

Programmiervorkurs für Erstsemester

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Daniel Hoff

- ▶ 5. Semester Informatik
- ▶ aktiver Fachschafter
- ▶ gewählter studentischer Vertreter im Fakultätsrat

Beispiel ohne Methoden

```
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
- ▶ (zu) viel Code
- ▶ unübersichtlich
- ▶ Änderungen kosten noch mehr Zeit
- ▶ Code oft nicht wiederverwendbar
- ▶ Arbeitsteilung kaum möglich

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Probleme?

- ▶ zeitaufwändig
- ▶ (zu) viel Code
- ▶ unübersichtlich
- ▶ Änderungen kosten noch mehr Zeit
- ▶ Code oft nicht wiederverwendbar
- ▶ Arbeitsteilung kaum möglich

Lösungen?

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Probleme?

- ▶ zeitaufwändig
- ▶ (zu) viel Code
- ▶ unübersichtlich
- ▶ Änderungen kosten noch mehr Zeit
- ▶ Code oft nicht wiederverwendbar
- ▶ Arbeitsteilung kaum möglich

Lösungen?

- ▶ ähnlichen Code auslagern
- ▶ wiederverwendbaren Code schreiben
- ▶ **Methoden!**

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Beispiel mit Methoden

```
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);  
        }  
    }  
}
```

```
public static void zaehlBisZehn () {  
    ...  
}
```

- ▶ **public static** immer am Anfang (wird im Vorkurs nicht behandelt)
- ▶ **Methodenname** vor den runden Klammern

Aufruf einer Methode

```
public static void Main () {  
    zaehlBisZehn ();  
}
```

- ▶ Methodenname
- ▶ ()
- ▶ ;

Methoden können mehr!

- ▶ Beim Methodenaufruf können zusätzliche Informationen (= Parameter) an die Methode übergeben werden
- ▶ Methoden können Informationen an den Aufrufer zurück geben
- ▶ Methoden können sich selbst aufrufen (= Rekursion) (nicht Teil des Vorkurses)

Beispiel ohne Methoden

```
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 mit Methoden

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

```
public static void zaehleBis(int z) {  
    ...  
}
```

- ▶ In die Runden Klammern kommen die Paramter
- ▶ Parameter werden mit Komma getrennt:

(int a , boolean b , double c)

- ▶ Ein Parameter besteht aus **Datentyp** und **Bezeichner**

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

```
public static void main() {  
    zaehleBis(9);  
    zaehleBis(10);  
}
```

- ▶ Parameter die man übergeben möchte durch Komma getrennt in die Runden Klammern

Was passiert?

► `zaehleBis(9);`

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

► `zaehleBis(10);`

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

Beispiel mit 2 Parametern

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

```
public static int zaehleVonBis(int v, int b) {  
    ...  
    return b - v + 1;  
}
```

- ▶ Möchte man keinen Wert zurück geben, 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 = zaehleVonBis(1, 9);
```

- ▶ Rechte Seite von "=" wird zuerst ausgewertet
 - ▶ zaehleVonBis(1, 9);

```
public static int zaehleVonBis(int v, int b) {  
    ...  
    return b - v + 1;  
}
```

- ▶ $v = 1, b = 9$
 - ▶ $\text{return } 9 - 1 + 1 = 9$
 - ▶ 9 wird zurückgegeben
- ▶ x wird der Wert 9 zugewiesen

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

```
public static Rückgabewert Name (Parameter) {  
    // Methodenrumpf  
    return ... ;  
}
```

- ▶ Wenn der Rückgabewert **void** ist, darf kein **return** vorkommen

Beispiel ohne Klassenvariablen

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

Probleme

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

Probleme

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

Lösungen

Probleme

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

Lösungen

- ▶ **Klassenvariablen**
 - ▶ mit Bedacht verwenden!

Beispiel mit Klassenvariablen

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

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit

Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

- ▶ Deklaration direkt nach Klassendeklaration
- ▶ **public static** Datentyp Bezeichner;
- ▶ sichtbar in der ganzen Klassenvariable
- ▶ Ohne manuelle Zuweisung wird der Defaultwert zugewiesen
- ▶ Beispiel:
public static boolean wurdeAusgegeben;

Vorstellung der Aufgabe

Programmiervorkurs
für Erstsemester

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz

Quellen und Lizenz

Programmiervorkurs
für Erstsemester

Über mich

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit
Rückgabewert

Klassenvariablen

Warum?

Aufgabe

Quellen & Lizenz