

# Zustandsmodelle

M. Jakob

Gymnasium Pegnitz

10. Dezember 2014

## Inhaltsverzeichnis

### Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

### Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

### Bedingungen und ausgelösten Aktionen in ZÜD

## Gliederung

### Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

### Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

### Bedingungen und ausgelösten Aktionen in ZÜD

3/43

## In diesem Abschnitt

### Einführung

**Inselspiel**

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

### Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

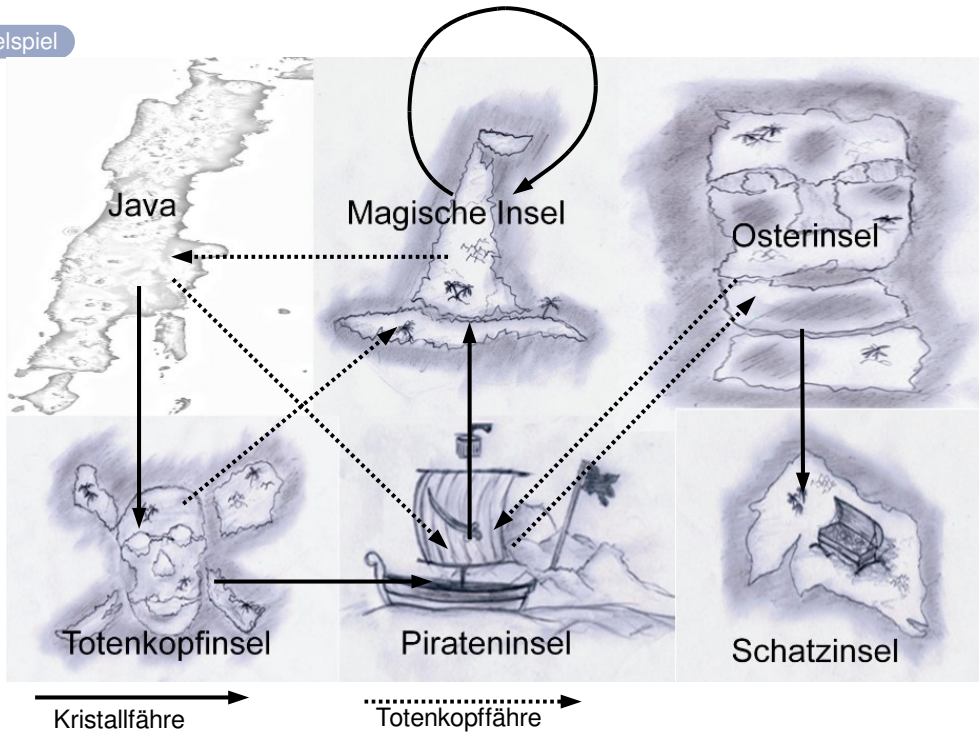
Verzweigungen

switch-case

### Bedingungen und ausgelösten Aktionen in ZÜD

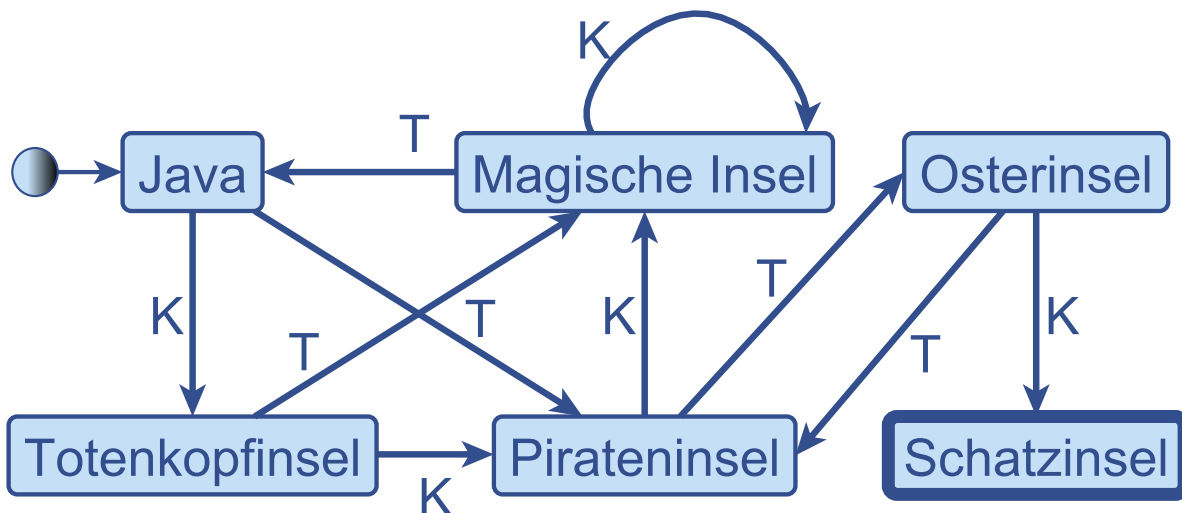
# Das Inselspiel

► Inselspiel



5/43

# Das Inselspiel — Zustandsdiagramm



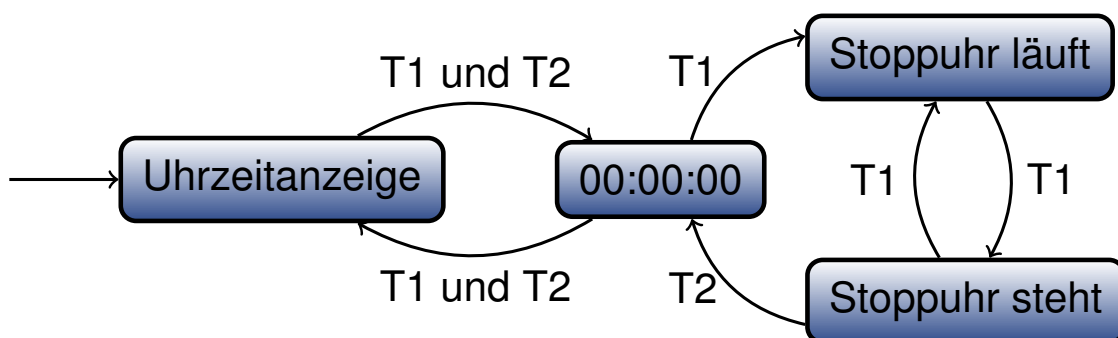
6/43

## Mikrobetriebsysteme

- ▶ Elektronische Geräte (Handy, mp3-Player,...) brauchen zur Funktion ein kleines Betriebssystem, das festlegt wie sich das Gerät verhalten soll.
- ▶ Zur Entwicklung solcher Betriebssysteme werden zuerst **Zustandsdiagramme** entworfen.

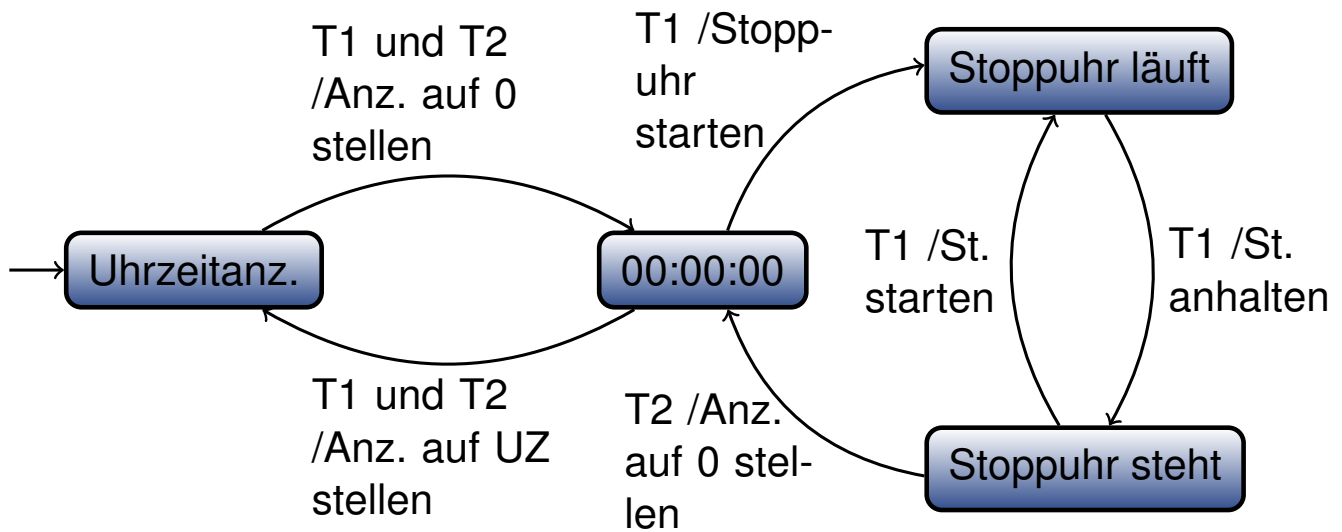
7/43

## Beispiel: Einfache Stoppuhr



8/43

## Beispiel: Einfache Stoppuhr mit ausgelösten Aktionen



9/43

## Übung

- Ü 1.1: Zustandsübergänge mit ausgelösten Aktionen, Vorlage: ZUEDÜbg.graphml  
run:Arbeitsmaterial/ZUED/ZUEDÜbg.graphml

Nachfolgend sind Zustände, Ereignisse oder ausgelöste Aktionen angegeben. Ergänze die fehlenden Bausteine

- Telefonhörer abheben.
- Eine Waschmaschine hat die Waschtemperatur erreicht.
- Eine Heizung soll aufhören zu heizen.
- Bei einem Getränkeautomat soll ein Getränk im Ausgabefach liegen.
- Eine Braut soll heiraten.
- Peter Lustig soll eine E-Mail von seinem Provider abrufen können.

10/43

# In diesem Abschnitt

## Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

## Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

## Bedingungen und ausgelösten Aktionen in ZÜD

---

Zustandsmodelle

└ Einführung

└ 2

1

UML-Diagramme

---

# Zustandsdiagramme in der **UnifiedModelling Language**

UML (**U**nified **M**odelling **L**anguage) bei  
Zustandsdiagrammen

- ▶ Zustände werden durch abgerundete Rechtecke dargestellt.
- ▶ Ein **Zustandsübergang** wird durch ein (**Übergangs-**)Ereignis hervorgerufen und durch einen Pfeil dargestellt.
- ▶ Bei jedem Zustandsübergang kann eine **Aktion** ausgelöst werden.
- ▶ Jedes Zustandsdiagramm muss **genau einen Anfangszustand haben** (Symbol: ●→).
- ▶ Endzustände (Symbol: —○) darf es beliebig viele geben.

# Übung

- Ü 1.2: Buch, Aufgabe 2.6  
TA8(a) in yEd-Textfeld bearbeiten, yEd-Datei hochmoodeln
- Ü 1.3: Buch, Aufgabe 2.13  
Mit ausgelösten Aktionen ausführen. yEd-Datei hochmoodeln
- Ü 1.4: Buch, Aufgabe 2.14  
Texteditor eingemoodelt

13/43

# Übung

- Ü 1.5: Zahnbürste

Eine elektrische Zahnbürste lässt sich als Automat mit genau zwei Zuständen (an und aus) darstellen. Das Ereignis für einen Zustandsübergang ist das Drücken den An-/Ausknopfes, die ausgelöste Aktion das Laufen bzw. Stoppen des Motors.

- Zeichne das ZÜD.
- Ergänze das ZÜD um einen dritten Zustand `lädt`, im dem sich die Bürste beim Laden befindet. Trage auch die dazugehörigen Ereignisse und ausgelösten Aktionen ein.

14/43

# Übung

- Ü 1.6: verliebt-verlobt-verheiratet-getrennt

Bekanntlich können zwei Personen verliebt, verlobt, verheiratet oder getrennt sein. Erstelle ein dazu passendes ZÜD mit Übergangseignissen und ausgelösten Aktionen.

15/43

# Gliederung

## Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

## Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

Bedingungen und ausgelösten Aktionen in ZÜD

16/43



# In diesem Abschnitt

## Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

## Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

## Bedingungen und ausgelösten Aktionen in ZÜD

Zustandsmodelle

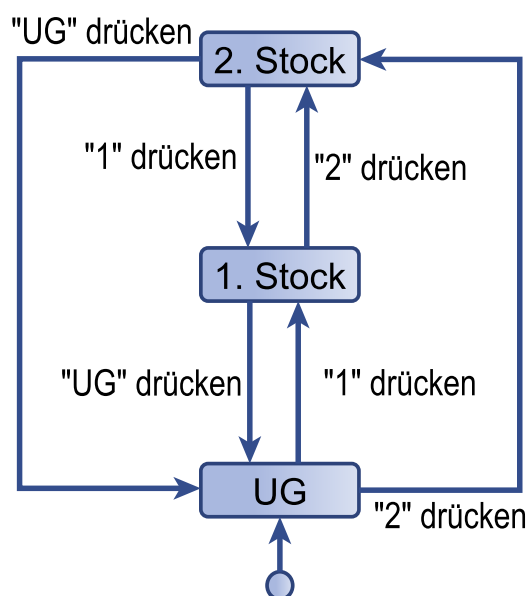
└ Implementation

└ 1

2

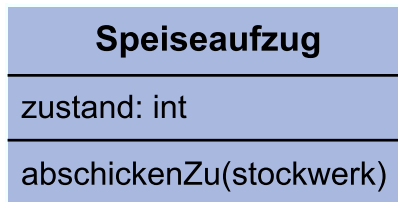
Beispiel

## Beispiel: Speiseaufzug



Die Zustände werden in der Variable `zustand` gespeichert, die Übergänge durch die Methode `abschickenZu` modelliert

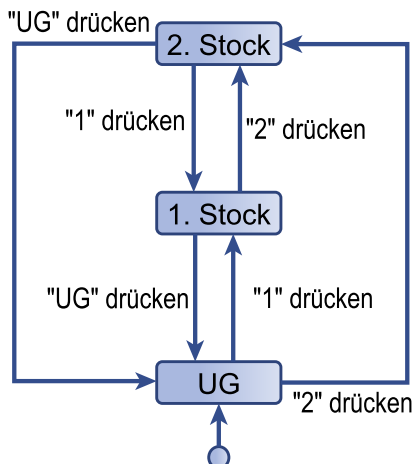
## Speiseaufzug Attributdeklaration



```
1 public class Speiseaufzug  
    () {  
2  
3     private int zustand;  
4  
5     //Konstruktor  
6     //Methoden  
7 }
```

19/43

## Speiseaufzug — Konstruktor



```
1 public Speiseaufzug() {  
2     zustand = 0;  
3 }
```

20/43

## Speiseaufzug — Methode abschicken(stockwerk)

```
1 public void abschickenZu(int stockwerk){
2
3     if ( zustand == 0 ){
4         if ( stockwerk==1 ){ zustand = 1; }
5         else {
6             if ( stockwerk==2 ){zustand = 2; }
7         }
8     }
9     if ( zustand == 1; ){
10        ...
11    }
12 }
```

21 / 43

## Zusammenfassung

### Implementation von Zustandsübergangsdigrammen

ZÜD werden wie folgt implementiert:

- ▶ ein Attribut `zustand` deklarieren,
- ▶ im Konstruktor dem Attribut `zustand` den Anfangszustand zuweisen,
- ▶ Methode(n) für die Zustandsübergänge implementieren. Dabei müssen in einer Fallunterscheidung für alle Zustände alle Übergänge implementiert werden.

22 / 43

# In diesem Abschnitt

## Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

## Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

## Bedingungen und ausgelösten Aktionen in ZÜD

---

Zustandsmodelle

└ Implementation

└ 2

2

Verzweigungen

---

## Verzweigungen

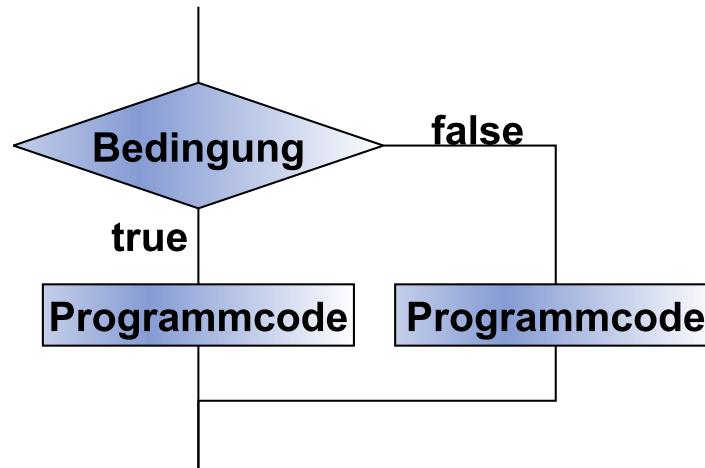
### Verzweigungen

werden mit der

**if** (Bedingung){...} **else** {...} -Kontrollstruktur umgesetzt. Der **if**-Teil wird ausgeführt, falls die Bedingung wahr ist, der **else**-Teil, falls die Bedingung falsch ist. **else**-Teil kann auch entfallen. Beispiel:

```
1 | if ( zustand == 0 ){  
2 |   System.out.println("Zustand ist Null");  
3 | }  
4 | else {  
5 |     System.out.println("Zustand ist nicht  
6 |       Null")  
7 | }
```

## Verzweigungen - Datenflussdiagramm



25 / 43

## Fallstrick

### Eine Wertzuweisung ist kein Vergleich

Wertzuweisung  $A=B$

Speicherzelle A wird mit dem Wert von Speicherzelle B belegt. Speicherzelle A wird also verändert.

logischer Vergleich  $A==B$

Es wird verglichen, ob die beiden Zelle A und B den gleichen Inhalt haben. Wenn „ja“ ist das Ergebnis von  $A==B$  gleich **true**, sonst **false**.

26 / 43

## Vergleichsoperatoren bei Zahlen und Einzelzeichen

Name	Term	true wenn
Gleichheit <sup>1</sup>	$A==B$	A und B den gleichen Wert haben
Ungleichheit <sup>a</sup>	$A!=B$	A und B verschiedene Werte haben
größer/kleiner	$A>B, A<B$	A größer/kleiner ist als B
größer/kleiner gleich	$A>=B, A<=B$	A größer/kleiner oder gleich B ist.

27 / 43

## Übungen

- Ü 2.1: Speiseaufzug

Material: Paket Speiseaufzug2Vorlage für TA c und d.

- Deklariere die Klasse Speiseaufzug (Attribute und Konstruktor) wie oben beschrieben.
- Ergänze die Methode positionGeben(), die dem Benutzer auf der Konsole angibt, in welchem Stockwerk sich der Aufzug befindet. Dabei soll folgender Text in der Konsole erscheinen:  
Der Aufzug befindet sich im UG bzw. ersten Stock bzw. zweiten Stock.
- Implementiere die Methode abschickenZu(int stockwerk) nach obiger Vorgabe und zeichne das Datenflussdiagramm mit yEd oder per Hand ins Heft.

27 / 43

# Übungen

- Ü 2.2: Kühlschrank

28 / 43

## In diesem Abschnitt

### Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —

UML-Diagramme

### Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

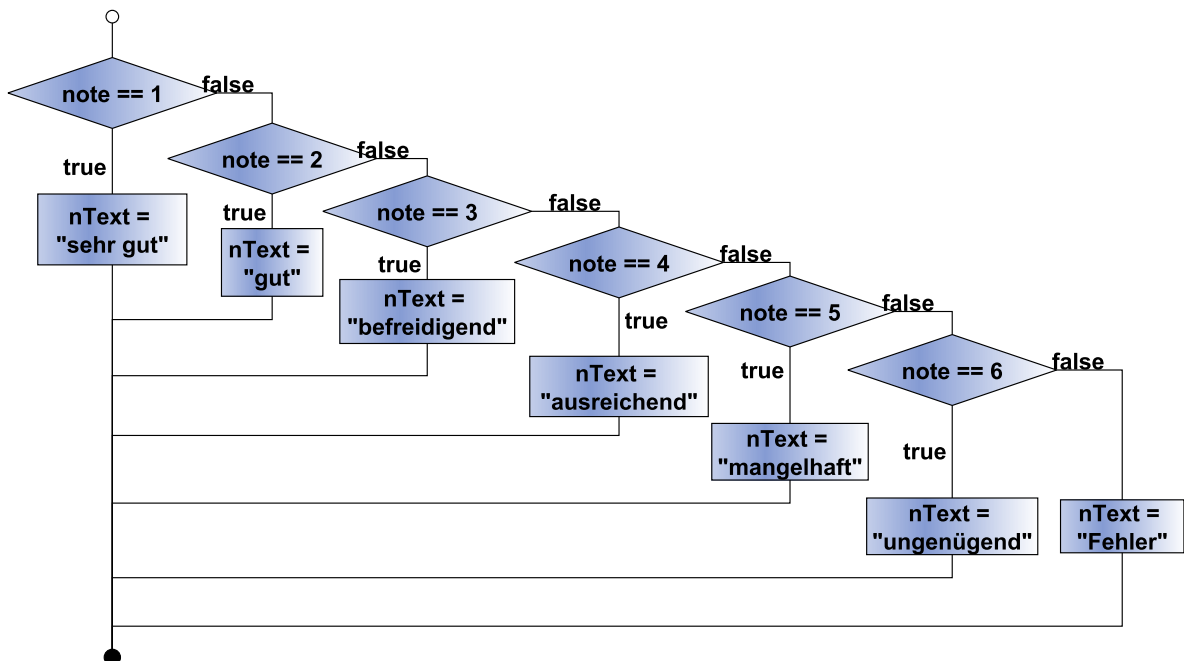
**switch-case**

### Bedingungen und ausgelösten Aktionen in ZÜD

## Unübersichtliche und verschachtelte if-Anweisungen

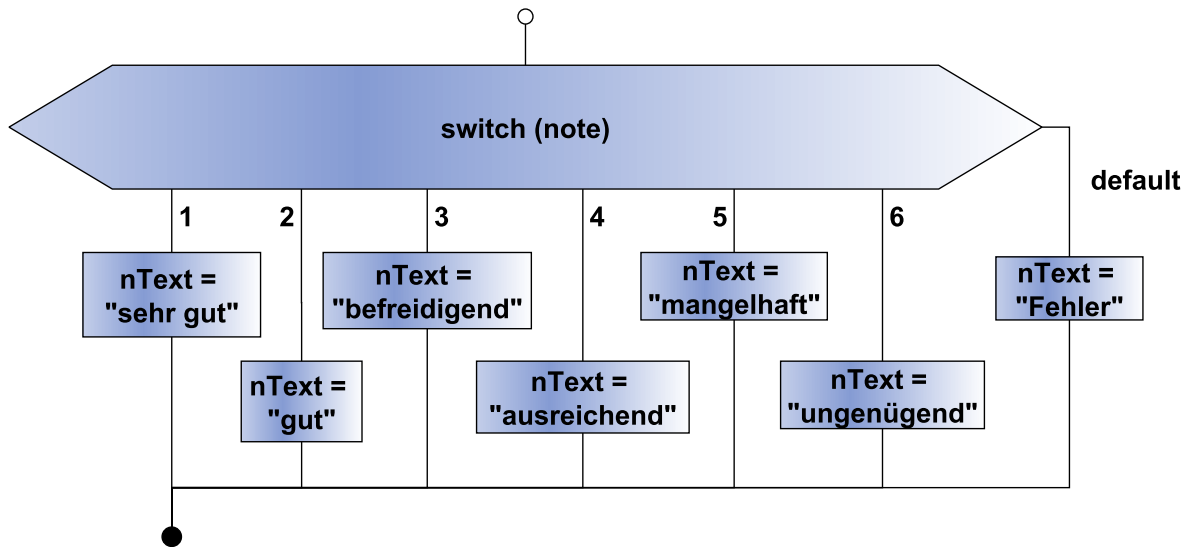
```
1 public void abschickenZu(int stockwerk){  
2  
3     if ( zustand == 0 ){  
4         if ( stockwerk==1 ){ zustand = 1; }  
5         else {  
6             if ( stockwerk==2 ){zustand = 2; }  
7         }  
8     }  
9     if ( zustand == 1; ){  
10        ...  
11    }  
12 }
```

## Unübersichtliche und verschachtelte if-Anweisungen





## Mehr Übersicht durch switch-Anweisung



32/43

## Implementation switch-Anweisungen

```
1  switch ( note ) {  
2      case 1: nText = "sehr gut";  
3          break;  
4      case 2: nText = "gut";  
5          break;  
6      ... //andere Fälle hier ausgelassen  
7      case 6: nText = "ungenügend";  
8          break;  
9      default: nText = "Fehler";  
10         break;  
11 }
```

33/43

## Bemerkungen zur switch-Anweisung

- ▶ Eine `switch`-Anweisung kann beliebig viele `case`-Klauseln enthalten.
- ▶ Die `break`-Anweisung nach einer `case`-Klausel sorgt dafür, dass nicht(!) die nächste `case`-Klausel bearbeitet, sondern ans Ende der `switch`-Anweisung gesprungen wird.
- ▶ Die `default`-Klausel ist optional und sammelt alle Fälle, die vorher nicht aufgeführt sind.

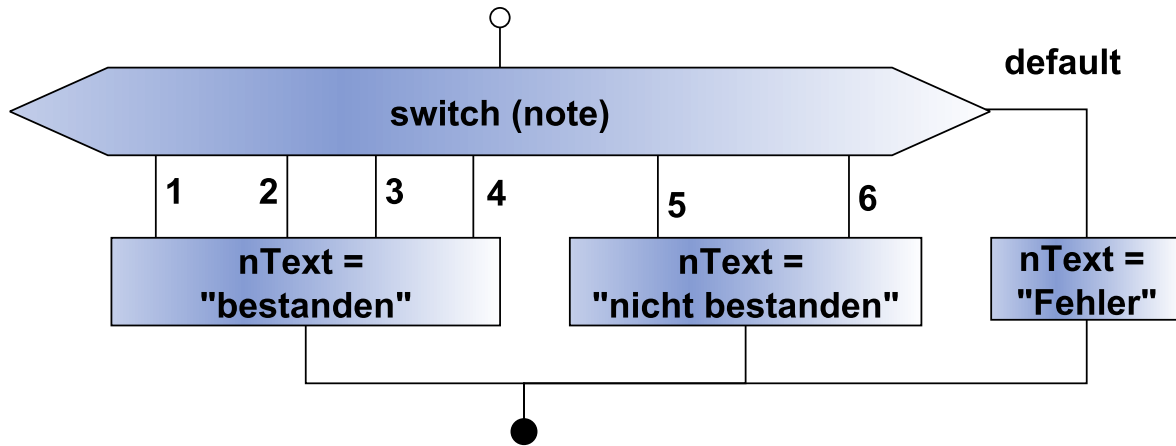
34/43

## Fälle zusammenfassen

```
1  switch ( note ) {  
2      case 1:  
3      case 2:  
4      case 3:  
5      case 4: nText = "bestanden";  
6          break;  
7      case 5:  
8      case 6: nText = "nicht bestanden";  
9          break;  
10     default: nText = "Fehler";  
11         break;  
12 }
```

35/43

## Fälle zusammenfassen



36 / 43

### • Ü 2.3: switch-Anweisungen

(a) Implementiere die beiden oben aufgeführten Beispiele, die Noten in Prädikate bzw. Noten in (nicht) bestanden umsetzen.

Erzeuge dazu in BlueJ eine Klasse `Switchen` und darin die beiden Methoden `public String noteToPraedikat(int note)` und `public String noteToBestanden(int note)`, in die du die `switch`-Anweisungen überträgst.

(b) Erstelle mit Hilfe von `switch`-Anweisungen die Methode `public String monatToMonatsname(int monat)`, die zur Nummer des Monats seine Bezeichnung ausgibt. `monatToMonatsname(9)` soll also September zurückgeben. Implementiere auch die Umkehrmethode `monatsnameToMonat`.

(c) Erstelle mit Hilfe von `switch`-Anweisungen die Methode

37 / 43

## Gliederung

### Einführung

Inselspiel

Zustandsdiagramme in der Unified-Modelling-Language —  
UML-Diagramme

### Implementation von Zuständen und Verzweigungen

Beispiel Speiseaufzug

Verzweigungen

switch-case

## Bedingungen und ausgelösten Aktionen in ZÜD

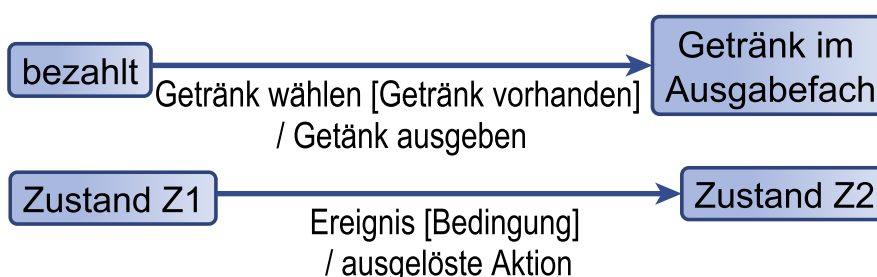
38 / 43

## ZÜD mit Bedingungen und ausgelösten Aktionen

### ZÜD mit Bedingungen und ausgelösten Aktionen

In ZÜD können Ereignisse (z.B. Getränk wählen) ergänzt werden durch **Bedingungen** (z.B. Getränk vorhanden). Der Übergang findet nur statt, wenn die Bedingung den Wert **true** liefert.

Mit dem dem Übergang kann eine **Aktion** ausgelöst werden (z.B. Getränk ausgeben)



39 / 43

## Übung

- Ü 3.1: Getränkeautomat

Ergänze das ZÜD des oben angegebenen Getränkeautomaten um den Zustand `bereit` und vervollständige das Diagramm durch Übergänge mit Bedingungen und ausgelösten Aktionen.

Folgende Ereignisse, Bedingungen bzw. Aktionen müssen enthalten sein:  
`genug bezahlt`, `Becher nehmen`, `Anzeige aktualisieren`.

40 / 43

## Übung

- Ü 3.2: Tresor

(a) Aufgabe 2.39

(b) Implementiere die Klasse `Tresor` nach dem im Buch auf S. 55 abgebildeten Zustandsdiagramm.

Hinweise

- ▶ Du musst zumindest die Attribute `zustand`, `anzahlFehlversuche` und `pin` deklarieren
- ▶ Dem Konstruktor soll die notwendige PIN als Parameter übergeben werden.
- ▶ Implementiere die Methoden `oeffnen` und `verriegeln`.

41 / 43

# Übung

- Ü 3.3: Soziales Netzwerk  
run: ./Arbeitsmaterial/SozialesNetzwerk/SozialesNetzwerk.pdf
- Ü 3.4: 2.51
- Ü 3.5: 2.52
- Ü 3.6: Wie Geht's