

**252-0027**

# **Einführung in die Programmierung**

## **1.0 EBNF**

*Thomas R. Gross*

**Department Informatik  
ETH Zürich**

# Programmierung

- **Eine Programmiersprache gibt uns vor, wie wir Lösung(en) eines Problems beschreiben können.**
  - Mehr Einschränkungen (in der Sprache) machen die Kommunikation leichter
  - .... und erfordern mehr Aufwand bei der Entwicklung
- **Genaue Beschreibung (von Programmiersprachen, Werten, Input, ...) wichtiges Thema für Informatik**

# Programmiersprachen

- **Bestimmen die Anweisungen, die ein Programm enthalten darf**
- **Bestimmen Formate und Wertebereiche, mit denen Programme arbeiten**

# Beispiel



Universität  
Zürich<sup>UZH</sup>

**Waist Circumference und Waist-to-Height-Ratio  
bei Schweizer Stellungspflichtigen 2016**

Entwurf Schlussbericht zuhanden des Bundesamtes für Gesundheit (BAG-  
Vertragsnummer 16.008898)

- **Ein Programm liest für jeden Stellungspflichtigen die Körpergrösse einlesen** (und mehr: Taillenumfang ...)
- **In cm (178), m (1.78 oder 1,78), oder mm (1780)?**
- **Mit welcher Genauigkeit (178.2 cm, 1782 mm, 18 dm)?**
  - Für alle gleich? (178 cm, 178.0 cm falls nur 3 Ziffern)
- **Welche Darstellung (1.782E+03 mm, 1782 mm, 1'782 mm)**

**Viele Optionen ... für ein einfaches Problem!**

# Beispiel

- **Wie können wir die erlaubten Formate beschreiben?**
- **Muss präzise und verständlich sein**
- **Erlaubte Werte (in einem Format) sind eine Sprache**
  - Beschreibung durch Text (in D or E?) -- Missverständnisse möglich
  - Formale Beschreibung (kann automatisch überprüft werden) besser
- **Häufiges Problem ... auch Programmiersprache muss beschrieben werden**
  - Am besten *ein* Formalismus (Menge an Regeln)

# EBNF Notation zur Beschreibung von Sprachen

- ... auch **Programmiersprachen** (müssen damit aber warten)

**E – Extended**

**B – Backus**

**N – Naur *oder* Normal**

**F – Form**

- **Beschreibt die Syntax einer Sprache**
  - Form/Struktur – nichts über die Bedeutung
  - Food for thought: Welche Sprachen kann man damit beschreiben?

# Es gibt Gründe für EBNF in EProg

- **Praktische Beweise relevanter Eigenschaften**
  - Beispiele kommen noch ...
- **Jede EBNF Beschreibung hat zwei Seiten**
  - Liefert eine Menge von Symbolen (die der EBNF Beschreibung genügen)
    - «Programmierung im ganz Kleinen»
  - Erlaubt zu prüfen ob ein Symbol der EBNF Beschreibung genügt
- **«Genügt»: (informell) der Beschreibung entsprechend**
- **«Symbol»: (informell) Zeichen oder Zeichenfolge**

# Übersicht

- Sie lernen die vier elementaren Ausdrucksmöglichkeiten in EBNF kennen
- Sie lernen EBNF Beschreibungen zu lesen und verstehen
- Sie lernen zu entscheiden ob ein Symbol legal ist (für eine EBNF Beschreibung)
- ....



# EBNF

- **Programmierung im ganz Kleinen**
- **Vier Elemente («control forms») die Sie in Java wiederfinden werden**
  - Aufreihung («sequence»)
  - Entscheidung («decision») – Auswahl und Option
  - Wiederholung («repetition»)
  - Rekursion («recursion»)

# EBNF

- **Beschreibungen haben einen Namen und diese Namen können wieder verwendet werden**
  - Um kompliziertere Beschreibungen zu erstellen
- **Erstellen einer EBNF Beschreibung  $\leftrightarrow$  Programmieren in Java**
  - Ähnliche Schritte
- **EBNF eine *formale* Beschreibung**
  - Präzise und verständlich

# 1.1 EBNF Regeln und Beschreibungen

- **EBNF Beschreibung besteht aus Menge von EBNF Regeln**
  - Menge: Reihenfolge der Regeln unwichtig
  - Jede Regel gibt an welche Symbole erlaubt sind
    - Beispiel: EBNF Regel für eine Ziffer
- **EBNF Beschreibung gibt an welche Symbole erlaubt sind**
  - Erlaubt: den Regeln entsprechend gebildet
    - Wir werden das noch präzisieren
  - Beispiel: EBNF Beschreibung für 2-stellige positive ganze Zahlen
    - Verwendet (möglicherweise) EBNF Regeln für Ziffern

# Wie sieht eine EBNF Regel aus?

- **Drei Bestandteile:**

- Linke-Seite (Left-Hand Side, LHS)
- Rechte-Seite (Right-Hand Side, RHS)
- $\Leftarrow$

$\Leftarrow$  trennt LHS von RHS, ausgesprochen «ist definiert als»

- d.h. **LHS  $\Leftarrow$  RHS**

- **LHS**

- Ein Wort (kursiv, kleingeschrieben) – der Name der EBNF Regel

## ■ RHS

- Die genaue Beschreibung für den Namen (d.h., der LHS)
- Kann enthalten
  - Zeichen (stellen das Zeichen da, d.h. wir erwarten dieses Zeichen und kein anderes)
  - ...

# Beispiel EBNF Regel

# Beispiel EBNF Regel

- $ziffer\_null \Leftarrow \emptyset$

---

- $ziffer\_eins \Leftarrow 1$

---

- $ziffer\_null \Leftarrow \emptyset$

- $ziffer\_eins \Leftarrow 1$

## ■ **RHS**

- Die genaue Beschreibung für den Namen (d.h., der LHS)
- Kann enthalten
  - Zeichen (stellen das Zeichen da, d.h. wir erwarten dieses Zeichen und kein anderes)
  - Namen (von EBNF Regeln)
  - ...



# Beispiel EBNF Regel

■ *ziffer\_null*  $\Leftarrow$   $\emptyset$

---

■ *ziffer\_eins*  $\Leftarrow$  1

---

■ *ziffer\_null*  $\Leftarrow$   $\emptyset$

*zahl\_n0*  $\Leftarrow$  *ziffer\_null*

---

## ■ **RHS**

- Die genaue Beschreibung für den Namen (d.h., der LHS)
- Kann enthalten
  - Zeichen (stellen das Zeichen da, d.h. wir erwarten dieses Zeichen und kein anderes)
  - Namen (von EBNF Regeln)
  - Kombinationen der vier Kontrollelemente («control forms») (auf folgenden Seiten)

- **EBNF Regel besteht aus:**

- Linke-Seite (Left-Hand Side, LHS)
- Rechte-Seite (Right-Hand Side, RHS)
- $\Leftarrow$  (trennt LHS von RHS, ausgesprochen «ist definiert als»)

- **LHS**

- Ein Wort (kursiv, kleingeschrieben) – der Name der EBNF Regel

- **RHS**

- Die genaue Beschreibung für den Namen (d.h., der LHS) durch
  - Namen (von EBNF Regeln) – kursiv und kleingeschrieben
  - Zeichen (stellen das Zeichen da, d.h. wir erwarten dieses Zeichen und kein anderes) – nicht kursiv
  - Kombinationen der vier Kontrollelemente («control forms») (auf folgenden Seiten)

# Müssen wir ein Alphabet angeben?

- **Gibt es «Zwischenräume»?**
  - Nur wenn Sie dafür ein (oder mehr) Zeichen definieren

# EBNF – statt *kursiv* zwei Klammern

- $digit\_9 \Leftarrow 9$
- $\langle digit\_9 \rangle \Leftarrow 9$

# EBNF

- **Vier Kombinationsmöglichkeiten («control forms») die Sie in Java wiederfinden werden**
  - Aufreihung («sequence»)
  - Entscheidung («decision») – Auswahl und Option
  - Wiederholung («repetition»)
  - Rekursion («recursion»)
- **Kombinieren (in der RHS) EBNF Regeln**

# 1.1.1 Control form: Aufreihung

- **Aufreihung («sequence»)**
- **Folge von Zeichen («Buchstabe», «Ziffern», ...)**
  - Von links nach rechts gelesen
  - Reihenfolge der Zeichen ist wichtig
- **Aufreihung Beispiel**

# Beschreibung: Menge von Regeln

## ■ Aufreihung

- Regel von links nach rechts gelesen
- Reihenfolge der Zeichen ist wichtig

## ■ Aufreihung Beispiel – verschiedene Mengen, selbes Symbol

*buchstabe\_8*  $\Leftarrow$  8  
*raum\_1*  $\Leftarrow$  *buchstabe\_d* *buchstabe\_2* *buchstabe\_8*  
*buchstabe\_2*  $\Leftarrow$  2  
*buchstabe\_d*  $\Leftarrow$  D

**Menge3**

*buchstabe\_d*  $\Leftarrow$  D  
*buchstabe\_8*  $\Leftarrow$  8  
*raum\_1*  $\Leftarrow$  *buchstabe\_d* *buchstabe\_2* *buchstabe\_8*  
*buchstabe\_2*  $\Leftarrow$  2

**Menge2**

*buchstabe\_d*  $\Leftarrow$  D  
*buchstabe\_2*  $\Leftarrow$  2  
*buchstabe\_8*  $\Leftarrow$  8  
*raum\_1*  $\Leftarrow$  *buchstabe\_d* *buchstabe\_2* *buchstabe\_8*

**Menge1**

*raum\_1*  $\Leftarrow$  D 2 8

**Menge4**



## 1.1.2 Kombinieren mit Entscheidungen

- **Auswahl – aus Alternativen**

- Eine Menge von Alternativen
  - Reihenfolge unwichtig
- Durch | (gesprochen *senkrechter Strich*) («stroke») getrennt
- Alternativen folgen den EBNF Bestimmungen für die RHS (Rechte-Seite)

- **Auswahl Beispiel**

*digit*  $\Leftarrow$  1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

-----  
*raum*  $\Leftarrow$  E 12 | D 28

*buchstabe\_d*  $\Leftarrow$  D

*buchstabe\_2*  $\Leftarrow$  2

*buchstabe\_8*  $\Leftarrow$  8

*buchstabe\_e*  $\Leftarrow$  E

*buchstabe\_1*  $\Leftarrow$  1

*raum\_1*  $\Leftarrow$  *buchstabe\_d* *buchstabe\_2* *buchstabe\_8*

*raum\_2*  $\Leftarrow$  *buchstabe\_e* *buchstabe\_1* *buchstabe\_2*

*raum*  $\Leftarrow$  *raum\_1* | *raum\_2*

# Wofür wir ( und ) brauchen

- **A B | C**
  - Was ist legal?
  - AB und C *oder*
  - AC und AB
- **Um Unklarheit zu vermeiden verwenden wir ( und )**
  - A ( B | C )
  - ( A B ) | C
    - Klammer schafft Klarheit

# Kombinieren mit Optionen

## ■ Option

- Element(e) in [ und ] (eckige Klammern) («square bracket»)
  - Element muss EBNF Bestimmungen für die RHS folgen
- Kann gewählt werden, muss aber nicht

## ■ Drei Beispiele

- $initials \Leftarrow T[R]G$
- $raum \Leftarrow ML[(D\ 28) | (E\ 12)]$
- $zahl \Leftarrow 0x0[0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]$

# Beispiel

- **Oft Auswahl ( $..|..$ ) und Option ( $[ .. ]$ ) kombiniert**
- EBNF Beschreibung *zahl*
  - $digit \Leftarrow 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0$
  - $vorzeichen \Leftarrow [ + | - ]$
  - $zahl \Leftarrow vorzeichen digit$
- 1, +2, -3, +0, 4 ... sind legale Symbole
- **Wie drücken wir aus, dass nicht gewählt wurde?**
  - $\epsilon$  (Ausgesprochen «epsilon») – die leere Zeichenfolge
  - $\epsilon$  erscheint nicht in Symbolen

## Ist +4 legal?

1.  $zahl \Leftarrow \text{vorzeichen digit}$

2.  $\Leftarrow + digit$

3.  $\Leftarrow + 4$

## Ist 4 legal?

1.  $zahl \Leftarrow \text{vorzeichen digit}$

2.  $\Leftarrow \epsilon digit$

3.  $\Leftarrow 4$

$digit \Leftarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 0$   
 $\text{vorzeichen} \Leftarrow [+ \mid -]$   
 $zahl \Leftarrow \text{vorzeichen digit}$

# Wenn nicht gewählt wird

*raum*  $\Leftarrow$  M L [ ( D 28 ) | ( E 12 ) ]

- [ ... ] kann gewählt werden, muss aber nicht
  - M L ist mögliches Symbol

- Vielfältige Kombinationsmöglichkeiten

*zahlen*  $\Leftarrow$  [ 0 | 1 ] [ 2 | 3 ] Z

Möglich sind:

# Wenn nicht gewählt wird

*raum*  $\Leftarrow$  M L [ ( D 28 ) | ( E 12 ) ]

- [ ... ] kann gewählt werden, muss aber nicht
  - M L ist mögliches Symbol

- **Vielfältige Kombinationsmöglichkeiten**

*zahlen*  $\Leftarrow$  [ 0 | 1 ] [ 2 | 3 ] Z

**Möglich sind:** 02Z 03Z 12Z 13Z



# Wenn nicht gewählt wird

*raum*  $\Leftarrow$  M L [ ( D 28 ) | ( E 12 ) ]

- [ ... ] kann gewählt werden, muss aber nicht
  - M L ist mögliches Symbol

- Vielfältige Kombinationsmöglichkeiten

*zahlen*  $\Leftarrow$  [ 0 | 1 ] [ 2 | 3 ] Z

Möglich sind: 02Z 03Z 12Z 13Z

0Z 1Z 2Z 3Z

# Wenn nicht gewählt wird

*raum*  $\Leftarrow$  M L [ ( D 28 ) | ( E 12 ) ]

- [ ... ] kann gewählt werden, muss aber nicht
  - M L ist mögliches Symbol

- Vielfältige Kombinationsmöglichkeiten

*zahlen*  $\Leftarrow$  [ 0 | 1 ] [ 2 | 3 ] Z

Möglich sind: 02Z 03Z 12Z 13Z

0Z 1Z 2Z 3Z

Z