

Programmierkurs

Einführung in Java

Tag 4

Torben Krüger

Wintersemester 2016/2017

Ablauf

Methoden

Warum?

Wie?

Mehr!

- Methoden mit Parameter
- Methoden mit Rückgabewert
- Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

- addNewTicket
- getSum
- calculateNewSum
- resetSum
- insertMoney
- getAmountLeft
- getChangeAmount
- getChangeCoins
- beginPayment

Quellen & Lizenz

Inhaltsübersicht Vorkurs

- ▶ Tag 1: Zustände, Variablen, Datentypen, Konvertierungen, Arithmetik, Eclipse Livedemo
- ▶ Tag 2: Kommentare, Boolesche Ausdrücke, If-Abfragen, Switch-Case
- ▶ Tag 3: Arrays, (Do-)While-Schleife, For-Schleifen, Weiterführung Debugging
- ▶ Tag 4: (statische) Methoden, Klassenvariablen

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz



Ablauf

- ▶ 09:30 Vorstellung der Lösungen des Vortages
- ▶ ab 10:00 Vorlesung
- ▶ 60 min Mittagspause
- ▶ um 12:30 Übungen

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

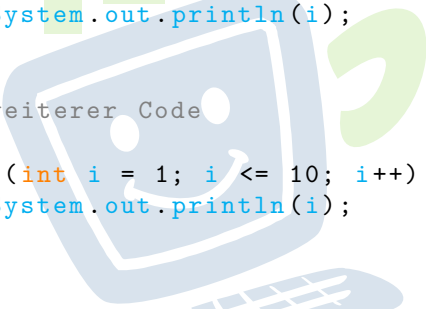
Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz

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);  
        }  
    }  
}
```



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

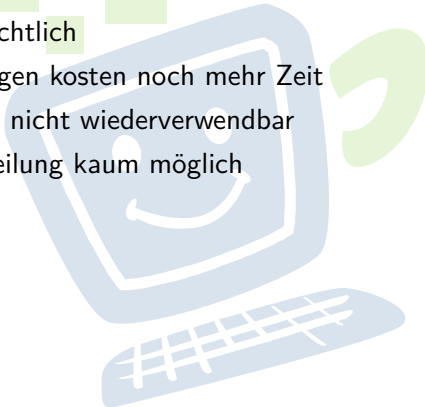
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Warum ist der Code problematisch?

Probleme:

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



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

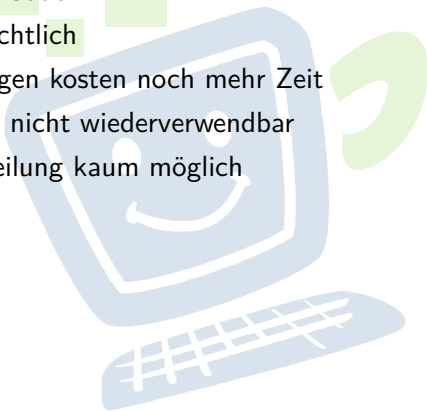
Quellen & Lizenz

Warum ist der Code problematisch?

Probleme:

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

Lösungen?



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz

Warum ist der Code problematisch?

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!**

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

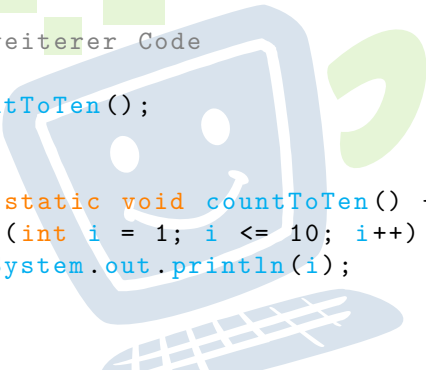
Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz

Beispiel mit Methoden

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



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Kopf der Methode

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

- ▶ **public static** immer am Anfang (wird im Vorkurs nicht behandelt)
- ▶ **Rückgabetyt** was gibt mir die Methode zurück
- ▶ **Methodenname** vor den runden Klammern

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

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)

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz

Aufruf einer Methode

```
public static void main() {  
    countToTen();  
    int x = get42();  
    int y = doubleValue(x);  
}
```

- ▶ Methodenname
- ▶ (Parameter)
- ▶ ;

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

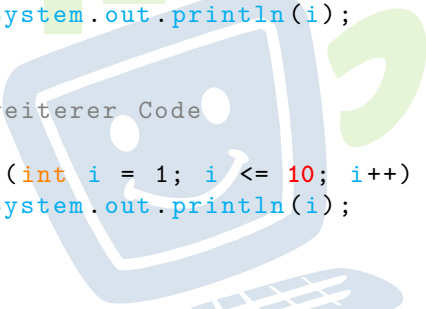
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

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);  
        }  
    }  
}
```



Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

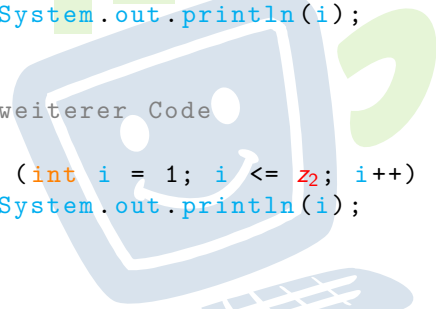
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Beispiel ohne Methoden

```
public class Main () {  
    public static void main() {  
        for (int i = 1; i <= z1; i++) {  
            System.out.println(i);  
        }  
  
        // weiterer Code  
  
        for (int i = 1; i <= z2; i++) {  
            System.out.println(i);  
        }  
    }  
}
```



Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

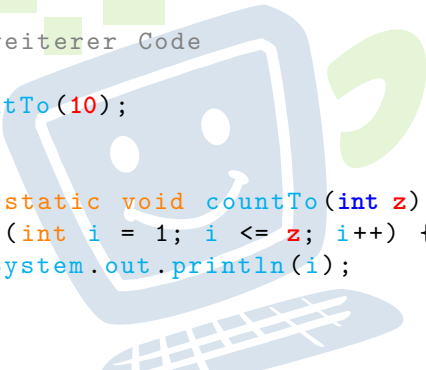
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Beispiel mit Methoden

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



Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Kopf der Methode

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

- ▶ In die runden Klammern kommen die Parameter
- ▶ Parameter werden mit Komma getrennt:
(int a, boolean b, double c)
- ▶ Ein Parameter besteht aus **Datentyp** und **Bezeichner**

Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Aufruf

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

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

Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Was passiert?

▶ `countTo(9);`

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

▶ `countTo(10);`

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

Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

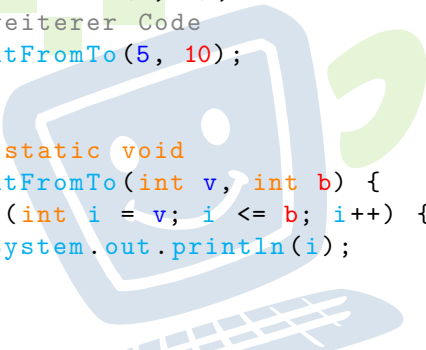
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Beispiel mit 2 Parametern

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



Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

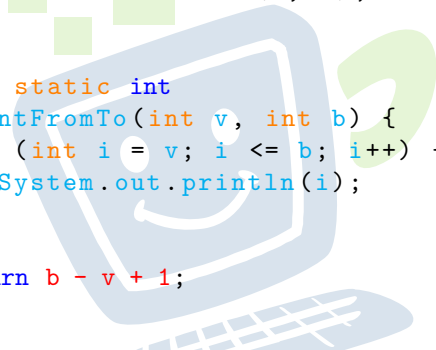
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Beispiel mit Rückgabewert

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



Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Parameter

Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Kopf der Methode

```
public static int countFromTo(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

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

Quellen & Lizenz

Was passiert?

```
int x = countFromTo(1, 9);
```

- ▶ Rechte Seite von "=" wird zuerst ausgewertet
 - ▶ `countFromTo(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

Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Parameter

Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Generell

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

- ▶ Wenn der Rückgabetyyp **void** ist, darf das **return** keinen Rückgabewert haben und ist optional

Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Parameter

Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

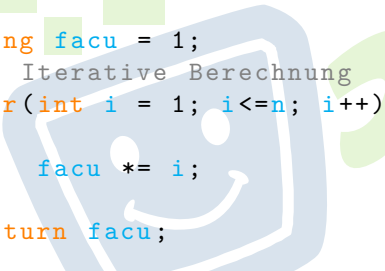
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Lesbarkeit durch Methoden

- Berechnung der Fakultät: $n! = 1 \times 2 \times \dots \times n$

```
static long fac(int n)
{
    long facu = 1;
    // Iterative Berechnung
    for(int i = 1; i<=n; i++)
    {
        facu *= i;
    }
    return facu;
}
```



- Um Fakultät zu berechnen benötigt man eine Schleife

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Binomialkoeffizient berechnen

$$\binom{n}{k} = \frac{n!}{(n-k)! \times k!}$$

- ▶ Für die Berechnung braucht man 3 Mal eine Fakultät
- ▶ Somit 3 Schleifen in der Berechnung
- ▶ Code unlesbar

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

Quellen & Lizenz

Binomialkoeffizient berechnen

$$\binom{n}{k} = \frac{n!}{(n-k)! \times k!}$$

- ▶ Für die Berechnung braucht man 3 Mal eine Fakultät
- ▶ Somit 3 Schleifen in der Berechnung
- ▶ Code unlesbar

Lösung

- ▶ Methode `long fak(int n)` benutzen

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

`addNewTicket`

`getSum`

`calculateNewSum`

`resetSum`

`insertMoney`

`getAmountLeft`

`getChangeAmount`

`getChangeCoins`

`beginPayment`

Quellen & Lizenz

Binomialkoeffizient berechnen

$$\binom{n}{k} = \frac{n!}{(n-k)! \times k!}$$

- ▶ Für die Berechnung braucht man 3 Mal eine Fakultät
- ▶ Somit 3 Schleifen in der Berechnung
- ▶ Code unlesbar

Lösung

- ▶ Methode `long fak(int n)` benutzen

```
static double calculateBinominalCoefficient(int n, int k) {  
    double bin = fak(n) / (fak(n - k) * fak(k));  
    return bin;  
}
```

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

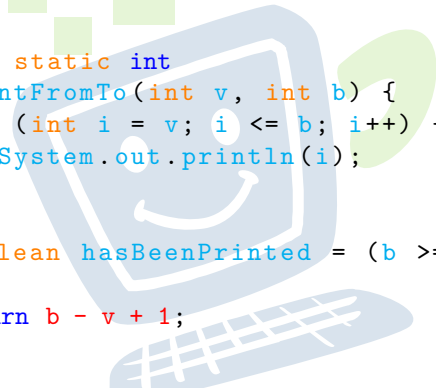
Methoden

`addNewTicket`
`getSum`
`calculateNewSum`
`resetSum`
`insertMoney`
`getAmountLeft`
`getChangeAmount`
`getChangeCoins`
`beginPayment`

Quellen & Lizenz

Beispiel ohne Klassenvariablen

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



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

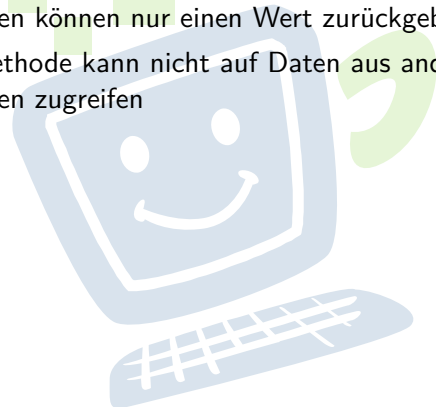
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz



Warum?

Probleme

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

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

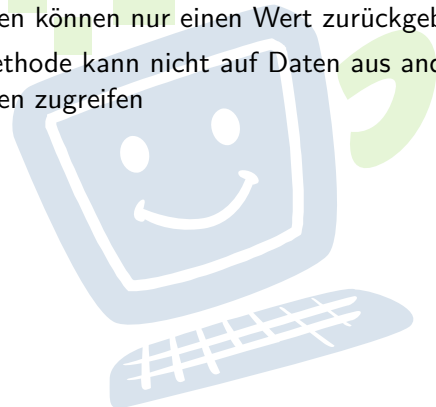


Warum?

Probleme

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

Lösung



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz



Warum?

Probleme

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

Lösung

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

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz

Beispiel mit Klassenvariablen

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

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit Rückgabewert

Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

Quellen & Lizenz

Deklaration von Klassenvariablen

- ▶ Deklaration direkt nach Klassendeklaration
- ▶ **public static** Datentyp Variablenname;
- ▶ sichtbar in der ganzen Klasse
- ▶ Ohne manuelle Zuweisung wird der Defaultwert zugewiesen
- ▶ Beispiel:

```
public static boolean hasBeenPrinted;
```

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Vorstellung der Aufgabe

Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bitte wählen Sie einen Zielbahnhof aus:

München Frankfurt Weiter...

Tickets:
München
Frankfurt
München (Bahncard)
Frankfurt (Bahncard)

Gesamtpreis: 153,82 EUR Alle Tickets löschen

Bezahlen

Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bitte werfen Sie Geld ein.
Noch zu zahlen: 153,82 EUR

20€ 10€ 100€
50€ 100€

Bezahlen

Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bitte werfen Sie Geld ein.
Rückgeld: 6,18 EUR

1€ 1€ 1€
1€ 1€

Rückgeld:

3x		1x	
0x		1x	
0x		1x	
0x		1x	

Bezahlen

Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz


Vorstellung der Aufgabe

Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bezahlen

Bitte wählen Sie einen Zielbahnhof aus:

 25

München Frankfurt Weiter...

Tickets:
München
Frankfurt
München (Bahncard)
Frankfurt (Bahncard)

Gesamtpreis: 153,82 EUR

Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bezahlen

Bitte werfen Sie Geld ein.

Noch zu zahlen: 153,82 EUR



Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bezahlen

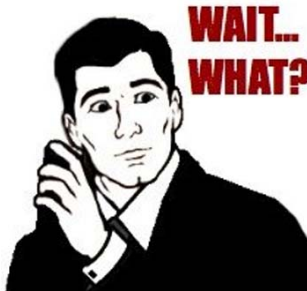
Bitte werfen Sie Geld ein.

Rückgeld: 6,18 EUR



Rückgeld:

3x		1x	
0x		1x	
0x		1x	
0x		1x	



Ablauf

Methoden

Warum?
Wie?
Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

```
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment
```

Quellen & Lizenz

Methoden

- ▶ `public static String[] addNewTicket(String[] oldTickets, String newTicket, boolean bahncard)`
- ▶ `public static void calculateNewSum(int distance, boolean bahncard)`
- ▶ `public static double getSum()`
- ▶ `public static void resetSum()`
- ▶ `public static void beginPayment()`
- ▶ `public static void insertMoney(int amount)`
- ▶ `public static double getAmountLeft()`
- ▶ `public static boolean isAmountLeft()`
- ▶ `public static double getChangeAmount()`
- ▶ `public static int[] getChangeCoins()`

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment
```

Quellen & Lizenz

addNewTicket

- ▶ Parameter:
 - ▶ oldTickets: String Array enthält die alten Tickets
 - ▶ newTicket: String
 - ▶ bahncard: boolean
- ▶ Beschreibung:

Soll den Parameter **oldTickets** um den Parameter **newTicket** erweitern und für den Fall dass der Parameter **bahncard true** ist um „(Bahncard)“ erweitern und anschließend zurück geben.

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz



- ▶ Hat keine Parameter
- ▶ Gibt den aktuellen Gesamtpreis als double in Euro zurück

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz

calculateNewSum

- ▶ Parameter:
 - ▶ distance: integer
 - ▶ bahncard: boolean
- ▶ Beschreibung: Berechnet den Gesamtpreis aller bisher sowie dem aktuell ausgewählten Ticket. Dabei gilt:
 - ▶ Bis 200km: 10 €+ 0.20 €pro km
 - ▶ Ab 200km: 5 €+ 0.15 €pro km
 - ▶ Mit einer Bahncard erhält man immer 25% Rabatt
- ▶ Hat keinen Rückgabewert

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden


addNewTicket
getSum

resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz



resetSum

- ▶ Hat keine Parameter
 - ▶ Hat keinen Rückgabewert
 - ▶ Setzt den Gesamtpreis auf 0 zurück
- 

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

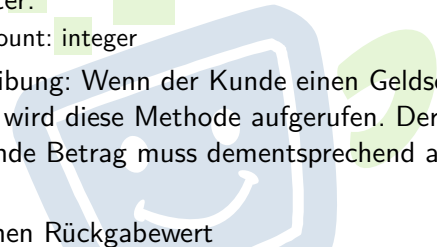
addNewTicket
getSum
calculateNewSum

insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz



insertMoney

- ▶ Parameter:
 - ▶ amount: integer
 - ▶ Beschreibung: Wenn der Kunde einen Geldschein einwirft wird diese Methode aufgerufen. Der noch zu bezahlende Betrag muss dementsprechend angepasst werden
 - ▶ Hat keinen Rückgabewert
- 

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum

insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz



getAmountLeft

- ▶ Hat keine Parameter
- ▶ Als Rückgabewert soll hier der noch zu zahlende Betrag als double in Euro zurück gegeben werden

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney

getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

getChangeAmount

- ▶ Hat keine Parameter
- ▶ Wird aufgerufen, wenn der Kunde ausreichend Geld eingeworfen hat
- ▶ Gibt den Betrag des Wechselgeld zurück (in Euro). Dieser muss positiv sein

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft
```

```
getChangeCoins  
beginPayment
```

Quellen & Lizenz

getChangeCoins

- ▶ Hat keine Parameter
- ▶ Beschreibung: Berechnet, wie viele Münzen von jeder Sorte der Kunde zurück bekommt.
- ▶ Gibt ein Array zurück, das die Anzahl der entsprechenden Münzen enthält
 - ▶ Rückgabe[0]: enthält die Anzahl der 2 €-Münzen
 - ▶ Rückgabe[1]: enthält die Anzahl der 1 €-Münzen
 - ▶ ...
 - ▶ Rückgabe[7]: enthält die Anzahl der 1 ct-Münzen

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount

beginPayment

Quellen & Lizenz

beginPayment (OPTIONAL)

- ▶ Diese Methode ist optional
- ▶ Wird nicht immer benötigt
- ▶ Kann für spezielle Aktionen zu Beginn des Bezahlvorgangs verwendet werden

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins
```

Quellen & Lizenz

Quellen und Lizenz



FACHSCHAFT INFORMATIK

HS Karlsruhe



- ▶ Original von Samuel Zeitvogel
- ▶ Überarbeitet 2012 von Daniel Hoff
- ▶ Überarbeitet 2013 von Tristan Wagner
- ▶ Überarbeitet 2014 von Torben Krüger
- ▶ Überarbeitet 2015 von Tobias Kerst
- ▶ Überarbeitet 2016 von Christian Wernet
- ▶ Überarbeitet 2017 von Torben Krüger

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert
Beispiel Lesbarkeit

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

```
addNewTicket  
getSum  
calculateNewSum  
resetSum  
insertMoney  
getAmountLeft  
getChangeAmount  
getChangeCoins  
beginPayment
```

Quellen & Lizenz