Einstieg in die Informatik mit Java

Variablenarten

Gerd Bohlender

Institut für Angewandte und Numerische Mathematik

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Lokale Variablen in Methoden (z.B. main)

```
public class Demo {
  static int methode () {
    int lokal;
    ...
  }
}
```

Eigenschaften

- Variable darf nur in *dieser* Methode verwendet werden – ab der Definitionsstelle bis zum Ende der Methode
- Sie ist in anderen Methoden unbekannt, darf dort unabhängig in anderer Bedeutung definiert werden
- Der Variablenname muss in der Methode eindeutig sein
- Der Wert wird *nicht* automatisch initialisiert

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Lokale Variablen in Blocks

```
public class Demo {
  static int methode () {
    ...
    { // Blockanfang
      int lokal;
      ...
    } // Blockende
    ...
  }
}
```

Eigenschaften

- Wie lokale Variable
- Gilt aber nur bis Blockende

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Lokale Variablen in for-Schleife

```
public class Demo {
  static int methode () {
    ...
    for (int i=0; ...
    ...
  }
}
```

Eigenschaften

- Wie lokale Variable
- Gilt aber nur bis Schleifenende

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Formale Argumente von Methoden

```
public class Demo {
   static int methode (double d) {
      ...
   }
}
```

Eigenschaften

- Wie lokale Variable
- Der Wert wird mit dem Wert des entsprechenden aktuellen Arguments initialisiert

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Klassenvariablen

```
public class Demo {
   static int klasse;
   ...
   static int methode () {
      ...
   }
}
```

Eigenschaften

- Variable darf in *allen* Methoden dieser Klasse verwendet werden
- Der Variablenname muss in der Klasse eindeutig sein, darf aber in Methoden für lokale Variablen wiederverwendet werden
- Dann gilt in der Methode der lokale Variablenname, die Klassenvariable ist zugänglich über Klassenname.Variablenname
- Der Wert wird automatisch initialisiert (je nach Typ mit 0, 0.0, false, null, ...)

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Klassenvariablen

```
public class Demo {
  static int methodeDavor () {
    return x;          // okay
  }
  static int x=333;
  static int y=x+1;
  //Initialisierung nur in dieser Reihenfolge
  ...
  static int methodeDanach () {
    return x;          // okay
  }
}
```

Eigenschaften

- Variable darf in *allen* Methoden verwendet werden, vor und nach der Definition – die Reihenfolge spielt keine Rolle
- Bei der Initialisierung muss die Reihenfolge beachtet werden

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Instanzvariablen

```
public class Demo {
  int x;
  ...
  int methode () {
    return x;          // okay
  }
}
```

Eigenschaften

- Instanzvariable darf in allen *Instanz*methoden verwendet werden, vor und nach der Definition – die Reihenfolge spielt keine Rolle
- Bei der Initialisierung muss die Reihenfolge beachtet werden
- In Klassenmethoden und bei der Initialisierung von Klassenvariablen sind Instanzvariablen nur erlaubt, wenn eine konkrete Instanz genannt wird:
  `Instanzname.Instanzvariable`

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Vergleich Klassen- und Instanzvariablen

Unterschiede

- Klassenvariablen existieren einmal für die gesamte Klasse.
- Instanzvariable existieren individuell für jede Instanz der Klasse.
- Gibt es *n* Instanzen einer Klasse, dann gibt es also
  - jede Klassenvariable einmal
  - jede Instanzvariable *n*-mal

```
public class Demo {
  static int x=1;
  int y=2;
  public static void main (String[] args) {
    // es gibt 1 x, 0 y
    Demo d = new Demo();
    // es gibt 1 x, 1 y
    Demo e = new Demo();
    // es gibt 1 x, 2 y
  }
}
```

Gliederung

1. Lokale Variablen
2. Lokale Variablen in Blocks
3. Lokale Variablen in for-Schleife
4. Formale Argumente
5. Klassenvariablen
6. Klassenvariablen – Reihenfolge
7. Instanzvariablen
8. Vergleich Klassen- und Instanzvariablen
9. Beispiele

Beispiele – erlaubte Wiederverwendung von Namen

```
public class Demo {
  int methode1 () {
    int x=1;    // okay
    return x;   // liefert 1
  }
  int methode2 () {
    int x=2;    // okay
    return x;   // liefert 2
  }
}
```

Beispiel

```
public class Demo {
  static int x=1;
  static int methode () {
    int x=2;              // okay
    return x+Demo.x;   // liefert 2+1=3
  }
}
```

Beispiele – verbotene Wiederverwendung von Namen

```
public class Demo {
  int methode (int x) {
    int x=2;    // Fehler
    return x;
  }
}
```

Achtung

Formales Argument und lokale Variable dürfen nicht den gleichen Namen haben!

Beispiele – verbotene Wiederverwendung von Namen

```
public class Demo {
  void methode () {
    int i;
    ...
    for (int i=0; i<10; i++) { // Fehler
    }
  }
}
```

Achtung

Zwei lokale Variablen dürfen nicht den gleichen Namen haben. Dies gilt auch für Variablen, die im Init-Teil einer for-Schleife definiert werden

Beispiele – verbotene Wiederverwendung von Namen

```
public class Demo {
  void methode () {
    int x;
    do {
      int x;
      ...
    } while (...);
  }
}
```

Achtung

Zwei lokale Variablen dürfen nicht den gleichen Namen haben. Dies gilt auch für Variablen, die in einem untergeordneten Block definiert werden

Beispiele – Initialisierung

```
public class Demo {
  int a;
  int methode (int b) {
    int c;
    int[] d = new int[5];
    return a+b+c+d[2];
  }
}
```

- a ist mit 0 initialisiert
- b ist mit dem Wert des aktuellen Arguments initialisiert
- c ist nicht initialisiert
- d[2] ist mit 0 initialisiert (Felder sind Objekte, Komponenten werden also bei new initialisiert)
- also tritt bei der return-Anweisung 1 Fehler auf