Informatikvorkurs Fachschaft Hochschule Karlsruhe - Technik und Wirtschaft

Samuel Zeitvogel

SS 2012

Informatikvorkurs Java

- Statische Methoden
  - Warum Methoden?
  - Was ist eine Methode?
- Klassenvariablen
  - Warum Klassenvariablen?
  - Was ist eine Klassenvariable?

Beispiel

```
public class Main {
    public static void main() {
        for(int i = 1; i <= 10; i++) {
            System.out.println(i);
        }

        //weiterer Code

        for(int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

Probleme

- Zeitaufwändig
- Viel Code entsteht
- Unübersichtlich
- Änderungen kosten noch mehr Zeit
- Code oft nicht wiederverwendbar
- Arbeitsteilung kaum Möglich

Lösung

- Ähnlicher Code Auslagern! => Methode

Beispiel

```
public class Main {
    public static void main() {
        zaehlBisZehn();
        //weiterer Code
        zaehlBisZehn();
    }
    public static void zaehlBisZehn() {
        for(int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

Kopf der Methode

- **public static** void zaehlBisZehn() {
  .....
  }

- **public static** immer am Anfang (wird im Vorkurs nicht behandelt)
- Vor den runden Klammern befindet sich der Methodenname

Aufruf einer Methode

- public static void main() {
  zaehlBisZehn();
  }

- Aufruf einer Methode:
  - 1) **Methodenname**
  - 2) **()**
  - 3) **;**

Methoden können noch mehr!

- Beim Methodenaufruf werden zusätzliche Informationen an die Methode übergeben (Parameter)
- Methoden geben Information an den Aufrufer zurück

Beispiel

```
public class Main {
    public static void main() {
        for(int i = 1; i <= 9; i++) {
            System.out.println(i);
        }

        //weiterer Code

        for(int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

Beispiel

```
public class Main {
    public static void main() {
        zaehlBis(9);
        //weiterer Code
        zaehlBis(10);
    }
    public static void zaehlBis(int z) {
        for(int i = 1; i <= z; i++) {
            System.out.println(i);
        }
    }
}
```

Kopf der Methode

- **public static** void zaehlBisZehn(int z) {
  .....
  }
- In die Runden Klammern kommen die Parameter und sie werden mit Kommas getrennt
- Beispiel:
  (int a, boolean b, double c)
- Ein Parameter besteht aus Datentyp + Bezeichner

Aufruf einer Methode mit Parameter

- public static void main() {
  zaehlBisZehn(9);
  zaehlBisZehn(10);
  }

- Aufruf einer Methode mit Paramter:
  - Paramter die man übergeben möchte mit Komma getrennt in die Runden Klammern

Was passiert?

- zaehlBis(9);

```
public static void zaehlBis(int z) {
    // z wird der Wert 9 zugewiesen
}

zaehlBisZehn(10);

public static void zaehlBis(int z) {
    // z wird der Wert 10 zugewiesen
}
```

Beispiel mit 2 Parametern

```
public class Main {
    public static void main() {
        zaehlVonBis(1, 9);
        //weiterer Code
        zaehlVonBis(5, 10);
    }
    public static void zaehlVonBis(int v, int b) {
        for(int i = v; i <= b; i++) {
            System.out.println(i);
        }
    }
}
```

Beispiel mit Rückgabewert

```
public class Main {
    public static void main() {
        int x = zaehlVonBis(1, 9);
    }
    public static int zaehlVonBis(int v, int b) {
        for(int i = v; i <= b; i++) {
            System.out.println(i);
        }
        return b – v + 1;
    }
}
```

Kopf der Methode

- **public static** int zaehlVonBis(int v, int b) {

  .....

  return b – v + 1;

  }
- Möchte man keinen wert zurückgeben so kommt nach **static** das Schlüsselwort void
- Ansonsten wird void durch den gewünschten Datentyp ersetzt
- Mit return wird der Wert zurückgegeben. Das return ist Pflicht und muss erreicht werden

Was passiert?

- int x = zaehlVonBis(1, 9);
- Rechte Seite von „=“ wird zuerst ausgewertet
- ZaehlVonBis(1, 9);
- public static int zaehlVonBis(int v, int b) {

  .....

  return b – v + 1;

  }
- v = 1, b = 9 => return 9 – 1 + 1 = 9 => 9 wird zurückgegeben
- x wird 9 zugewiesen

Methodenkopf

- **public static Rückgabetyp**
  Name(Parameter) {
  //Methodenrumpf
  //return falls nötig
  }

Beispiel

```
public class Main {
    public static void main() {
        int x = zaehlVonBis(1, 9);
    }
    public static int zaehlVonBis(int v, int b) {
        for(int i = v; i <= b; i++) {
            System.out.println(i);
        }
        boolean wurdeAusgegeben =
            b >= v;
        return b – v + 1;
    }
}
```

Klassenvariablen

- Probleme
  - Methoden können nur einen Wert zurückgeben
  - Eine Methode kann nicht auf Daten aus anderen Methoden zugreifen

Klassenvariablen

- Lösung: Klassenvariablen (mit Bedacht verwenden)

Beispiel

```
public class Main {
    public static boolean wurdeAusgegeben;
    public static void main() {
        int x = zaehlVonBis(1, 9);
        System.out.println(wurdeAusgegeben);
    }
    public static int zaehlVonBis(int v, int b) {
        //Ausgabe
        wurdeAusgegeben =
            b >= v;
        return b – v + 1;
    }
}
```

Deklaration von Klassenvariablen

- Deklaration direkt nach Klassendeklaration
- **public static** Datentyp Bezeichner;
- sichtbar in der ganzen Klasse
- Ohne manuelle Zuweisung wird der Defaultwert Zugewiesen
- zB(public static boolean wurdeAusgegeben;)