

5. Auswahl: Bedingte Anweisungen

Schwerpunkte

- If-Anweisung: Auswahl aus 2 Alternativen
- Kurzform von if: Spezialfall der Langform
- Fehlerquellen
- Hängendes else
- switch: Spezialfall von if

Syntax

If-Anweisung: Auswahl aus zwei Alternativen

EBNF:

```
if ( Ausdruck ) Anweisung else Anweisung
```

→ je Alternative: e i n e Anweisung !

Beispiele:

```
if (x == 0)
    System.out.print(0);
else
    System.out.print(y/x);
```

```
if (x >= y)
    max = x;
else
    max = y;
```

Unterschiede zu Pascal?

Pascal

```
if (x >= y) then ...
else ...
```

If-Anweisung: mehrere Anweisungen

EBNF:

```
if ( Ausdruck ) Anweisung else Anweisung
```

mehrere Einzelanweisungen?

→ mit { ... } zu einer machen

Beispiel:

```
if ( x == 0 ) { //Ausnahme: Div durch 0
    System.out.print(0);
    x = y;
}
else {
    System.out.print(y/x);
    y = x;
}
```

If-Anweisung: Kurzform

EBNF: (noch Langform)

```
if ( Ausdruck ) Anweisung else Anweisung
```

```
if ( a + b <= c )
    System.out.print("kein Dreieck");
else ;
```

z. B. Leeranweisung

EBNF: (Kurzform = Spezialfall der Langform)

```
if ( Ausdruck ) Anweisung
```

```
if ( a + b <= c )
    System.out.print("kein Dreieck");
```

Typische Fehler

```
if a > b
```

fehlende Klammern

```
...
else
...
```

```
if ( a = 1 )
```

Falle für
Pascal-Programmierer !

```
...
else
...
```

Wert von a?

```
if ( x > y )
    x = y; y = 0;
if ( y == 0 )
    ...
```

Welchen Wert
hat y hier?

```
if ( x > y )
    x = y;
y = 0;
if ( y == 0 )
    ...
```

Hängendes else (dangling else)

Hängendes else: Beispiel

Eingangswerte:

x	y	z	ok
2	1	100	-100

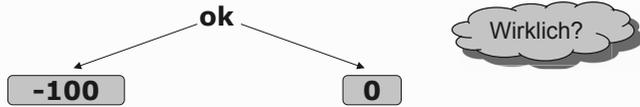
```

if (x > y)
    if (y > z)
        ok = 1;
    else
        ok = 0;
    
```

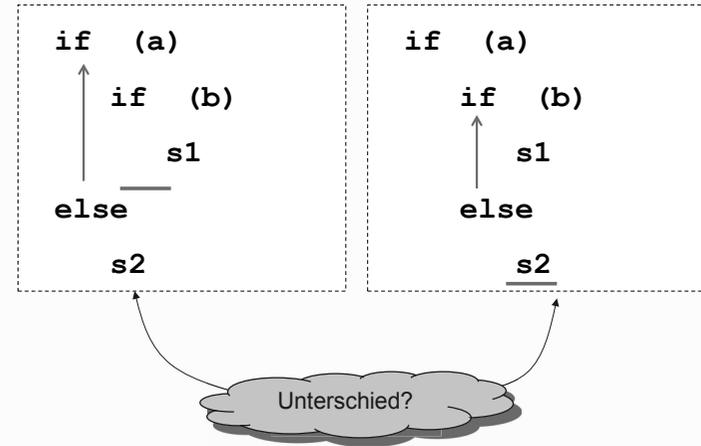
```

if (x > y)
    if (y > z)
        ok = 1;
    else
        ok = 0;
    
```

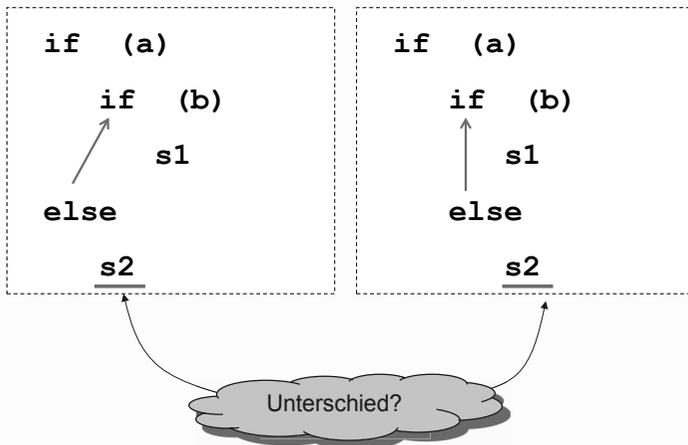
Resultat :



Hängendes else: das Problem



Hängendes else: das Problem



Beide Varianten: identisch

Nur anders eingerückt (Layout)

→ Absicht des Programmierers:

else s2; gehört zu ...

- 1. Form: äußerem if
- 2. Form: innerem if

Bedeutung eines Programms: unabhängig vom Layout

```
if (a) if (b) s1 else s2
```

→ Festlegung: 2. Form ist verbindlich !

Hängendes-else-Problem: Ursache

Grammatik von Java nicht eindeutig !

Was heißt das?

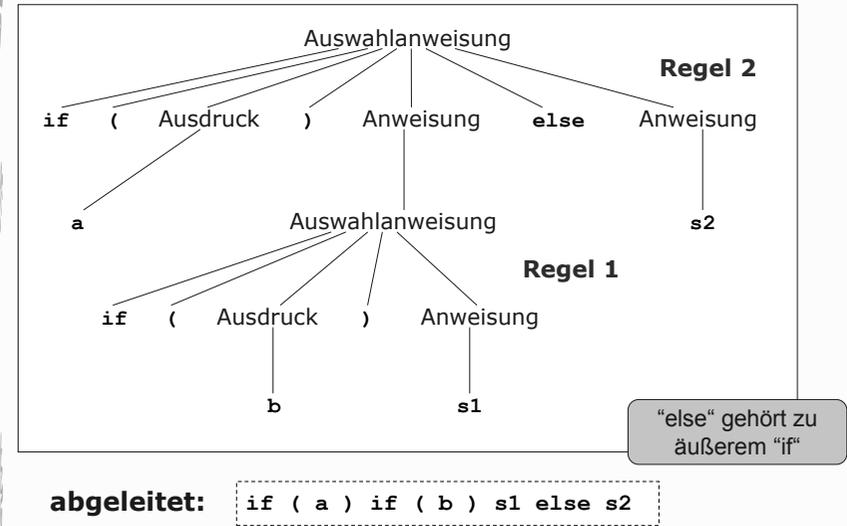
Auswahlanweisung ::=
`if (Ausdruck) Anweisung [else Anweisung] |`
`switch ...`

3 Regeln

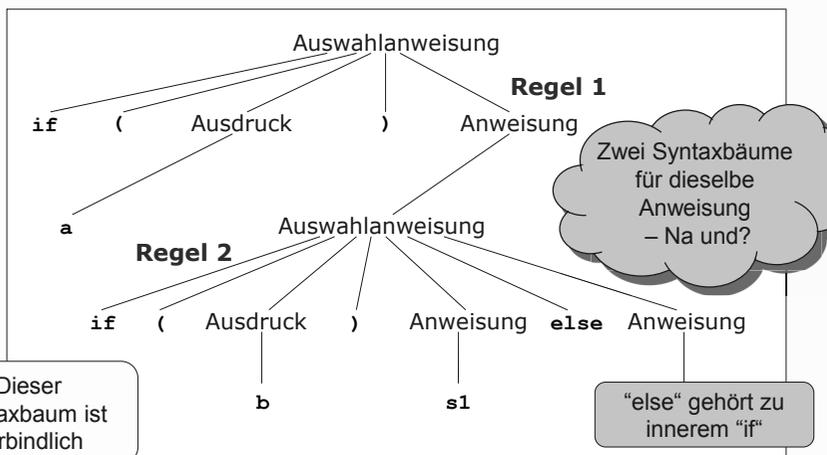
Auswahlanweisung ::= **Regel 1**
`if (Ausdruck) Anweisung |`
`if (Ausdruck) Anweisung else Anweisung |` **Regel 2**
`switch ...`

`if (a) if (b) s1 else s2`

Syntaxbaum: Variante 1



Syntaxbaum: Variante 2

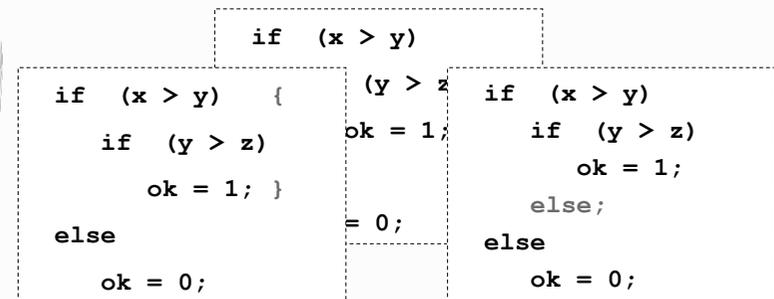


Dieser Syntaxbaum ist verbindlich

Zwei Syntaxbäume für dieselbe Anweisung – Na und?

Probleme

- Was tun: Wenn man doch die 1. Variante will?



- Wie könnte die Java-Syntax eindeutig gemacht werden?

Jede if-Anweisung wird abgeschlossen mit einem ... `endif`

Eindeutigkeit durch „endif“

```

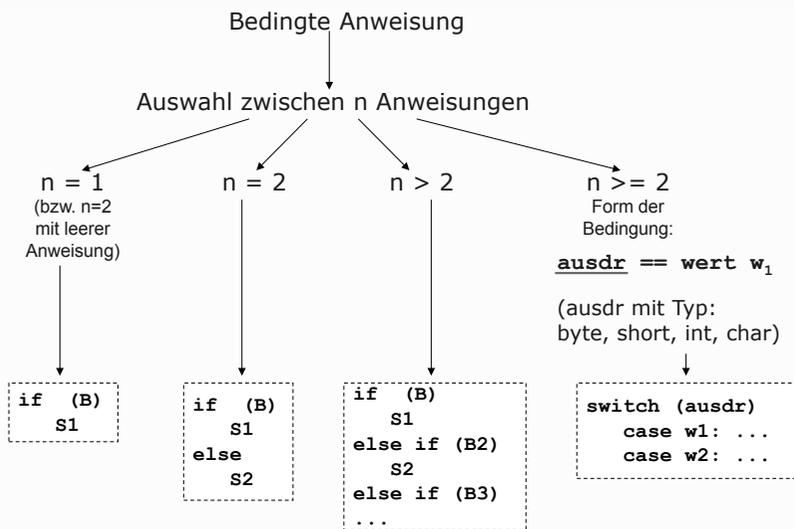
if (x > y)
  if (y > z)
    ok = 1;
  endif
else
  ok = 0;
endif
    
```

```

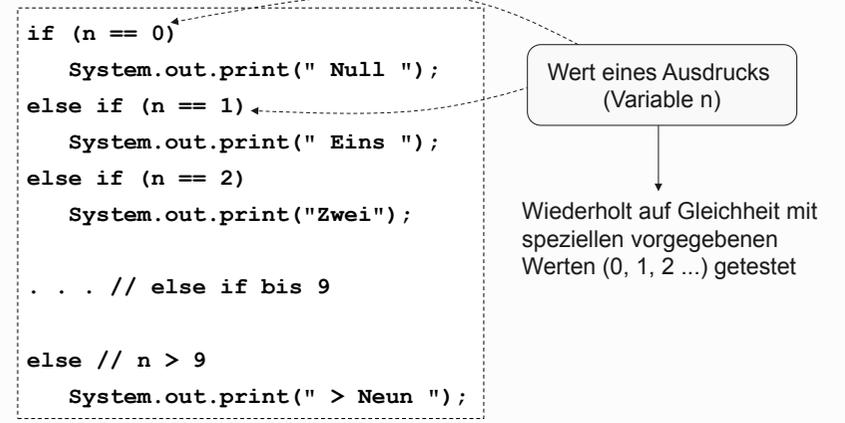
if (x > y)
  if (y > z)
    ok = 1;
  else
    ok = 0;
  endif
endif
    
```

Switch

Auswahl einer bedingten Anweisung



Switch-Anweisung: Motivation



Switch-Anweisung: Auswahl aus mehreren Varianten

```
switch (n) {
    case 0: System.out.print( "Null" );
            break;
    case 1: System.out.print( "Eins" );
            break;
    case 2: System.out.print( "Zwei" );
            break;
    ....
    default: System.out.println( " > Neun " );
}
```

→ dieselbe Semantik wie eben

switch: schalten, umlenken

Switch: sinnvolles Beispiel

```
→ final int jan = 1, feb = 2, mar = 3, ... dez = 12;
int monat, jahr, anzTage;
... // Einlesen von monat, jahr
switch (monat) {
    case feb: ...modulo... ...and...
                if (((jahr % 4) == 0) && ((jahr % 100) != 0)
                    || ((jahr % 400) == 0)) //Schaltjahrtest
                ...or...
                    anzTage = 29;
                else anzTage = 28;
                break;
    case apr: case jun: case sep: case nov:
                anzTage = 30; break;
    default: anzTage = 31;
}
```

Konstante

Initialisierung

≠

...or...

mehrere Fälle gleichzeitig

Standard

Tage eines Monats

Switch: sinnvolles Beispiel

Konstante

```
→ final int jan = 1, feb = 2, mar = 3, ... dez = 12;
int monat, jahr, anzTage;
... // Einlesen von monat, jahr
switch (monat) {
    case feb: ...modulo... ...and...
                if (((jahr % 4) == 0) && ((jahr % 100) != 0)
                    || ((jahr % 400) == 0)) //Schaltjahrtest
                ...or...
                    anzTage = 29;
                else anzTage = 28;
                break;
    case apr: case jun: case sep: case nov:
                anzTage = 30; break;
    default: anzTage = 31;
}
```

Initialisierung

≠

...or...

**Jahr ist Schaltjahr (29. Februar),
wenn es durch 4 teilbar,
aber nicht auch durch 100 teilbar ist,
mit der Ausnahme,
dass ein durch 400 teilbares Jahr
ein Schaltjahr ist (z. B. das Jahr 2000).**