

Programmievorkurs

Einführung in Java

Tag 4

Niko Strugar
WS 2014/2015



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

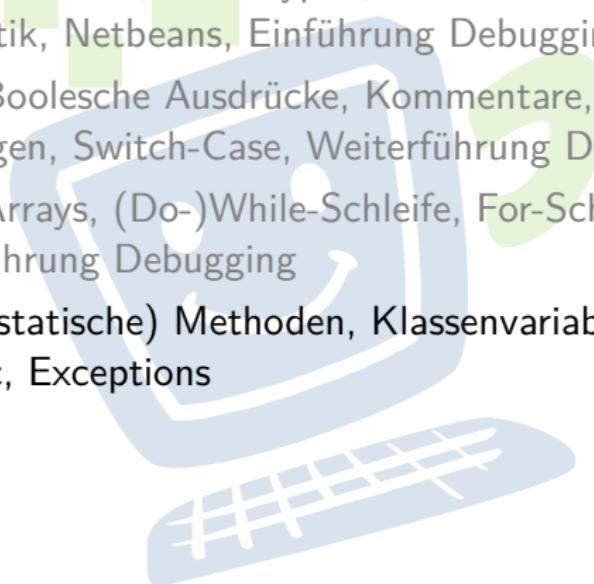
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Inhaltsübersicht Vorkurs

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



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

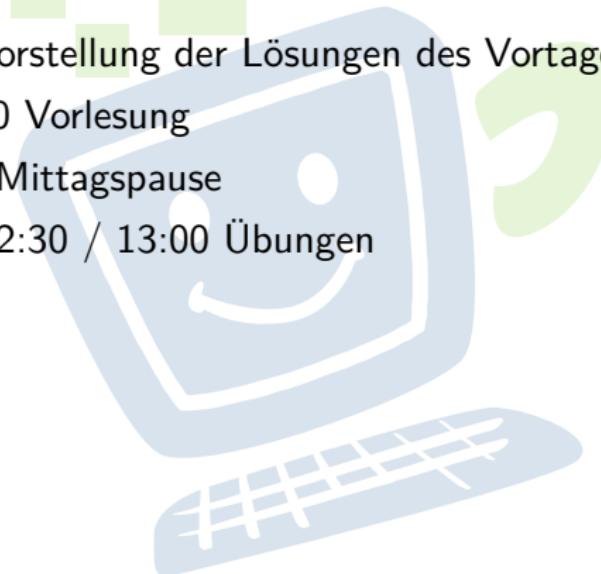
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Ablauf

- ▶ 09:30 Vorstellung der Lösungen des Vortages
- ▶ ab 10:00 Vorlesung
- ▶ 90 min Mittagspause
- ▶ gegen 12:30 / 13:00 Übungen



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 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

KlassenvARIABLEN

Warum?

Wie?

Aufgabe

Methoden

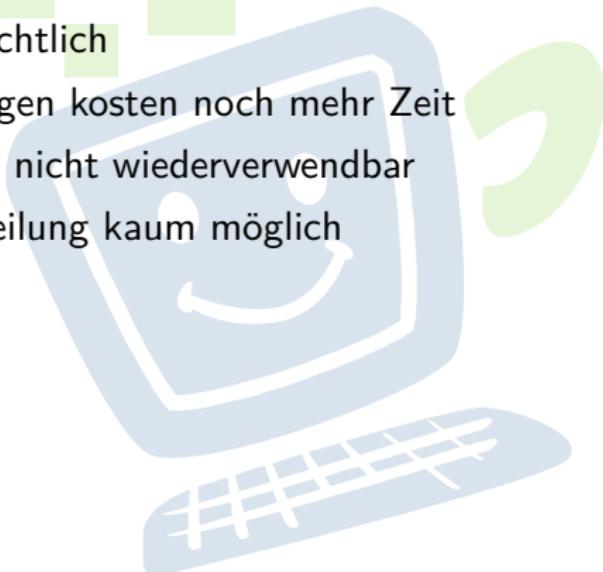
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Warum?

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

KlassenvARIABLEN

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

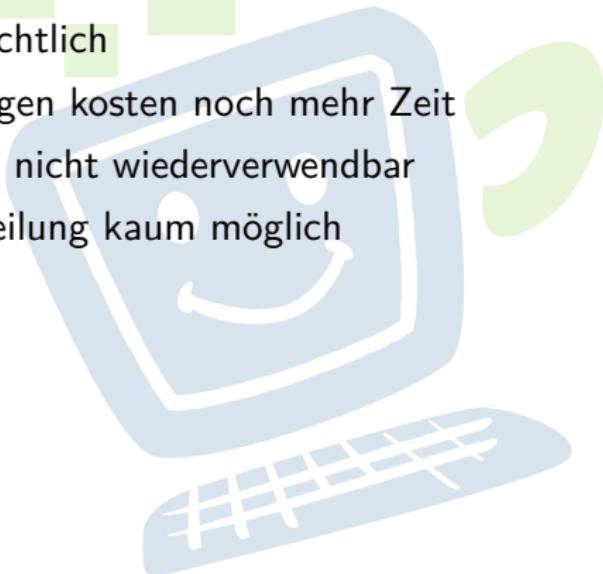
Quellen & Lizenz

Warum?

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

KlassenvARIABLEN

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Warum?

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

KlassenvARIABLEN

Warum?

Wie?

Aufgabe

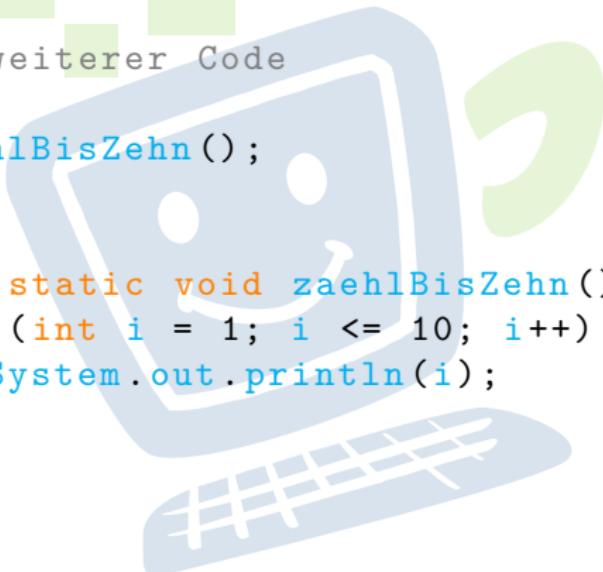
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Beispiel mit Methoden

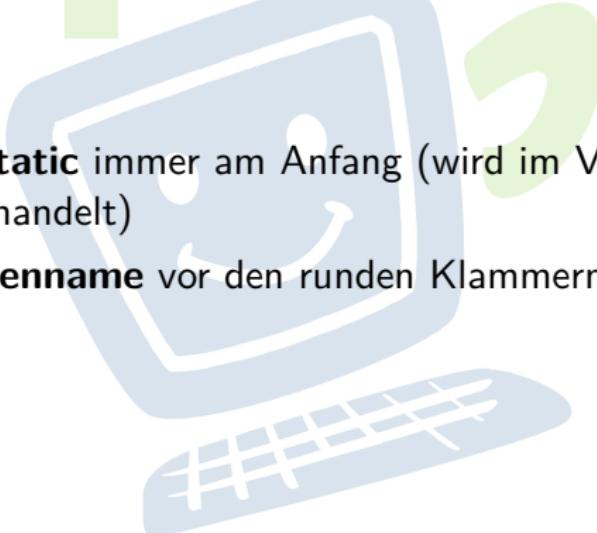
```
public class Main {  
    public static void main () {  
        zaehlBisZehn();  
        // weiterer Code  
  
        zaehlBisZehn();  
    }  
  
    public static void zaehlBisZehn() {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```



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

Kopf der Methode

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



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

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

Aufruf einer Methode

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

- ▶ Methodename
- ▶ ()
- ▶ ;

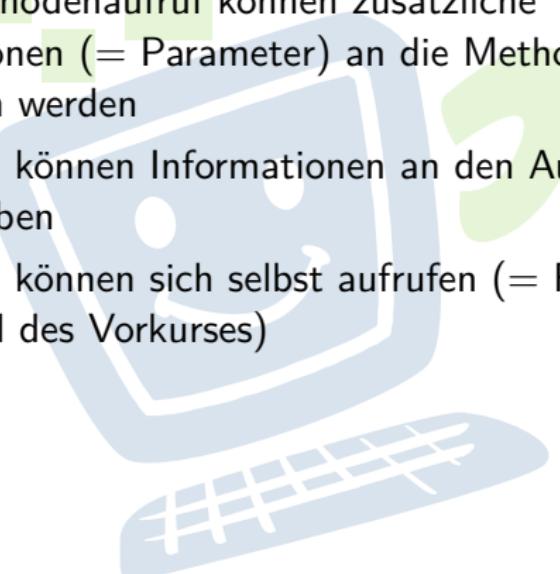


- Ablauf
- Methoden
 - Warum?
 - Wie?
 - Mehr!
- Methoden mit Parameter
- Methoden mit Rückgabewert
- Klassenvariablen
 - Warum?
 - Wie?
- Aufgabe
- Methoden
 - addNewTicket
 - getSum
 - calculateNewSum
 - resetSum
 - insertMoney
 - getAmountLeft
 - getChangeAmount
 - getChangeCoins
 - beginPayment

Quellen & Lizenz

Methoden können mehr!

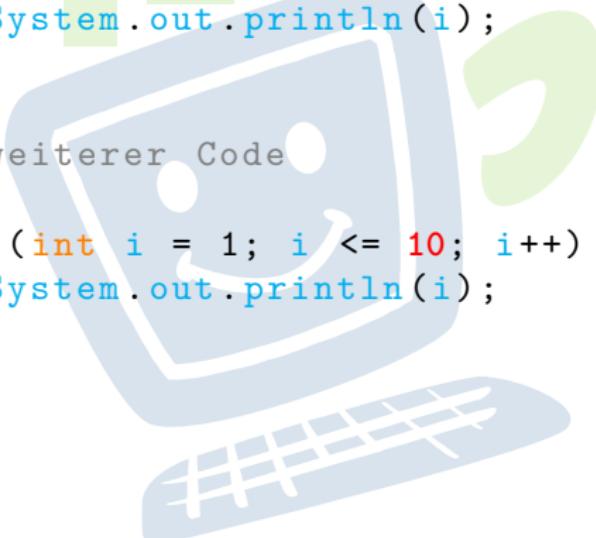
- ▶ Beim Methodenaufruf können zusätzliche Informationen (= Parameter) an die Methode übergeben werden
- ▶ Methoden können Informationen an den Aufrufer zurück geben
- ▶ Methoden können sich selbst aufrufen (= Rekursion) (nicht Teil des Vorkurses)



Ablauf
Methoden
Warum?
Wie?
Mehr!
Methoden mit Parameter
Methoden mit Rückgabewert
KlassenvARIABLEN
Warum?
Wie?
Aufgabe
Methoden
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment
Quellen & Lizenz

Beispiel ohne Methoden

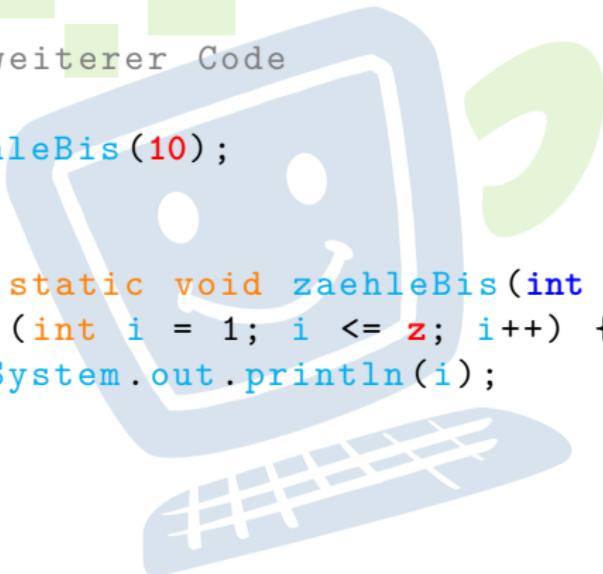
```
public class Main () {  
    public static void main() {  
        for (int i = 1; i <= 9; i++) {  
            System.out.println(i);  
        }  
  
        // weiterer Code  
  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```



Ablauf
Methoden
Warum?
Wie?
Mehr!
Methoden mit Parameter
Methoden mit Rückgabewert
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() {  
        zaehleBis(9);  
        // weiterer Code  
        zaehleBis(10);  
    }  
  
    public static void zaehleBis(int z) {  
        for (int i = 1; i <= z; i++) {  
            System.out.println(i);  
        }  
    }  
}
```



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

Kopf der Methode

```
public static void zaeheleBis(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 Parameter
Methoden mit Rückgabewert
Klassenvariablen
Warum?
Wie?
Aufgabe
Methoden
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Aufruf

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

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

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

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  
}
```

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

`addNewTicket`

`getSum`

`calculateNewSum`

`resetSum`

`insertMoney`

`getAmountLeft`

`getChangeAmount`

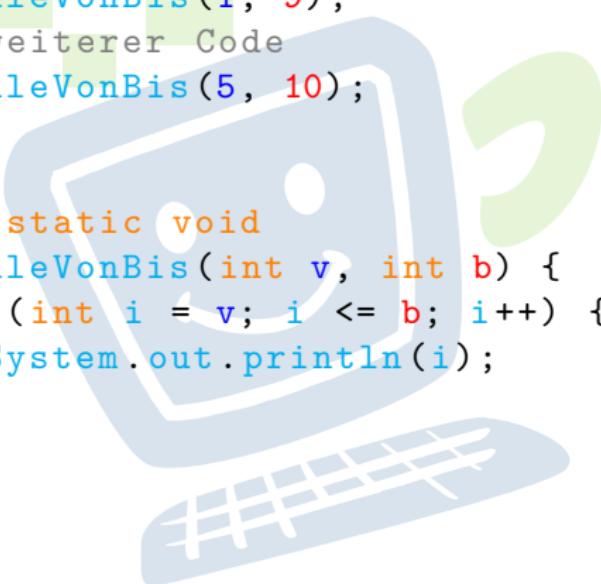
`getChangeCoins`

`beginPayment`

Quellen & Lizenz

Beispiel 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);  
            }  
        }  
}
```



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 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;  
        }  
}
```

- Ablauf
- Methoden
 - Warum?
 - Wie?
 - Mehr!
- Methoden mit Parameter
 - Methoden mit Parameter
- Klassenvariablen
 - Warum?
 - Wie?
- Aufgabe
- Methoden
 - addNewTicket
 - getSum
 - calculateNewSum
 - resetSum
 - insertMoney
 - getAmountLeft
 - getChangeAmount
 - getChangeCoins
 - beginPayment
- Quellen & Lizenz

Kopf der Methode

```
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

Ablauf Methoden

Warum?
Wie?
Mehr!

Methoden mit Parameter
Methoden mit Rücksprung

KlassenvARIABLEN

Warum?
Wie?

Aufgabe

Methoden
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

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;
}
```

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

Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter

Methoden mit Klasseneigenschaft

Klasseneigenschaften

Warum?

Wie?

Aufgabe

Methoden

`addNewTicket`

`getSum`

`calculateNewSum`

`resetSum`

`insertMoney`

`getAmountLeft`

`getChangeAmount`

`getChangeCoins`

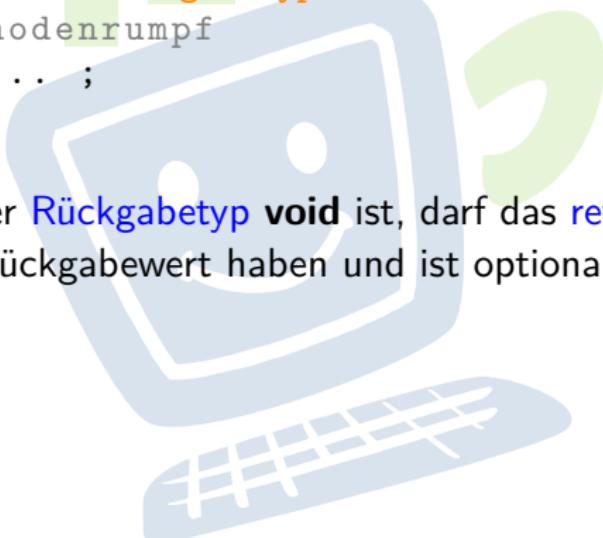
`beginPayment`

Quellen & Lizenz

Generell

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

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



- Ablauf
- Methoden
 - Warum?
 - Wie?
 - Mehr!
- Methoden mit Parameter
 - Methoden mit Parameter
- Klassenvariablen
 - Warum?
 - Wie?
- Aufgabe
- Methoden
 - addNewTicket
 - getSum
 - calculateNewSum
 - resetSum
 - insertMoney
 - getAmountLeft
 - getChangeAmount
 - getChangeCoins
 - beginPayment
- Quellen & Lizenz

Beispiel ohne Klassenvariablen

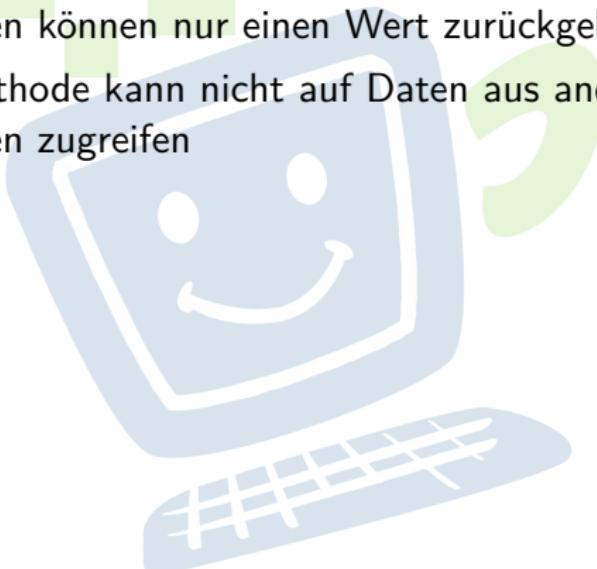
```
public class Main {  
    public static void main() {  
        int x = zaehleVonBis(1, 9);  
    }  
  
    public static int  
        zaehleVonBis(int v, int b) {  
            for (int i = v; i <= b; i++) {  
                System.out.println(i);  
            }  
  
            boolean wurdeAusgegeben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

- 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

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

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

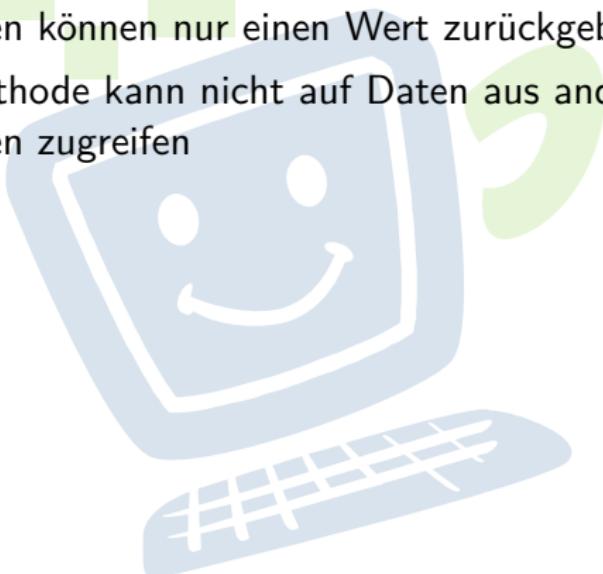
Quellen & Lizenz

Warum?

Probleme

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

Lösungen



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

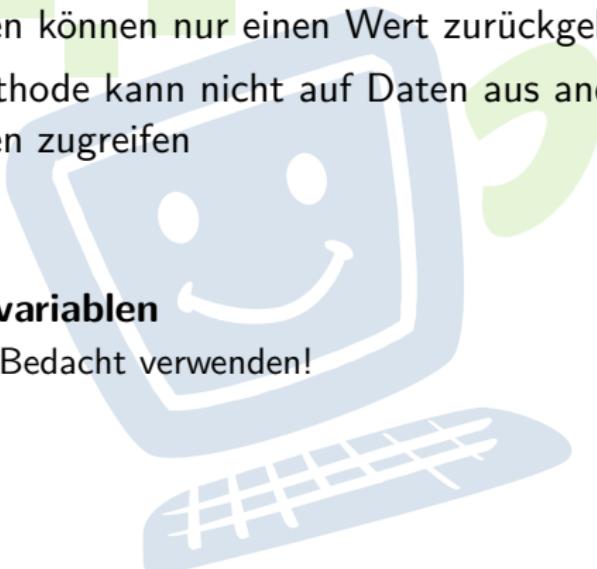
Warum?

Probleme

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

Lösungen

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



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

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 wurdeAusgegeben;
```



- 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 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);  
        }  
        this.wurdeAusgegeben = b >= v;  
        return b - v + 1;  
    }  
}
```

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

Vorstellung der Aufgabe

Abfahrtbahnhof: Karlsruhe HBF

Zielbahnhof: < Bitte auswählen >

Bezahlen

Bitte wählen Sie einen Zielbahnhof aus:



München Frankfurt Weitere ...

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

Gesamtpreis: 153,82 EUR

Alle Tickets löschen

Abfahrtbahnhof: Karlsruhe HBF

Zielbahnhof: < Bitte auswählen >

Bezahlen

Bitte werfen Sie Geld ein.

Noch zu zahlen: 153,82 EUR





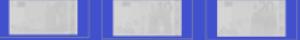
Abfahrtbahnhof: Karlsruhe HBF

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

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:

München Frankfurt Weitere ...

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

Gesamtpreis: 153,82 EUR

Alle Tickets löschen

Abfahrtsbahnhof: Karlsruhe HBF

Zielbahnhof: < Bitte auswählen >

Bezahlen

Bitte werfen Sie Geld ein.

Noch zu zahlen: 153,82 EUR

50 100 20

500 100 20

Abfahrtsbahnhof: Karlsruhe HBF

Zielbahnhof: < Bitte auswählen >

Bezahlen

Bitte werfen Sie Geld ein.

Rückgeld: 6,18 EUR

50 100 20

500 100 20

Rückgeld:

3x 2€	1x 1€
0x 50¢	1x 20¢
0x 20¢	1x 10¢
0x 10¢	1x 5¢



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

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
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 das der Parameter ***bahncard*** **true** ist um (Bahncard) erweitern und anschließend zurück geben.

Ablauf
Methoden

Warum?
Wie?
Mehr!
Methoden mit Parameter
Methoden mit Rückgabewert

KlassenvARIABLEN
Warum?
Wie?

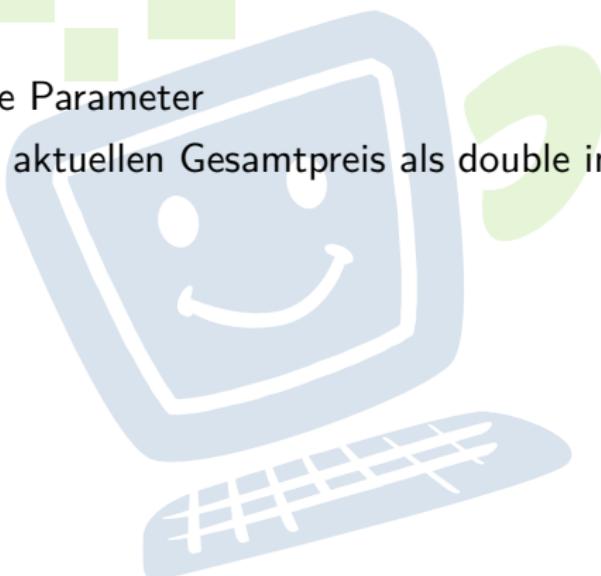
Aufgabe
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

getSum

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



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

calculateNewSum

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

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

resetSum

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



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert

KlassenvARIABLEN

Warum?

Wie?

Aufgabe

Methoden

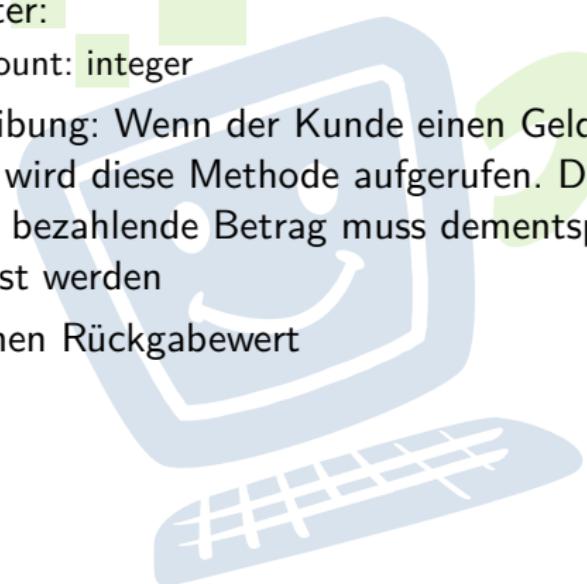
addNewTicket
getSum
calculateNewSum

insertCoin
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

insertMoney

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



Ablauf

Methoden

Warum?

Wie?

Mehr!

Methoden mit Parameter
Methoden mit Rückgabewert

KlassenvARIABLEN

Warum?

Wie?

Aufgabe

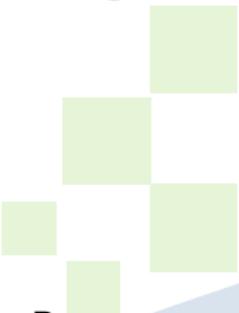
Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney

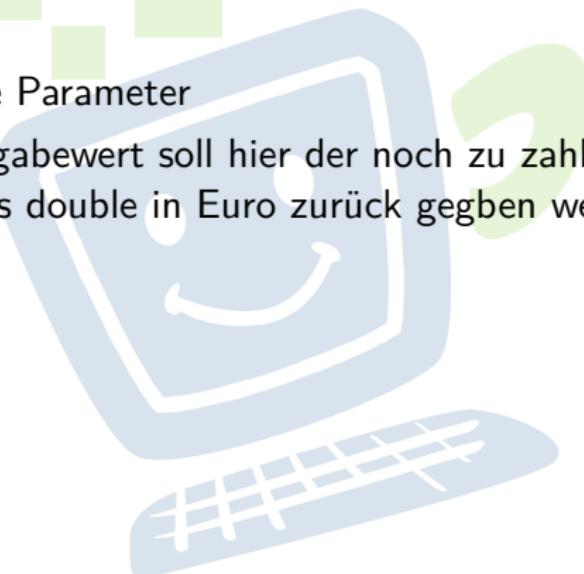
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

getAmountLeft



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

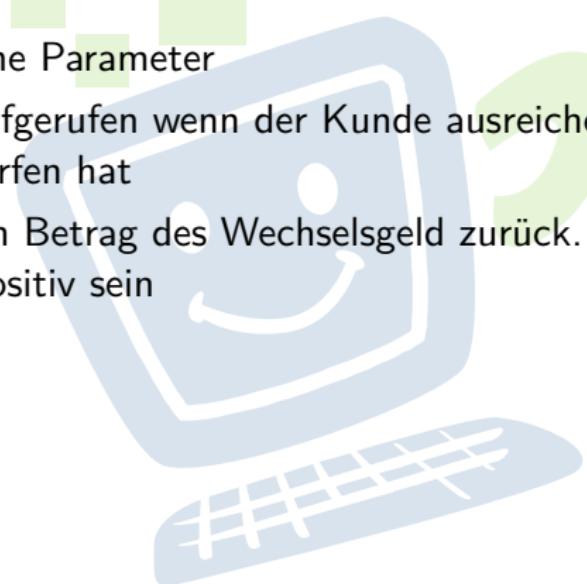


Ablauf
Methoden
Warum?
Wie?
Mehr!
Methoden mit Parameter
Methoden mit Rückgabewert
KlassenvARIABLEN
Warum?
Wie?
Aufgabe
Methoden
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

getChangeAmount

- ▶ Hat keine Parameter
- ▶ Wird aufgerufen wenn der Kunde ausreichen Geld eingeworfen hat
- ▶ Gibt den Betrag des Wechselgeld zurück. Dieser muss Positiv sein



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

getChangeCoins

- ▶ Hat keine Parameter
- ▶ Beschreibung: Berechnet wieviel Münzen von jeder Sorte der Kunde zurück bekommt.
- ▶ Gibt ein Array zurück in dem 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

KlassenvARIABLEN Warum? Wie?

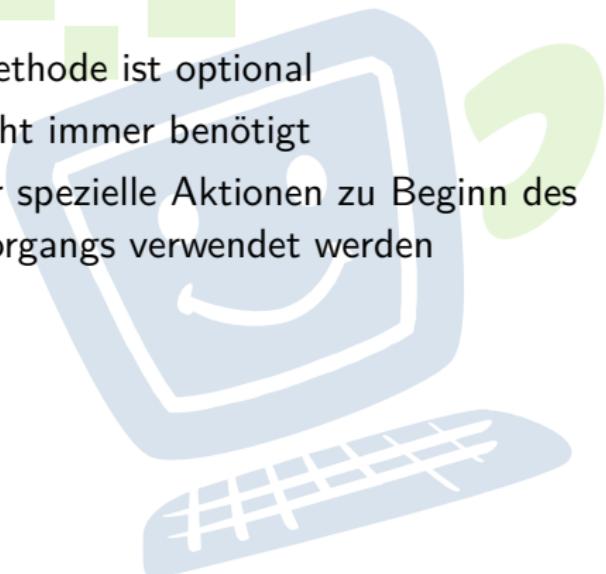
Aufgabe Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
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

Klassenvariablen

Warum?

Wie?

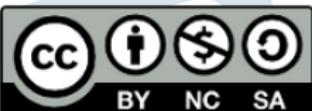
Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Quellen und Lizenz



- ▶ Original von Samuel Zeitvogel
- ▶ Überarbeitet 2012 von Daniel Hoff
- ▶ Überarbeitet 2013 von Tristan Wagner
- ▶ Überarbeitet 2014 von Niko Strugar

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