



Vorlesung an der Berufsakademie Oldenburg

Unterrichtseinheit 7

Kontrollstrukturen (Hamster-Modell)

Dr. Dietrich Boles

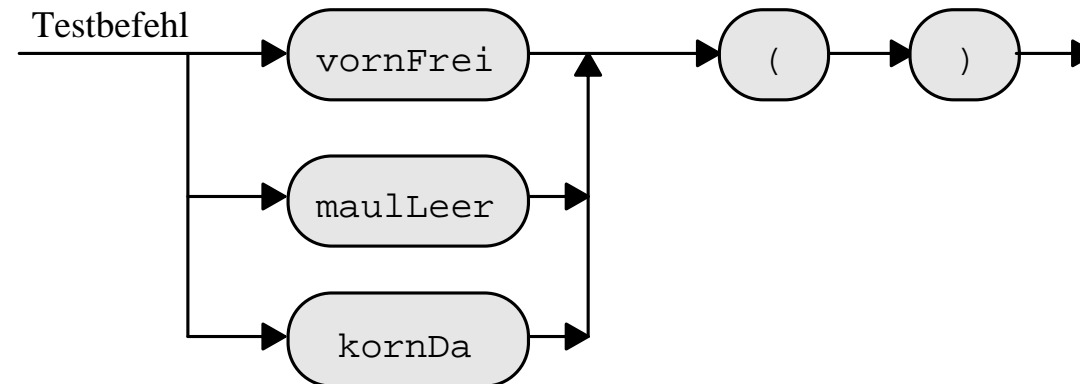
- Kontrollstrukturen
- Testbefehle
- Boolesche Ausdrücke
- Blockanweisung
- Leeraanweisung
- Bedingte Anweisung
- Alternativanweisung
- Wiederholungsanweisung
- Endlosschleife
- Beispiele
- Codekonventionen
- Aufgaben

Motivation

- Nur wenn der Hamster Körner im Maul hat, soll er eines davon ablegen
- Nur wenn die Kachel vor dem Hamster frei ist, soll er nach vorne gehen; andernfalls soll er sich linksum drehen.
- Solange der Hamster nicht vor einer Mauer steht, soll er wiederholt eine Kachel nach vorne gehen.
- Solange sich noch Körner auf der Kachel des Hamsters befinden, soll er wiederholt ein Korn aufnehmen.
- Solange der Hamster nicht vor einer Mauer steht und er noch Körner im Maul hat, soll er wiederholt eine Kachel nach vorne gehen und jeweils ein Korn ablegen.
- Hamster-Programme für eine bestimmte Klasse von Territorien!

Drei Testbefehle:

Syntax:



Semantik:

Lieferung von booleschen Werten (**true**, **false**):

vornFrei()

ist das Feld vor dem Hamster blockiert?

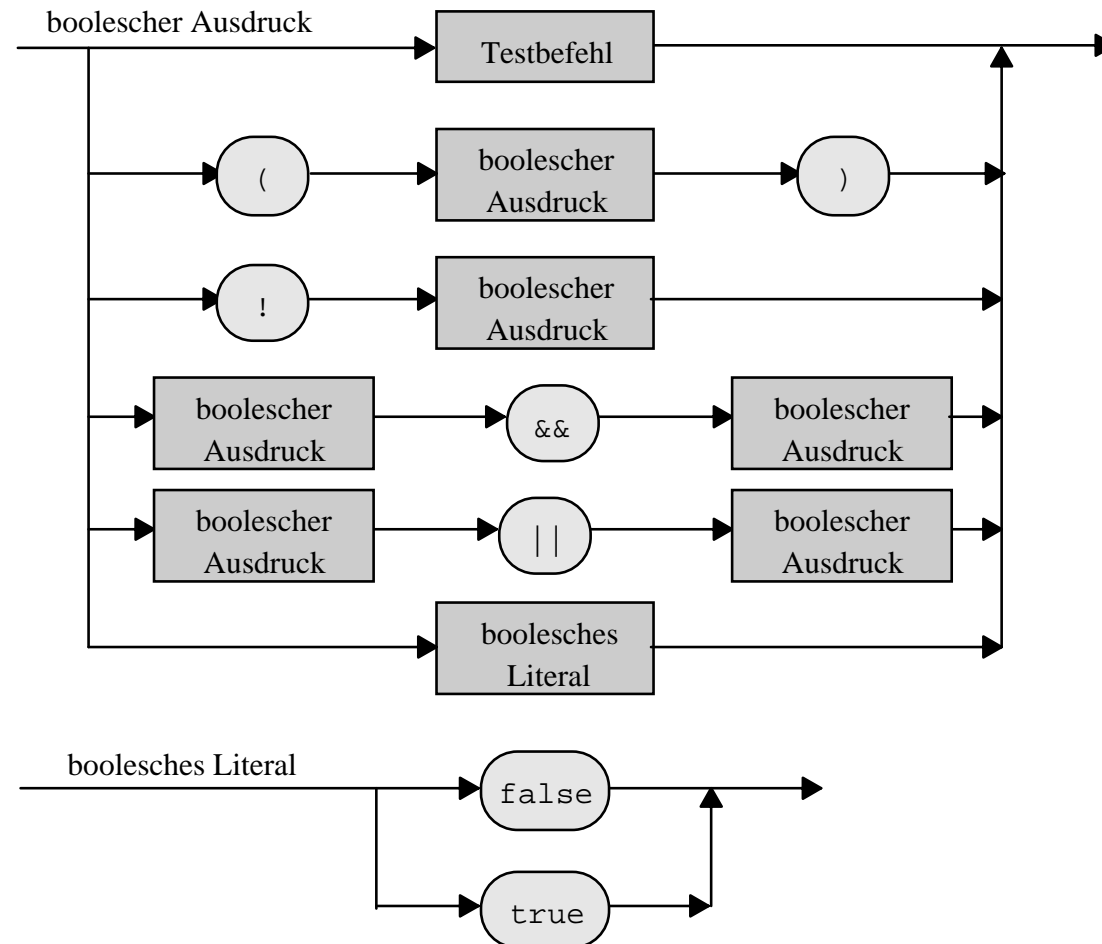
maulLeer()

ist das Maul des Hamsters leer?

kornDa()

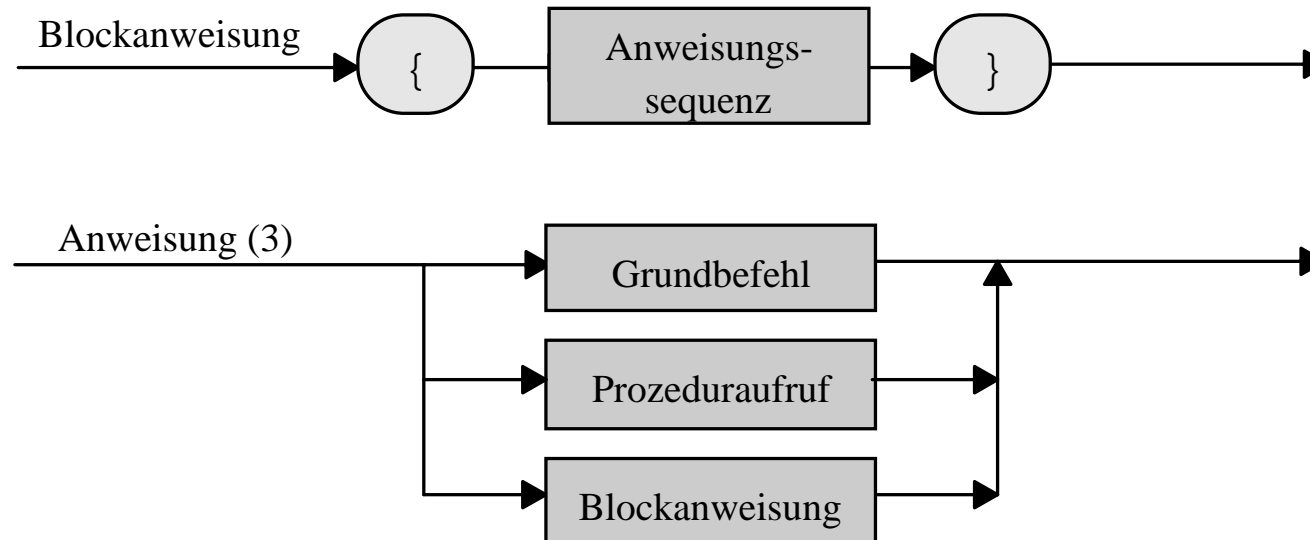
liegt ein Korn auf der aktuellen Kachel?

Syntax:



Semantik: siehe Aussagenlogik

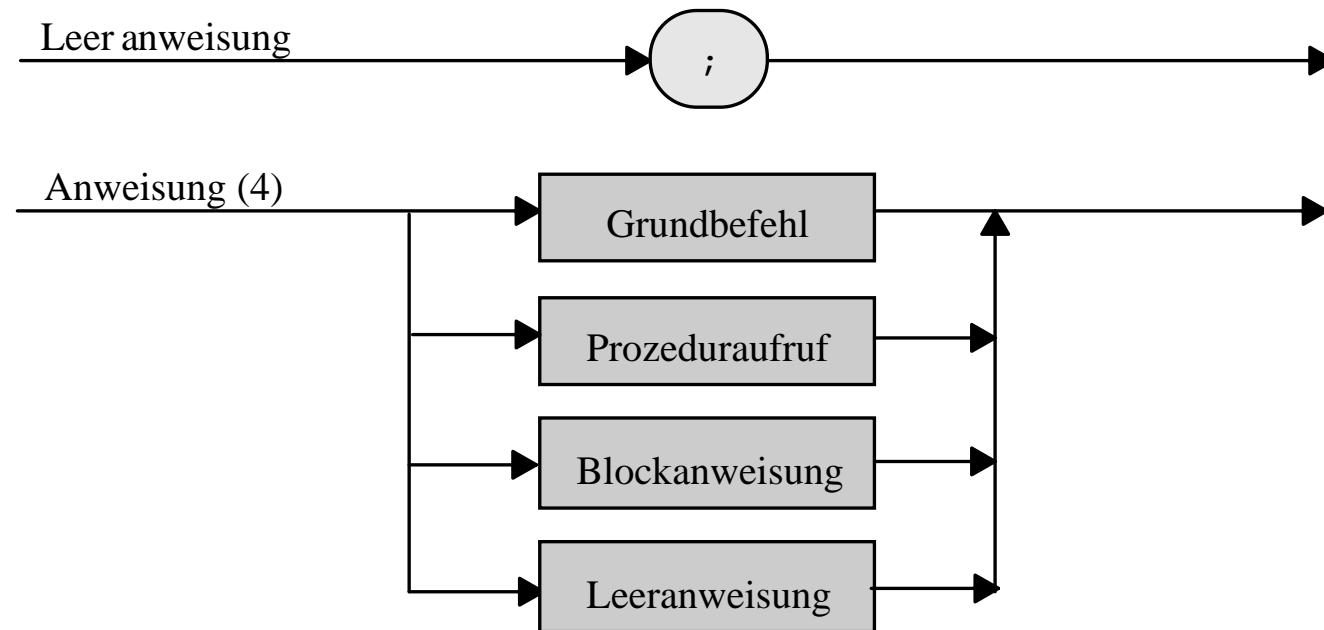
Syntax:



Semantik:

Zusammenfassung mehrerer Anweisungen zu einer
(zusammengesetzten) Anweisung

Syntax:

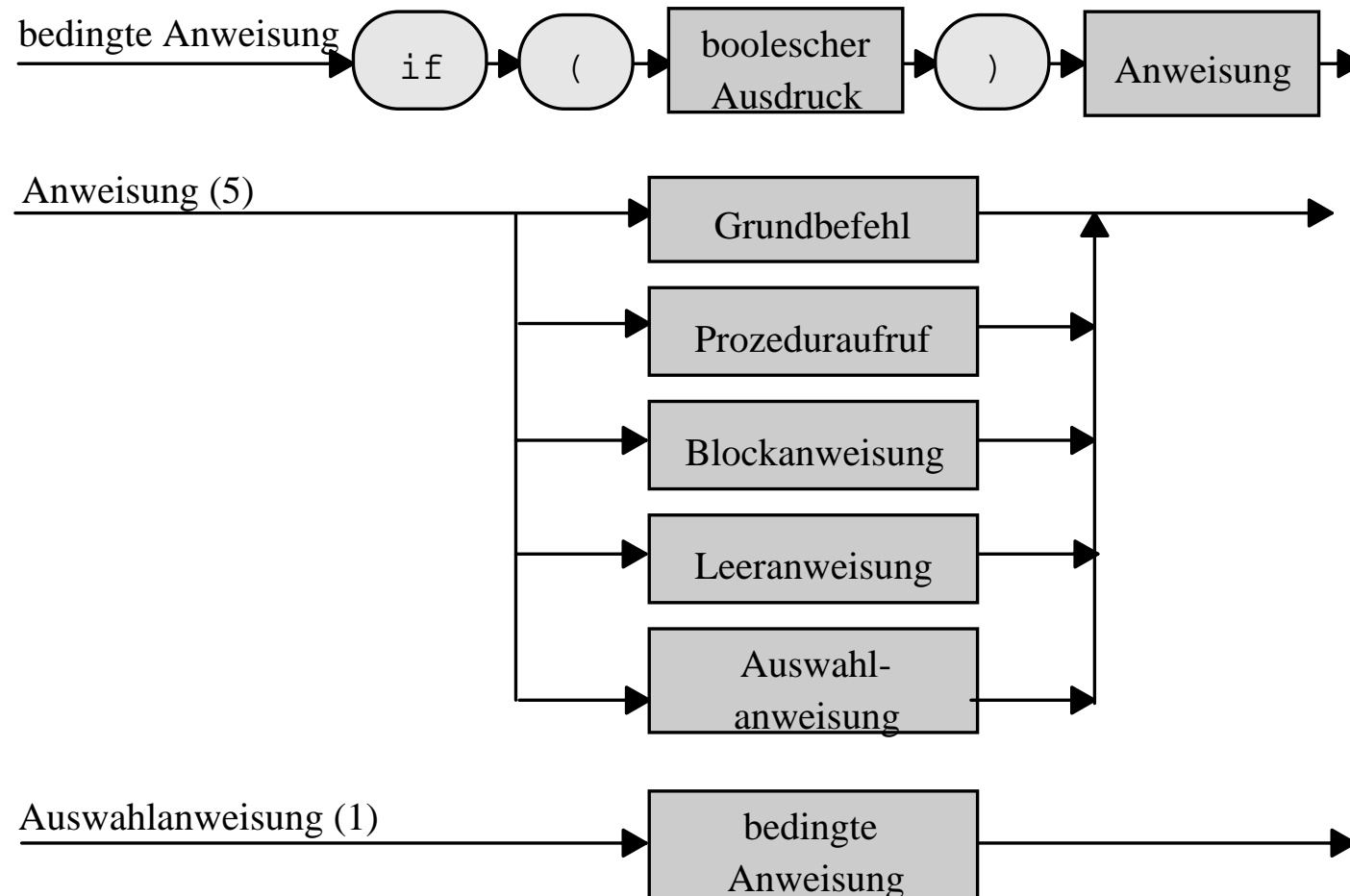


Semantik:

keine Auswirkungen auf den Programmablauf

Bedingte Anweisung (1)

Syntax:



Semantik:

1. Werte den booleschen Ausdruck aus.
2. Nur falls der Ausdruck den Wert **true** liefert, führe die true-Anweisung aus.

Beispiele:

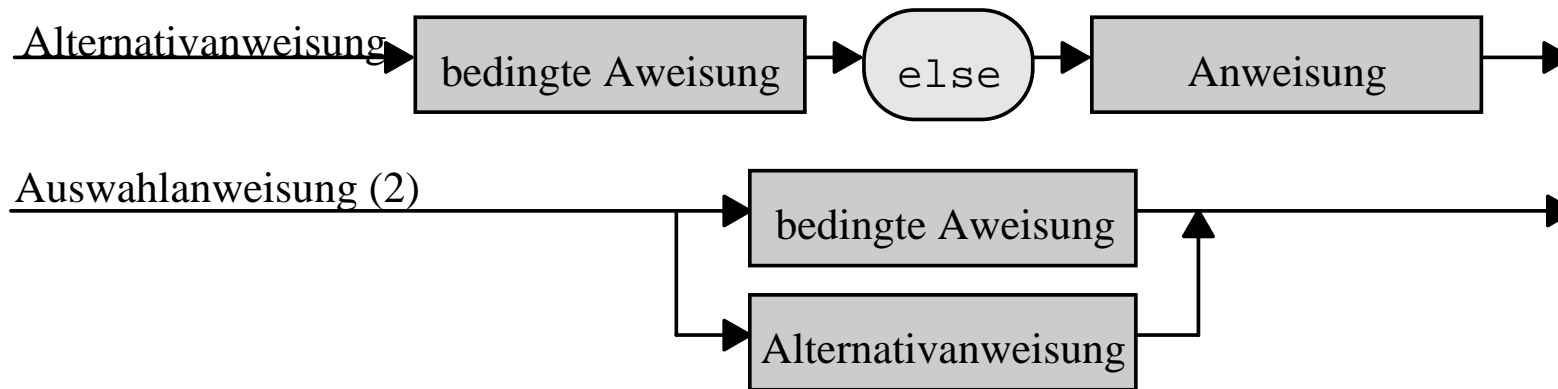
```
if (vornFrei())  
    vor();
```

```
if (!maulLeer()) {  
    gib();  
}
```

```
if (kornDa() && vornFrei()) {  
    nimm();  
    vor();  
}
```

```
if (kornDa())  
    if (vornFrei()) {  
        nimm();  
        vor();  
    }
```

Syntax:



Semantik:

1. Werte den booleschen Ausdruck aus.
2. Falls der Ausdruck den Wert **true** liefert, führe die true-Anweisung aus.
3. Andernfalls führe die false-Anweisung aus.

Beispiele:

```
if (vornFrei()) {  
    vor();  
} else {  
    linksUm();  
}
```

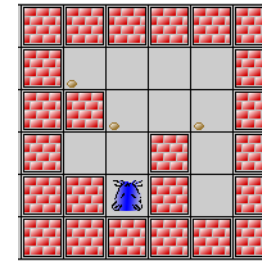
```
if (maulLeer())  
    ;  
else  
    gib();  
linksUm();
```

```
if (vornFrei())  
    vor();  
else if (kornDa())  
    nimm();  
else if (!maulLeer())  
    gib();  
else  
    linksUm();
```

Motivation: Der Hamster soll bis zur nächsten Wand laufen.



```
void main() {  
    vor();  
    vor();  
}
```

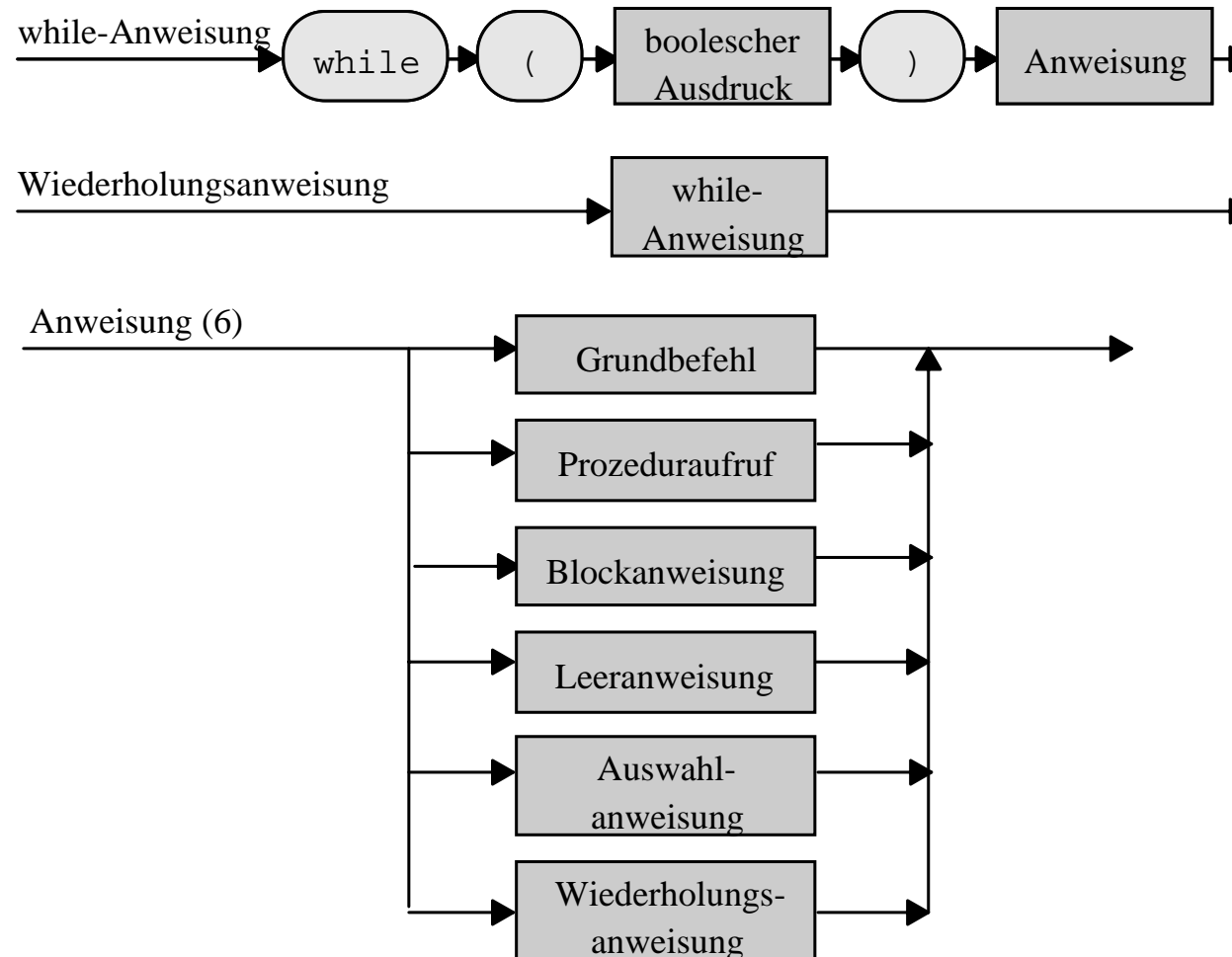


```
void main() {  
    vor();  
    vor();  
    vor();  
}
```

Allgemeingültig: `solange vornFrei(): vor();`

Wiederholungsanweisung (1)

Syntax:

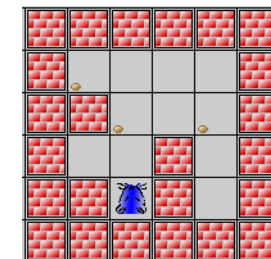
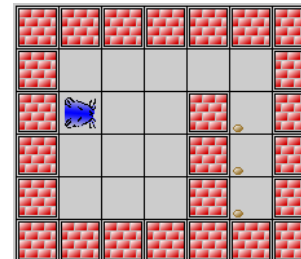


Semantik:

1. Werte den booleschen Ausdruck (*Schleifenbedingung*) aus.
2. Falls die Schleifenbedingung den Wert **false** liefert, beende die Wiederholungsanweisung.
3. Falls die Schleifenbedingung den Wert **true** liefert, führe die Anweisung (Iterationsanweisung) aus und fahre bei (1.) fort.

Beispiel:

```
void main() {  
    while (vornFrei()) {  
        vor();  
    }  
}
```



```
void main() {  
    while (vornFrei()) {  
        linksUm();  
    }  
}
```



Aufgabe: Landschaft beliebig! Der Hamster soll bis zur nächsten Wand laufen und dabei alle Körner fressen.

```
void laufeBisZurNaechstenWandUndSamme() {  
    samme();  
    while (vornFrei()) {  
        vor();  
        samme();  
    }  
}  
  
void samme() {  
    while (kornDa()) {  
        nimm();  
    }  
}
```

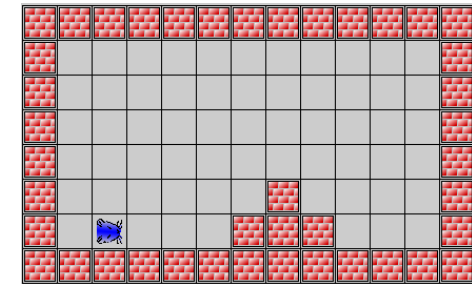
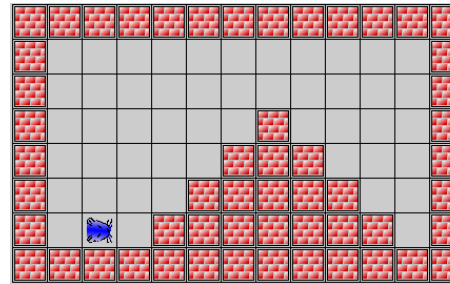

Aufgabe:

Der Hamster steht vor einem regelmäßigen Berg unbekannter Höhe. Er soll den Gipfel erklimmen.

Programm:

```
void main() {
    laufeZumBerg();
    erklimmeGipfel();
}
void laufeZumBerg() {
    while (vornFrei())
        vor();
}
void erklimmeGipfel() {
    while (!vornFrei()) {
        erklimmeEineStufe();
    }
}
```

Typische Landschaften:



```
void erklimmeEineStufe() {
    linksUm();
    vor();
    rechtsUm();
    vor();
}

void rechtsUm() {
    linksUm();
    linksUm();
    linksUm();
}
```

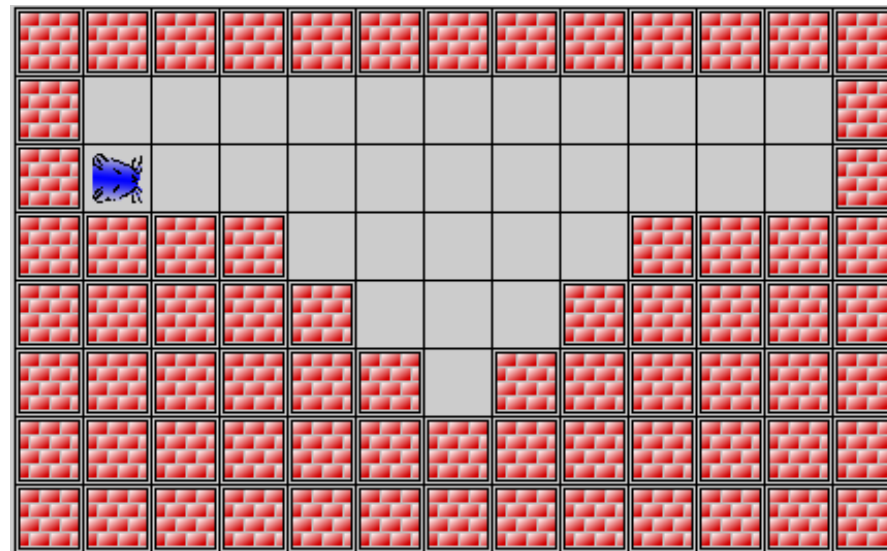
- Hinter `if` und `while` ein Leerzeichen
- Vor und hinter duadischen Operatoren ein Leerzeichen
- Für `true`-, `false`- und Iterationsanweisung möglichst immer die Blockanweisung verwenden
- falls Blockanweisung: `}` unter `i` von `if` bzw. `w` von `while`
- innere Anweisungen um 4 Spalten einrücken
- Alternativanweisung: `}` `else` `{` (in eine Zeile)
- geschachtelte Schleifen vermeiden (→ Prozeduren)

Aufgabe 1

Aufgabe:

Der Hamster steht links vor einer unbekannt tiefen regelmäßigen Mulde. Er soll zur tiefsten Stelle hinabsteigen und dort stehen bleiben.

Beispiellandschaft:



Aufgabe 2

Aufgabe:

Gegeben ein beliebiges Territorium ohne innere Mauern. Der Hamster soll alle Körner fressen.

Beispiellandschaft:

