

252-0027

**Einführung in die Programmierung
Übungen**

Woche 1: IntelliJ und EBNF

Timo Stucki

**Departement Informatik
ETH Zürich**

Organisatorisches

- Mein Name: *Timo Stucki*
- Bei Fragen: *tistucki@student.ethz.ch*
 - Mails bitte mit «[EProg25]» im Betreff
- Neue Aufgaben: **Dienstag Abend** (im Normalfall)
- Abgabe der Übungen bis **Dienstag Abend (23:59)** Folgewoche
 - Abgabe immer via Git
 - Lösungen in separatem Projekt auf Git

Was ihr von dieser Gruppe erwarten könnt

- Es wird in allen Gruppen der gleiche Stoff behandelt!
- Hier: Wenn Zeit bleibt, Fokus auf Prüfungsvorbereitung und sonstige Tipps fürs Basisjahr
- Discord: timostucki

Webseite



- timostucki.com

IntelliJ bereits früher installiert?

- Neue Version: **Java 21**
 - Java 21 aus dem Oracle Archiv installieren -> JRE 21 in IntelliJ als Standard auswählen
 - Informationen zu Fehlermeldungen und Behebungen direkt auf der Website
 - Sonst IntelliJ und Java deinstallieren, JDK 21 und IntelliJ neu installieren
 - Sehr wichtig, weil Korrektur der Bonusaufgaben auch Java 21 benutzt



Mehr bei IntelliJ

IntelliJ Installation

- Instruktion zur Installation findet ihr in Übung 1.
 - <https://lec.inf.ethz.ch/infk/eprog/2025/exercises/sheets/u01.pdf>

EBNF Notation

- In alten Prüfungen wird oft kursiv verwendet für EBNF Regeln.
- *digit* statt <digit>

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

digit $\Leftarrow$ $\boxed{0} \mid \boxed{1} \mid \boxed{2} \mid \boxed{3} \mid \boxed{4} \mid \boxed{5} \mid \boxed{6} \mid \boxed{7} \mid \boxed{8} \mid \boxed{9}$

separator $\Leftarrow$ $\boxed{-}$

char $\Leftarrow$ $\boxed{A} \mid \boxed{B} \mid \boxed{C} \mid \boxed{D} \mid \boxed{E} \mid \boxed{F} \mid \boxed{a} \mid \boxed{b} \mid \boxed{c} \mid \boxed{d} \mid \boxed{e} \mid \boxed{f}$

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [$\boxed{.}$ *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } $\boxed{h}$

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

1245

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

digit $\Leftarrow$ $\boxed{0} \mid \boxed{1} \mid \boxed{2} \mid \boxed{3} \mid \boxed{4} \mid \boxed{5} \mid \boxed{6} \mid \boxed{7} \mid \boxed{8} \mid \boxed{9}$

separator $\Leftarrow$ $\boxed{-}$

char $\Leftarrow$ $\boxed{A} \mid \boxed{B} \mid \boxed{C} \mid \boxed{D} \mid \boxed{E} \mid \boxed{F} \mid \boxed{a} \mid \boxed{b} \mid \boxed{c} \mid \boxed{d} \mid \boxed{e} \mid \boxed{f}$

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [$\boxed{.}$ *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } $\boxed{h}$

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

00972

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

00100h

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

1a00

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

1a00h

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ _

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

1_000_000

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

001ab.001h

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

209AB

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

digit $\Leftarrow$ $\boxed{0} \mid \boxed{1} \mid \boxed{2} \mid \boxed{3} \mid \boxed{4} \mid \boxed{5} \mid \boxed{6} \mid \boxed{7} \mid \boxed{8} \mid \boxed{9}$

separator $\Leftarrow$ $\boxed{-}$

char $\Leftarrow$ $\boxed{A} \mid \boxed{B} \mid \boxed{C} \mid \boxed{D} \mid \boxed{E} \mid \boxed{F} \mid \boxed{a} \mid \boxed{b} \mid \boxed{c} \mid \boxed{d} \mid \boxed{e} \mid \boxed{f}$

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [$\boxed{.}$ *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } $\boxed{h}$

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

4.9901

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

deadface

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ _

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

4_000.0

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

00100H

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

001ab.001

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

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

separator $\Leftarrow$ -

char $\Leftarrow$ A | B | C | D | E | F | a | b | c | d | e | f

num $\Leftarrow$ *digit* { [*separator*] *digit* }

int $\Leftarrow$ *digit* { *digit* }

real $\Leftarrow$ *digit* { *digit* } [. *digit* { *digit* }]

cd $\Leftarrow$ *char* | *digit*

hexa1 $\Leftarrow$ *cd* { *cd* }

hexa2 $\Leftarrow$ *digit* { *digit* } h

hexa $\Leftarrow$ *hexa1* | *hexa2*

value $\Leftarrow$ *num* | *real* | *int* | *hexa*

0x0ABC

Zusatzaufgaben

- Erstellen Sie eine Beschreibung `<palindrome>`, welche als legale Symbole alle Zahlen zulässt, die von vorne und hinten gleich gelesen werden und die nur die Ziffern von 1 bis 4 verwenden. Beispiele sind 11, 232, 444
- Erstellen Sie eine Beschreibung `<five>`, welche alle Summen von positiven Zahlen zulässt, welche 5 ergeben. Beispiele sind "1 + 4", "2 + 1 + 1 + 1", "5"
- Erstellen Sie eine Beschreibung für `<oddEight>`, die alle Zahlen enthält, in denen die Ziffer 8 ungerade oft vorkommt.

EProg Overview

- **Note: 1/3 Theorie, 2/3 Programmieren**
- **Prüfung in zwei Teilen:**
 - **Theorie schriftlich (40 min), Programmieren (120 min)**

Vorbereitung

- Theorieprüfung
 - Vorlesungsinhalt, Clicker während Vorlesung / Kahoots, Theorie Aufgaben, Alte Prüfungen (Lernphase) (, inoffizielles/PVW Skript)
- Programmieren
 - Weekly Assignments, regelmässiges Üben, Alte Prüfungen (Lernphase)

Webseite



- timostucki.com

Gegeben sei in Abbildung 1 die EBNF-Beschreibung von q .

$$\begin{aligned}x &\Leftarrow \boxed{a} \, y \, \boxed{a} \mid \boxed{a} \, \boxed{b} \\y &\Leftarrow \boxed{b} \, x \, \boxed{b} \\z &\Leftarrow \boxed{a} \, \boxed{b} \, z \mid z \, \boxed{a} \, \boxed{b} \mid \boxed{a} \, \boxed{a} \\q &\Leftarrow x \mid z\end{aligned}$$

Abbildung 1: **EBNF-Beschreibung** von q

Geben Sie für jeden folgenden Ausdruck an, ob er nach der EBNF-Beschreibung von q in Abbildung 1 gültig ist.

Ausdruck	Gültig	Ungültig	Ausdruck	Gültig	Ungültig
a a a b b			a b a b		
a b a b a b a			a b a b b a		
a b a b a a			a b a b a b b a b a		