



Vorlesung an der Berufsakademie Oldenburg

Unterrichtseinheit 4

Anweisungen und Programme (Hamster-Modell)

Dr. Dietrich Boles

- Lexikalik
- Token
- Schlüsselwörter
- Bezeichner
- Hamster-Befehle
- Anweisungen
- Programme
- Kommentare
- Beispiele
- Codekonventionen
- Aufgaben

- Java zugrundeliegender Zeichensatz: **Unicode**
- 16-Bit-Zeichensatz (2^{16} Zeichen)
- erste 128 Zeichen: **ASCII** (7-Bit-Zeichensatz)

**Möglichst nur ASCII-Zeichen bzw.
Zeichen auf der Tastatur verwenden !!!!!!!!!!!!!!!**

- Token: lexikalische Einheiten

- Symbole: `<, =, <=, ...`
- Schlüsselwörter: `while, if, ...`
- Bezeichner: *Prozedurnamen, Klassennamen, ...*
- Literale: `true, 23, 24.5f, "hello world", ...`

- Trennung von Token:

- Leerzeichen (Blank)
- Tabulator
- Zeilenende
- Zeilenvorschub
- Seitenvorschub

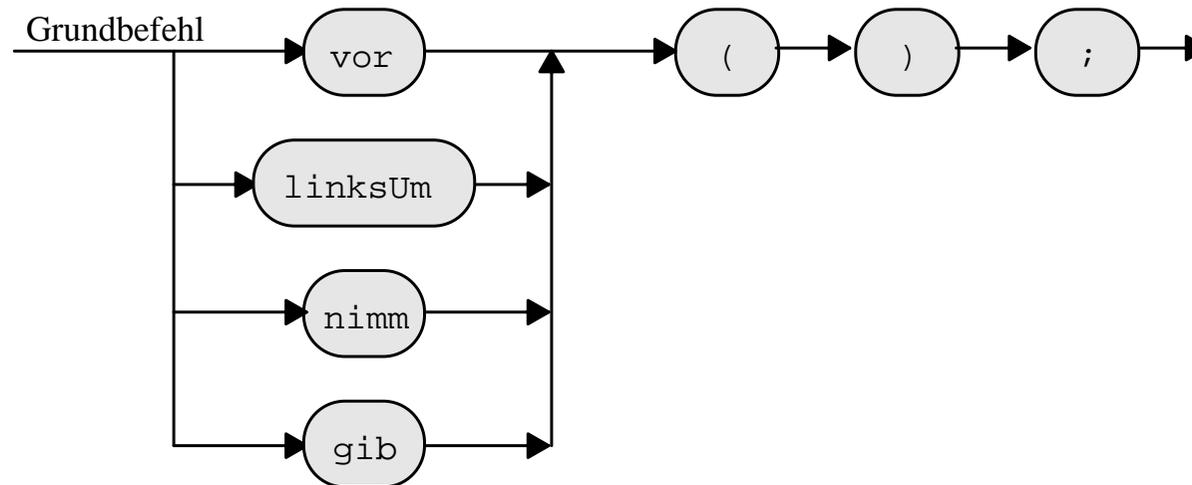
<code>abstract</code>	<code>assert</code>	<code>boolean</code>	<code>break</code>
<code>byte</code>	<code>case</code>	<code>catch</code>	<code>char</code>
<code>class</code>	<code>const</code>	<code>continue</code>	<code>default</code>
<code>do</code>	<code>double</code>	<code>else</code>	<code>enum</code>
<code>extends</code>	<code>false</code>	<code>final</code>	<code>finally</code>
<code>float</code>	<code>for</code>	<code>goto</code>	<code>if</code>
<code>implements</code>	<code>import</code>	<code>instanceof</code>	<code>int</code>
<code>interface</code>	<code>long</code>	<code>native</code>	<code>new</code>
<code>null</code>	<code>package</code>	<code>private</code>	<code>protected</code>
<code>public</code>	<code>return</code>	<code>short</code>	<code>static</code>
<code>strictfp</code>	<code>super</code>	<code>switch</code>	<code>synchronized</code>
<code>this</code>	<code>throw</code>	<code>throws</code>	<code>transient</code>
<code>true</code>	<code>try</code>	<code>void</code>	<code>volatile</code>
<code>while</code>			

- Benennung von deklarierten Einheiten und Labeln:
 - Klassennamen
 - Variablennamen
 - Prozedurnamen
 - ...

- Beginn mit **Buchstabe, Unterstrich (_)** oder **\$-Zeichen**
- anschließend: **Buchstaben, Ziffern, Unterstriche, \$-Zeichen**
- Möglichst keine Umlaute und kein ß verwenden!
- Beispiele:
 - **Katze**
 - **Kaetzchen**
 - **_zahl**
 - **\$more_money\$**

Vier Grundbefehle:

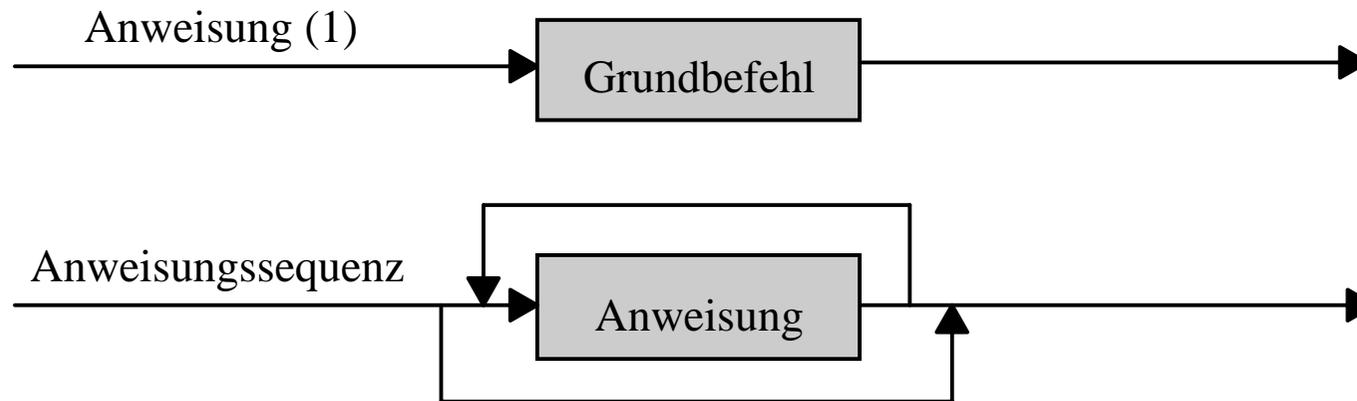
Syntax:



Semantik:

- | | |
|-------------------------|---|
| <code>vor();</code> | ein Feld nach vorne springen |
| <code>linksUm();</code> | 90 Grad nach links drehen |
| <code>nimm();</code> | ein Korn von der aktuellen Kachel aufnehmen |
| <code>gib();</code> | ein Korn aus dem Maul auf der akt. Kachel ablegen |

Syntax:

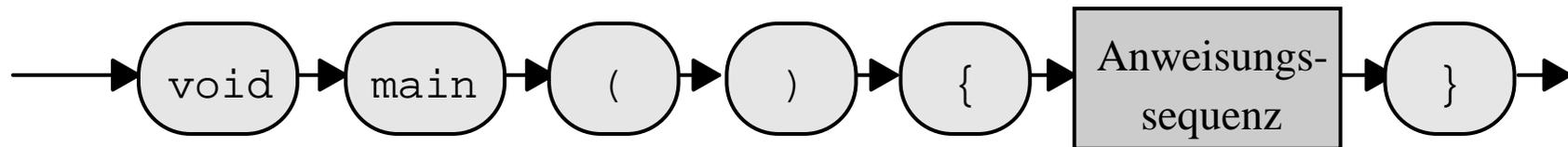


Semantik:

Die Anweisungen der Anweisungssequenz werden nacheinander ausgeführt.

Syntax:

Programm (1)



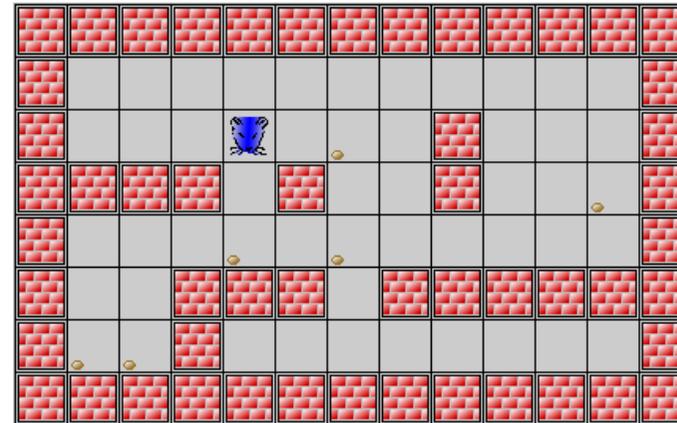
Semantik:

Beim Aufruf des Programms wird die Anweisungssequenz ausgeführt.

Aufgabe:

Gegeben sei das folgende
Hamster-Territorium.
Der Hamster soll zwei Körner
einsammeln.

Landschaft:



Programm:

```
void main() {  
    // friss erstes Korn  
    vor(); vor(); nimm();  
  
    /*  
     * friss zweites Korn  
     */  
    linksUm(); vor(); vor(); nimm();  
}
```

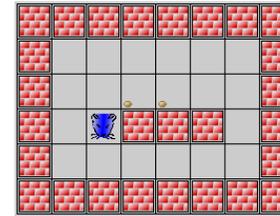
Aufgabe:

Gegeben sei das folgende Territorium.
Der Hamster habe vier Körner im Maul.
Er soll in jeder Ecke eins ablegen und
in seine Ausgangssituation zurückkehren.

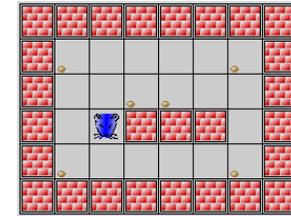
Programm:

```
void main() {  
    // begib dich an den Rand  
    vor(); linksUm();  
    // laufe in die rechte untere Ecke  
    vor(); vor(); vor(); vor(); gib(); linksUm();  
    // laufe in die rechte obere Ecke  
    vor(); vor(); vor(); gib(); linksUm();  
    // laufe in die linke obere Ecke  
    vor(); vor(); vor(); vor(); vor(); gib(); linksUm();  
    // laufe in die linke untere Ecke  
    vor(); vor(); vor(); gib(); linksUm();  
    // begib dich in die Ausgangsposition zurück  
    vor(); linksUm(); vor(); linksUm(); linksUm();  
}
```

Landschaft:



vorher



nachher

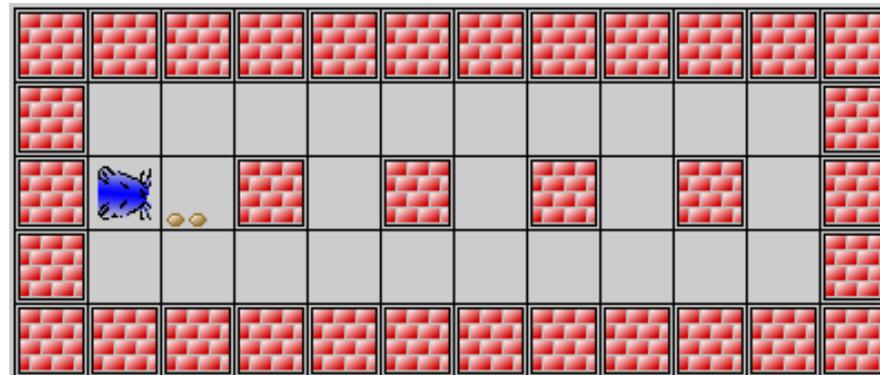
- auf Lesbarkeit des Codes achten
- `void main() {` in eine Zeile
- `}` unterhalb des `v` von `void`
- innere Anweisungen um 4 Spalten einrücken
- pro Zeile **eine** Anweisung
- Leerzeile vor Kommentaren
- Bereichskommentare folgendermaßen strukturieren:

```
/*  
 * Kommentar  
 */
```

Aufgabe 1

Aufgabe:

Gegeben sei das folgende Hamster-Territorium mit einem Slalom-Parcours. Der Hamster soll die 2 Körner (Skistöcke ☺) aufnehmen, den Slalom-Parcours bewältigen und am Ziel die Körner wieder ablegen.



Aufgabe 2

Aufgabe:

Der Hamster steht irgendwo mit Blickrichtung Osten in einem mauerlosen Territorium. Er hat 11 Körner im Maul. Er soll wie unten skizziert einen Körner-Weihnachtsbaum ins Territorium zeichnen und dann zu seiner Ausgangsposition zurückkehren.

