

Hochsprachenprogrammierung

Labor

Objektorientierung

Prof. Dr.-Ing. Kai Mecke

- Aufgabenstellung Sortierer
 - Einfache Arrays
 - Algorithmen
 - do-while
 - Objektorientierung
 - Objekte erzeugen
 - im Speziellen: Sortierer

AUFGABE:

ERZEUGEN SIE EINE KLASSE FÜR EINEN SORTIERER

- Erstellen Sie eine Klasse „Sortierer“
 - In der Main Methode erzeugen Sie ein Objekt von der Klasse selbst
 - Fügen Sie den Konstruktor hinzu, der
 - unsortierte Zahlen bereithält (int[] Array)
 - die Methode BubbleSort aufruft
 - den Sortierten Array ausgibt
 - BubbleSort (weitere Methode) den Array sortiert und zurückliefert

Sortierer
<pre>+ Sortierer() + BubbleSort(int[] intArray):int[] + Main(String[] args): static void</pre>

- Wir brauchen etwas zum Sortieren
- Arrays sind mehrere Elemente eines Datentyps

```
int[] intArray;  
intArray = new int[5];
```

```
intArray[0] = 7;  
intArray[1] = 3;  
intArray[2] = 1;  
intArray[3] = 2;  
intArray[4] = 6;
```

```
// Alles in einer Zeile  
int[] intArray = new int[] { 7, 3, 1, 2, 6 };
```

- Wir brauchen etwas, das sortieren kann

Algorithmus:

1. Vergleiche nacheinander die Nachbarn in einem Array von Zahlen
2. Vertausche diese, rechts kleiner ist als links
3. Wenn ein Tausch stattgefunden hatte, dann ist der Array noch nicht sortiert, also mache das Ganze noch einmal.

Übersicht: <http://www.sorting-algorithms.com/>

Detail: <http://math.hws.edu/eck/jsdemo/sortlab.html>

➔ Bubble Sort auswählen

ZWEITER BAUSTEIN

BUBBLESORT

7	3	1	2	6
3	1	2	6	7
1	2	3	6	7
1	2	3	6	7

[.....]

```
int[] intArray = new int[] { 7, 3, 1, 2, 6 };  
outArray(intArray);  
outArray(bubbleSort(intArray));
```

[.....]

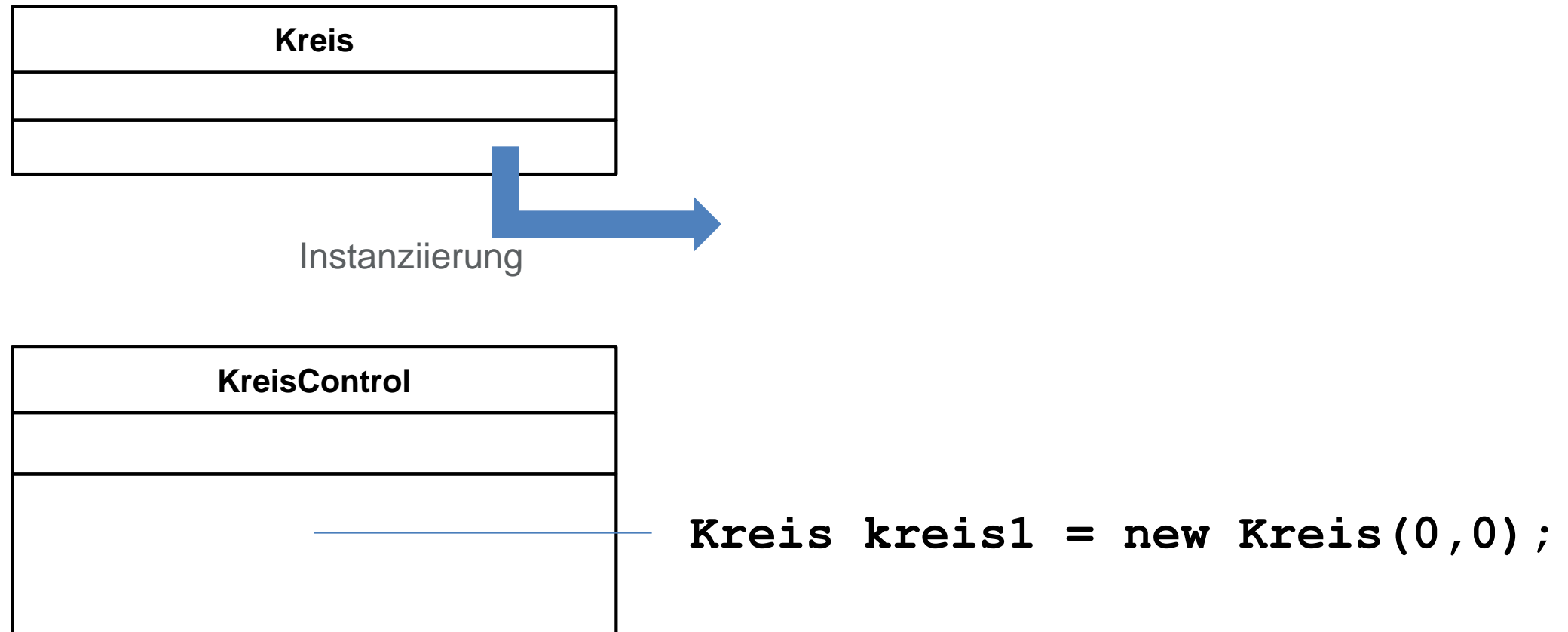
```
public void outArray(int[] intArray)  
{  
    [.....]  
}
```

```
public int[] bubbleSort(int[] intArr)  
{  
    Boolean sortiert;  
    do  
    {
```

[.....]

```
    } while (!sortiert);  
    return intArr;  
}
```

- Objekte werden aus einem anderen Objekt heraus erzeugt! → Instanziierung




```
class KreisControl
{
    KreisControl()
    {
        Kreis meinKreis1 = new Kreis(0, 0);
        meinKreis1.setMittelpunkt(10f, 30f);
    }

    static void Main(string[] args)
    {
        new KreisControl();
    }
}
```

Code aus der Vorlesung

```
public class Kreis
{
    private float x, y;
    private float radius;

    public Kreis(float l_x, float l_y)
    {
        x = l_x;
        y = l_y;
    }

    public void setMittelpunkt(
        float l_x, float l_y)
    {
        x = l_x;
        y = l_y;
    }
}
```

- Erstellen Sie eine Klasse „Sortierer“
 - In der Main Methode erzeugen Sie ein Objekt von der Klasse selbst
 - Fügen Sie den Konstruktor hinzu, der
 - unsortierte Zahlen bereithält (int[] Array)
 - die Methode BubbleSort aufruft
 - den Sortierten Array ausgibt
 - BubbleSort (weitere Methode) den Array sortiert und zurückliefert

Sortierer
<pre>+ Sortierer() + BubbleSort(int[] intArray):int[] + Main(String[] args): static void</pre>

- Erstellen Sie eine Klasse „Sortierer“
 - In der Main Methode erzeugen Sie ein Objekt von der Klasse selbst
 - Fügen Sie den Konstruktor hinzu, der
 - unsortierte Zahlen bereithält (int[] Array)
 - die Methode BubbleSort aufruft
 - den Sortierten Array ausgibt
 - BubbleSort (weitere Methode) den Array sortiert und zurückliefert

Program
+ Main(String[] args): static void

Sortierer
+ Sortierer() + BubbleSort(int[] intArray):int[]