

Programmierkurs

Einführung in Java

Tag 4

Matthäus Prasse

Sommersemester 2019

Ablauf

Methoden

Warum?

Mehr!

Wie?

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: 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?
Mehr!
Wie?

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

Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

Quellen & Lizenz

Was sind Methoden?

Eine Klasse entspricht einer Fabrik

- ▶ Mehrere Arbeiter arbeiten in einer Fabrik
- ▶ Ziel ist es, ein fertiges Produkt zu erzeugen

Methoden entsprechen Arbeiter

- ▶ Warten bis sie aufgerufen werden
- ▶ Führen eine definierte Arbeit aus
- ▶ Können zusammenarbeiten
- ▶ Müssen nicht unbedingt wissen, was der andere macht

Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

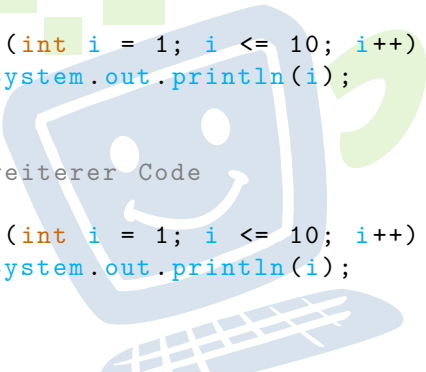
getChangeCoins

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?
Mehr!
Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

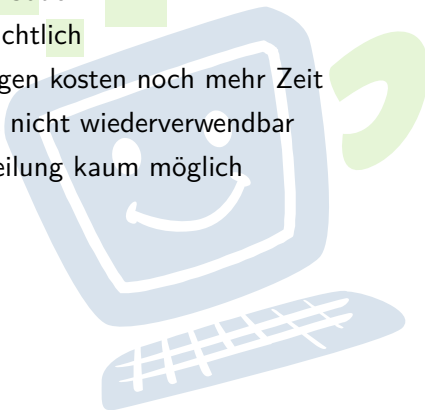
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Warum ist dieser Code problematisch?

Probleme:

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



Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

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

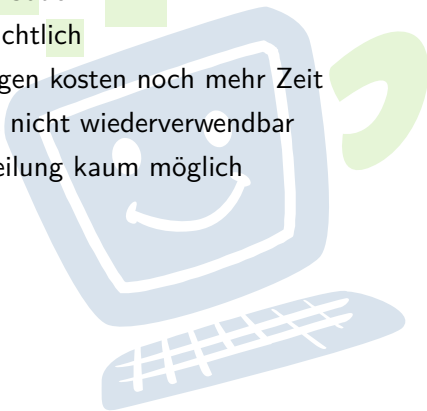
Quellen & Lizenz

Warum ist dieser Code problematisch?

Probleme:

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

Lösungen?



Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

Quellen & Lizenz

Warum ist dieser 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?

Mehr!

Wie?

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?

Mehr!

Wie?

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

Ablauf

Methoden

Warum?

Mehr!

Wie?

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 () {  
    //Code der Methode  
}
```

- ▶ **public static** immer am Anfang (wird im Vorkurs nicht behandelt)
- ▶ **Rückgabotyp** (void) was gibt mir die Methode zurück
- ▶ **Methodenname** (zaehlBisZehn) vor den runden Klammern
- ▶ **Parameter** (in dem Fall keine) in den runden Klammern

Ablauf

Methoden

Warum?

Mehr!

Wie?

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();  
    int x = gibMir42();  
    int y = verdoppelWert(x);  
}
```

- ▶ Methodenname
- ▶ (Parameter)
- ▶ ;

Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Parameter einer Methode

- ▶ Wenn Methoden Arbeiter entsprechen, sind die Parameter die Werkstoffe!

```
public static void zaehleBis(int z) {  
    //Code der Methode  
}
```

- ▶ 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?
Mehr!
Wie?

Methoden mit Rückgabewert Klassenvariablen

Warum?
Wie?

Aufgabe

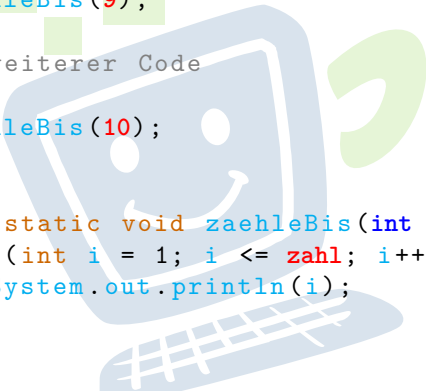
Methoden

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



Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Rückgabewert

Klassenvariablen

Warum?
Wie?

Aufgabe

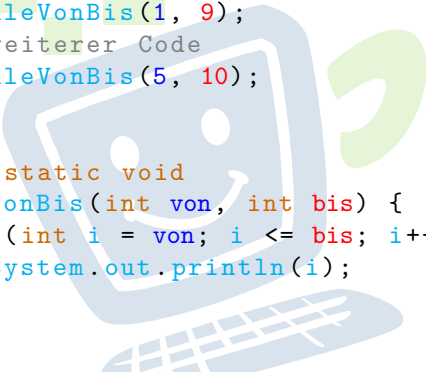
Methoden

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 von, int bis) {  
        for (int i = von; i <= bis; i++) {  
            System.out.println(i);  
        }  
    }  
}
```



Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Rückgabewert

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

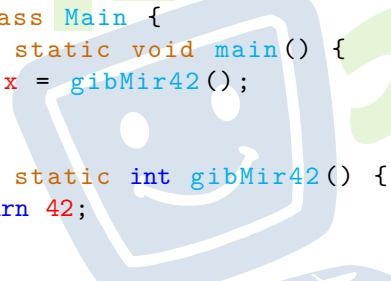
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Rückgabewerte

- ▶ Das Ergebnis was am Ende zurückgegeben wird

```
public class Main {  
    public static void main() {  
        int x = gibMir42();  
    }  
  
    public static int gibMir42() {  
        return 42;  
    }  
}
```



Ablauf

Methoden

Warum?
Mehr!
Wie?

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 gibMir42() {  
    ...  
    return 42;  
}
```

- ▶ Möchte man keinen Wert zurück geben, so kommt nach **static** das Schlüsselwort **void**
- ▶ Ansonsten wird **void** durch den gewünschten Datentyp ersetzt (z.b. **int**, **string**)
- ▶ Mit **return** wird der Wert zurückgegeben. Das **return** ist Pflicht und muss erreicht werden

Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Was passiert?

```
int x = gibMir42();
```

- ▶ Rechte Seite von "=" wird zuerst ausgewertet
 - ▶ gibMir42();

```
public static int gibMir42() {  
    return 40+2;  
}
```

- ▶ return 40 + 2
 - ▶ return 42
 - ▶ 42 wird zurückgegeben
- ▶ x wird der Wert 42 zugewiesen

Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Generell Definition

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

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

Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Generell Aufruf

```
Name (Parameter);
```

```
Variable = Name (Parameter);
```

- ▶ Entweder erfolgt der Aufruf direkt oder als Zuweisung zu einer Variable

Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz

Klassenvariablen

- ▶ Bisher haben wir Variablen nur innerhalb der main-Methode verwendet

Probleme

- ▶ Variablen existieren nur innerhalb der Methoden in der sie deklariert wurden
- ▶ Eine Methode kann nicht auf Variablen aus anderen Methoden zugreifen

Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?
Wie?

Aufgabe

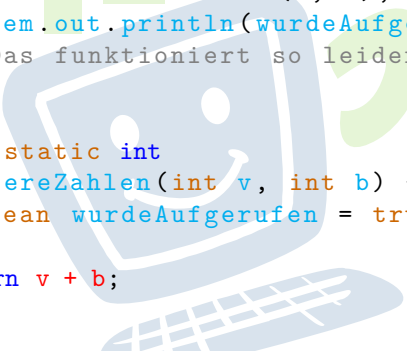
Methoden

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

Quellen & Lizenz

Beispiel ohne Klassenvariablen

```
public class Vorkurs {  
    public static void main() {  
        int x = addiereZahlen(1, 9);  
        System.out.println(wurdeAufgerufen);  
        // Das funktioniert so leider nicht  
    }  
  
    public static int  
    addiereZahlen(int v, int b) {  
        boolean wurdeAufgerufen = true;  
  
        return v + b;  
    }  
}
```



Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?
Wie?

Aufgabe

Methoden

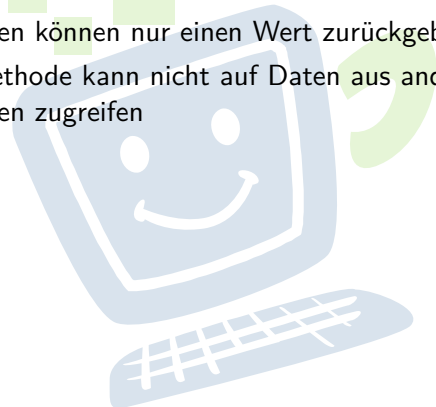
addNewTicket
getSum
calculateNewSum
resetSum
insertMoney
getAmountLeft
getChangeAmount
getChangeCoins
beginPayment

Quellen & Lizenz



Warum?

Probleme

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

Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

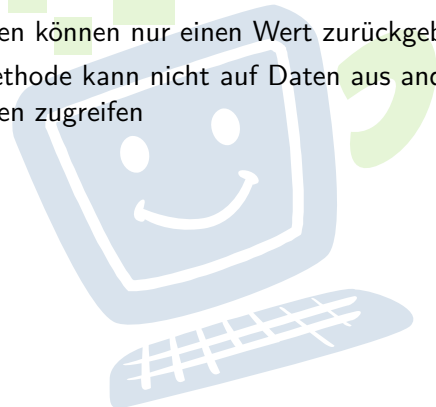
Quellen & Lizenz

Warum?

Probleme

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

Lösung



Ablauf

Methoden

Warum?
Mehr!
Wie?

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

Lösung

- ▶ **Klassenvariablen**

Ablauf

Methoden

Warum?
Mehr!
Wie?

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 Vorkurs {  
    public static boolean wurdeAufgerufen;  
  
    public static void main() {  
        int x = addiereZahlen(1, 9);  
        System.out.println(wurdeAufgerufen);  
    }  
  
    public static int  
        addiereZahlen(int v, int b) {  
  
        wurdeAufgerufen = true;  
        return v + b;  
    }  
}
```

Ablauf

Methoden

Warum?

Mehr!

Wie?

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 wurdeAufgerufen;
```

Ablauf

Methoden

Warum?

Mehr!

Wie?

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 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?
Mehr!
Wie?

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 >

Bitte wählen Sie einen Zielbahnhof aus:

München Frankfurt Weiter...

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

Gesamtpreis: 153,82 EUR

Alle Tickets löschen

Abfahrtsbahnhof:
Karlsruhe HBF

Zielbahnhof:
< Bitte auswählen >

Bitte werfen Sie Geld ein.

Noch zu zahlen: 153,82 EUR

Bezahlen

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?
Mehr!
Wie?

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?

Mehr!

Wie?

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

Ablauf

Methoden

Warum?
Mehr!
Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

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?

Mehr!

Wie?

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 bisherigen, 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?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

getAmountLeft

resetSum

insertMoney

getAmountLeft

getChangeAmount

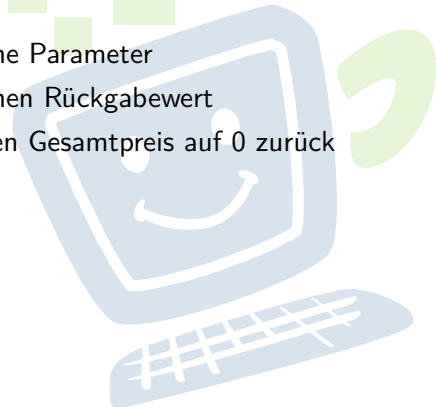
getChangeCoins

beginPayment

Quellen & Lizenz



resetSum

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

Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

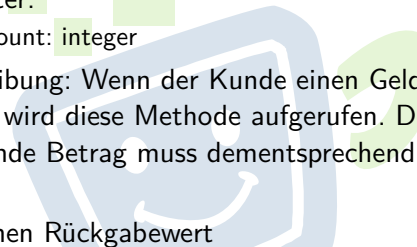
addNewTicket
getSum
calculateNewSum

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?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum

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?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket
getSum
calculateNewSum
resetSum
insertMoney

~~newMoney~~

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?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

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?

Mehr!

Wie?

Methoden mit Parameter
Methoden mit Rückgabewert

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?

Mehr!

Wie?

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

© 2014-2015

Quellen & Lizenz

Quellen und Lizenz



FACHSCHAFT INFORMATIK

HS Karlsruhe



- ▶ Original von Samuel Zeitvogel
- ▶ Überarbeitet 2012 von Daniel Hoff
- ▶ Überarbeitet 2013 von Tristan Wagner
- ▶ Überarbeitet 2015 von Tobias Kerst
- ▶ Überarbeitet 2016 von Christian Wernet
- ▶ Überarbeitet 2018 von Jan Oliver Zerulla
- ▶ Überarbeitet 2018 von Johannes Beierle
- ▶ Überarbeitet 2019 von Matthäus Prasse

Ablauf

Methoden

Warum?

Mehr!

Wie?

Methoden mit Parameter

Methoden mit Rückgabewert

Klassenvariablen

Warum?

Wie?

Aufgabe

Methoden

addNewTicket

getSum

calculateNewSum

resetSum

insertMoney

getAmountLeft

getChangeAmount

getChangeCoins

beginPayment

Quellen & Lizenz