

# Variablen in Java

Martin Wirsing

in Zusammenarbeit mit  
Michael Barth, Philipp Meier und Gefei Zhang

11/04

## Ziele

- Unterschiedliche Arten von Variablen in Java verstehen
- Bestimmen von Gültigkeitsbereich und Lebensdauer von Variablen

M. Wirsing: Variablen in Java

## Variablen in Java

Beispiel:

```
public class Point
{
    private int x;
    private int y;
    public void move(int x, int y)
    {
        this.x = this.x + x;
        this.y = this.y + y;
    }
    public static void main (String[] args)
    {
        Point p = new Point(10,20);
        Point p1 = new Point();
        p1.move(10,10); . . .
    }
}
```

M. Wirsing: Variablen in Java

## Lebensdauer, Gültigkeit, Initialisierung von Variablen

- In Java gibt es 4 Arten von Variablen:
  - Instanzvariablen
  - formale Parameter
  - lokale Variablen
  - (Klassenvariablen – siehe Literatur, in der Vorlesung nicht behandelt)
- Die **Gültigkeit** ist der Teil des Programms, in dem auf die Variable zugegriffen werden kann.
- Die **Lebensdauer** einer Variablen ist die Zeit von der Erzeugung der Variablen bis zu dem Zeitpunkt, in dem sie im Speicher gelöscht wird.

M. Wirsing: Variablen in Java

## Gültigkeit von Variablen

### Gültigkeit

- Attribute werden üblicherweise als „private“ spezifiziert; Instanzvariablen sind deshalb nur in den Methodenrumpfen der eigenen Klasse gültig.
- Der Gültigkeitsbereich einer lokalen Variable erstreckt sich von der Deklaration bis zum Ende des einschließenden Blocks.
- Der Gültigkeitsbereich eines formalen Parameters ist der Rumpf des zugehörigen Blocks.
- Attribute können durch lokale Variablen und formale Parameter (eines Methodenrumpfs der gleichen Klasse) mit gleichem Namen verschattet werden. Dann ist die Kurzschreibweise für Instanzvariablen (ohne `this`) nicht möglich.

M. Wirsing: Variablen in Java

## Gültigkeit von Variablen

Beispiel:

```
public class Point
{
    private int x;
    private int y;

    public void move(int x, int y)
    {
        this.x = this.x + x;
        this.y = this.y + y;
    }

    public int getX()
    {
        return this.x;
    }

    public static void main (String[] args)
    {
        Point p = new Point(10,20);
        Point p1 = new Point();
        p1.move(10,10); . . .
    }
}
```

M. Wirsing: Variablen in Java

## Gültigkeit von Variablen

### Beispiel:

```
public class Point
{
    private int x;
    private int y;

    public void move(int x, int y)
    {
        this.x = this.x + x;
        this.y = this.y + y;
    }

    public int getX()
    {
        return this.x;
    }

    public static void main (String[] args)
    {
        Point p = new Point(10,20);
        Point pl = new Point();
        pl.move(10,10); . . .
    }
}
```

Gültigkeitsbereich der formalen Parameter x, y

M. Wirsing: Variablen in Java

## Gültigkeit von Variablen

### Beispiel:

```
public class Point
{
    private int x;
    private int y;

    public void move(int x, int y)
    {
        this.x = this.x + x;
        this.y = this.y + y;
    }

    public int getX()
    {
        return this.x;
    }

    public static void main (String[] args)
    {
        Point p = new Point(10,20);
        Point pl = new Point();
        pl.move(10,10); . . .
    }
}
```

Gültigkeitsbereich der lokalen Variablen p

M. Wirsing: Variablen in Java

## Gültigkeit von Variablen

### Beispiel:

```
public class Point
{
    private int x;
    private int y;

    public void move(int x, int y)
    {
        this.x = this.x + x;
        this.y = this.y + y;
    }

    public int getX()
    {
        return this.x;
    }

    public static void main (String[] args)
    {
        Point p = new Point(10,20);
        Point pl = new Point();
        pl.move(10,10); . . .
    }
}
```

Gültigkeitsbereich der lokalen Variablen p1

M. Wirsing: Variablen in Java

## Lebensdauer von Variablen

### Lebensdauer

- Die **Instanzvariablen** eines Objekts werden erzeugt, wenn das Objekt konstruiert wird. Die Variable lebt, bis der Speicherbereinigungsalgorithmus das Objekt beseitigt.
- Eine **lokale Variable** wird bei ihrer Deklaration erzeugt und lebt, bis der umfassende Block verlassen wird.
- Ein **formaler Parameter** ist ein lokale Variable des Methodenrumpfs. Er wird beim Methodenaufruf erzeugt und nach Ausführung des Methodenaufrufs wieder gelöscht.

M. Wirsing: Variablen in Java

## Initialisierung von Variablen

### Initialisierung

- Instanzvariable werden automatisch bei der Erzeugung mit einem Standardwert initialisiert (0 für ganze Zahlen, 0.0 für Gleitzahlen, **false** für **boolean**, **null** für Objekte), falls der Initialwert nicht explizit spezifiziert wurde (im Konstruktor oder bei der Deklaration).
- Parametervariable werden mit lokalen Kopien der Werte der aktuellen Parameter initialisiert.
- Lokale Variablen müssen durch das Programm explizit initialisiert werden und dürfen vor der Initialisierung nicht benutzt werden

M. Wirsing: Variablen in Java

## Zusammenfassung

- In Java gibt es vier Arten von Variablen: Instanzvariablen, lokale Variablen, formale Parameter und Klassenvariablen.
- Die **Lebensdauer** einer Variablen ist der Zeitraum von der Erzeugung der Variablen im Speicher bis zum Löschen der Variablen. Der **Gültigkeitsbereich** ist derjenige Teil des Programmtextes, in dem auf die Variable zugegriffen werden kann.

M. Wirsing: Variablen in Java