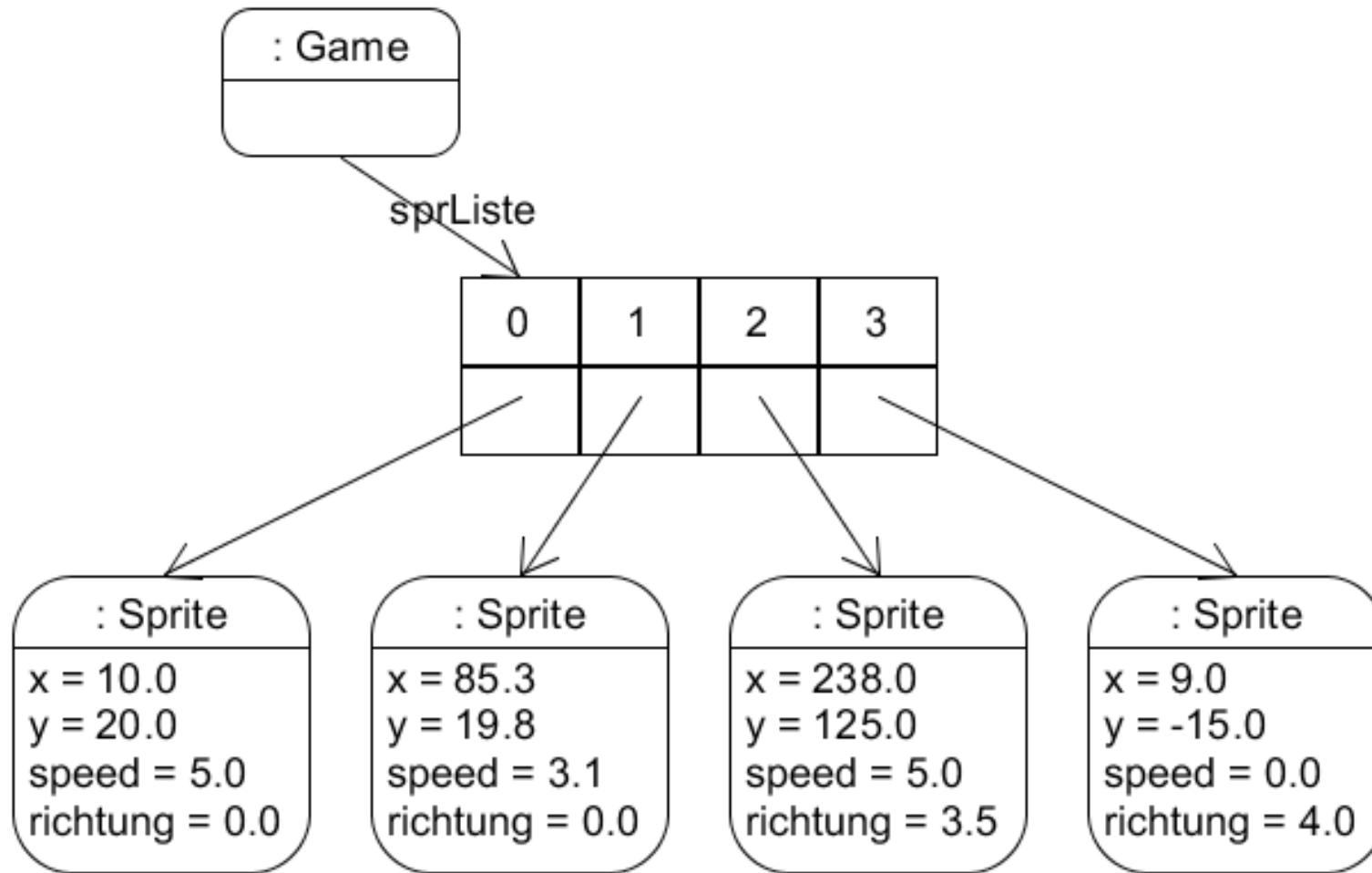
Game
- sprListe: <b>Sprite</b> [ ]
+ Game() + main()

Sprite
- x, y: double - speed: double - richtung: double

Ein Objekt der Klasse Game „hat“ Referenzen der Klasse Sprite.

Die Assoziation wird nicht (immer) dargestellt.

# Objektdiagramm



UML bietet eigentlich keine Darstellung für Arrays  
→ mit der hier gezeigten kann man aber arbeiten.



# Array von Objekten erzeugen

```
public class Game  
{
```

```
    private Sprite[] sprListe;
```

Deklaration

```
    public Game()  
    {
```

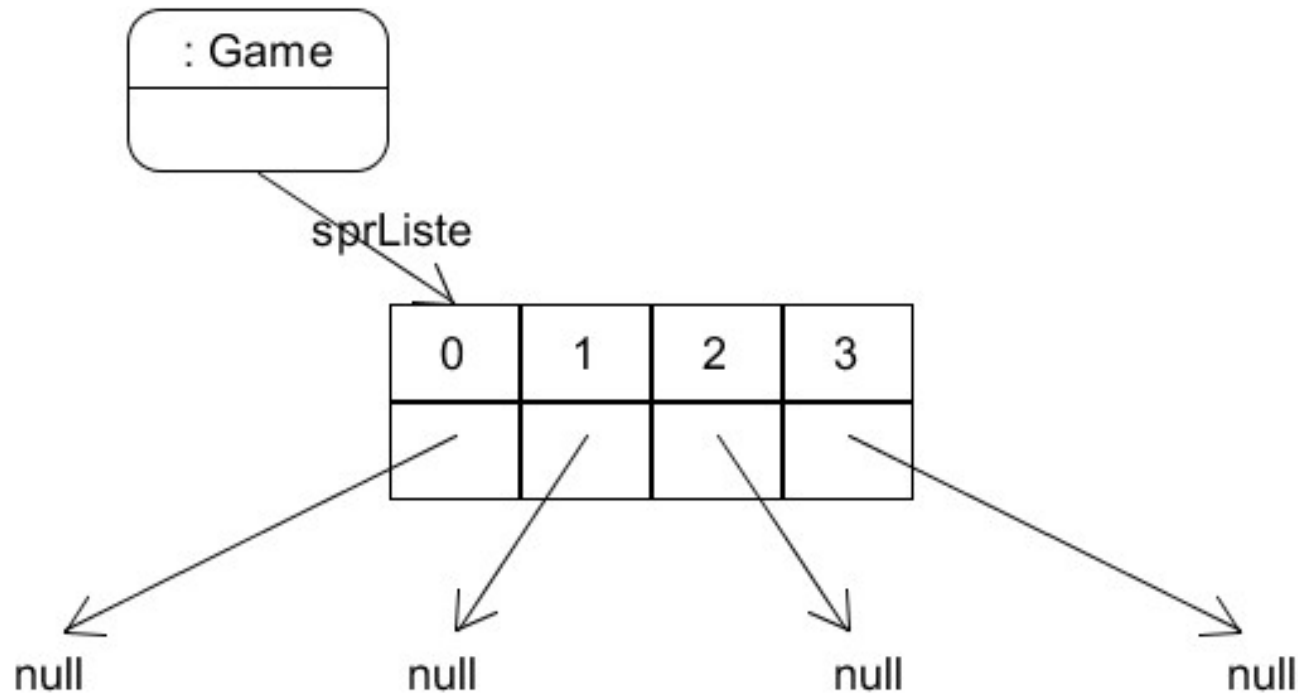
```
        sprListe = new Sprite[4];
```

```
        ...
```

```
    }
```

erzeugt **nur**  
das Array,  
nicht die  
Objekte!

# Array von Objekten erzeugen

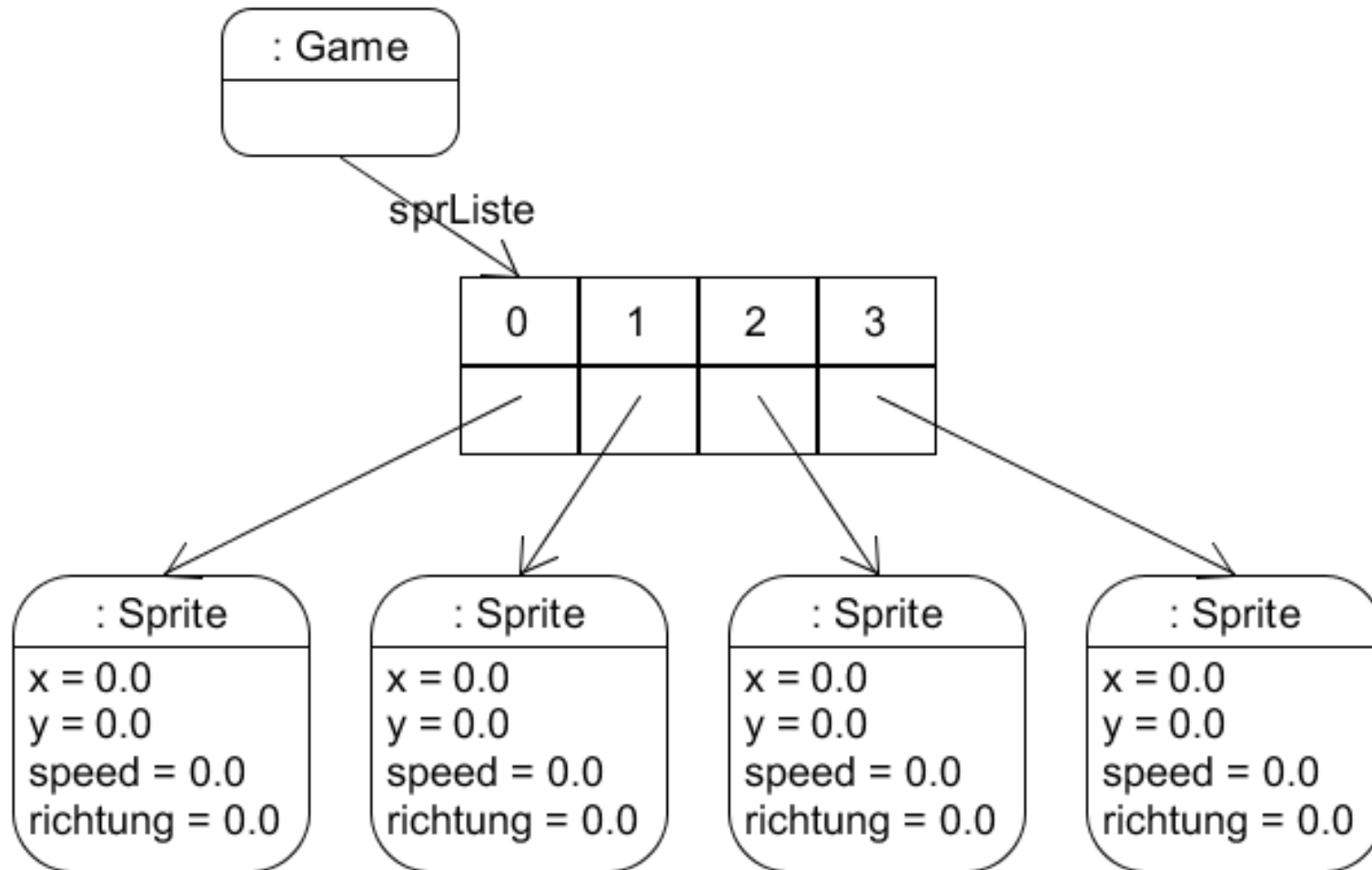


Referenzen im Array sind zunächst null.  
Die Objekte müssen noch erzeugt werden.

# Array von Objekten erzeugen

```
public class Game
{
    ...
    public Game()
    {
        sprListe = new Sprite[4];
        sprListe[0] = new Sprite(...);
        sprListe[1] = new Sprite(...);
        sprListe[2] = new Sprite(...);
        sprListe[3] = new Sprite(...);
    }
}
```

# Objektdiagramm



# Methoden ausführen

```
public class Game
{
    ...
    public void main()
    {
        sprListe[0].setPosition(100, 200);
        ...
    }
}
```

Für einzelne Referenzen des Arrays:  
Methoden mit **Punktoperator** ausführen

# Änderungen der Klasse Sprite

```
public class Game
{
    private GameImage img1;
    private Sprite sp1, sp2;

    public Game()
    {
        img1 = new GameImage("./img/bild.png");
        sp1 = new Sprite(img1);
        sp2 = new Sprite(img1);
    }
}
```

# Änderungen der Klasse Sprite

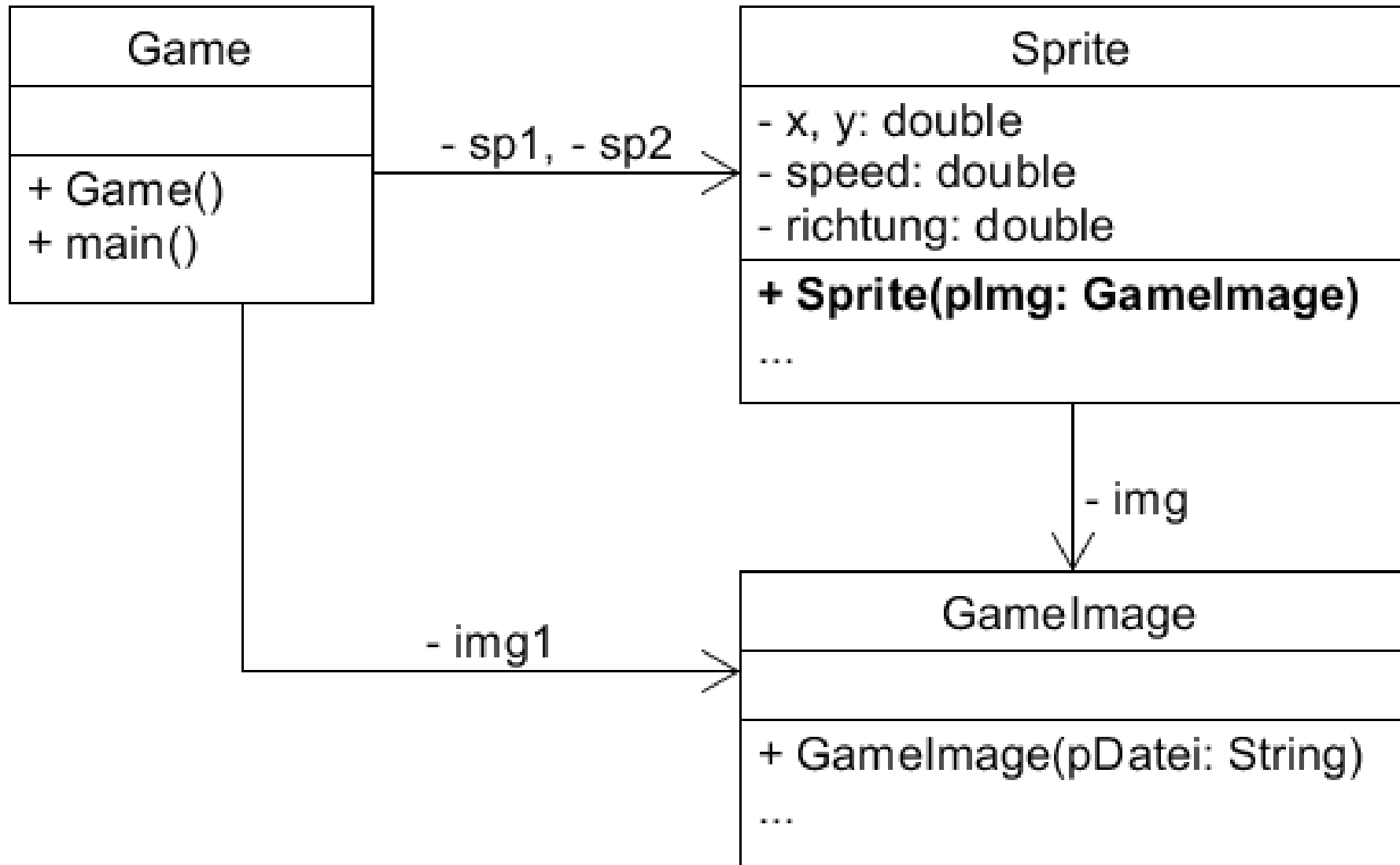
```
...  
public Game()  
{  
    img1 = new GameImage("./img/bild.png");  
    sp1 = new Sprite(img1);  
    sp2 = new Sprite(img1);  
}
```

Einem Sprite wird ein GameImage zugeordnet  
(vorher erzeugten Sprites selbst ihre Bilder).

Das GameImage wird vom Hauptobjekt erzeugt.

→ ein Bild für mehrere Sprites verwendbar

# Änderungen der Klasse Sprite





# Autor / Quellen

Autor:

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

