

## Übungen zu Einführung in die Informatik I

### Eingeschränkte Java-Grammatik mit funktionalen und imperativen Konstrukten

Zur Übersicht über die bisher in den Übungen behandelten Elemente der Programmiersprache Java geben wir eine Grammatik in erweiterter Backus-Naur-Form an.

Alle gegenüber dem Arbeitsblatt 2b geänderten bzw. erweiterten Produktionen sind durch **Fett-druck** hervorgehoben. Neu hinzugekommene Produktionen sind *kursiv* dargestellt.

<System>	::=	<Klasse> <Klasse> <sup>+</sup>
<Klasse>	::=	'class' <Klassenbezeichner> [ 'extends' <Klassentyp> ] '{ ( <Attributvariable>   <Konstruktordefinition>   <Methodendefinition> ) <sup>*</sup> '}'
<Attributvariable>	::=	<Deklarationsanweisung>
<Variablendeklaration>	::=	<Typ> <Variablenbezeichner>
<Typ>	::=	<Grundtyp>   <Klassentyp>
<Grundtyp>	::=	'int'   'double'   'boolean'   'char'
<Klassentyp>	::=	'String'   'Object'   <Klassenbezeichner>
<Konstruktordefinition>	::=	<Klassenbezeichner> '(' <Parameterliste> ')' '{ <Konstruktorrumpf> '}'
<Konstruktorrumpf>	::=	<Anweisungssequenz>
<Methodendefinition>	::=	<Typ> <Methodenbezeichner> '(' <Parameterliste> ')' '{ <Methodenrumpf> '}'
<Parameterliste>	::=	[ <Variablendeklaration> ]   <Variablendeklaration> (',' <Variablendeklaration>) <sup>+</sup>
<Methodenrumpf>	::=	<Anweisungssequenz>
<Ausdruck>	::=	<Variablenbezeichner>   <Konstruktoraufruf>   <Methodenaufruf>   <Wahrheitswert>   <Zahl>   <Zeichenkette>   <BedingterAusdruck>   <Zuweisung>
<Konstruktoraufruf>	::=	'new' <Klassenbezeichner> '(' <Argumentliste> ')'
<Methodenaufruf>	::=	<Methodenbezeichner> '(' <Argumentliste> ')'
<Argumentliste>	::=	[ <Ausdruck> ]   <Ausdruck> (',' <Ausdruck>) <sup>+</sup>
<BedingterAusdruck>	::=	<Bedingung> '?' <Ausdruck> ':' <Ausdruck>
<Bedingung>	::=	<BoolescherAusdruck>
<Anweisung>	::=	<Deklarationsanweisung>   <Zuweisungsanweisung>   <BedingteAnweisung>   <Verbundanweisung>   <for-Anweisung>   <while-Anweisung>   <Aufrufanweisung>   <Rückgabenanweisung>
<Deklarationsanweisung>	::=	<Variablendeklaration> [ '=' <Ausdruck> ] ';' ;

<i>&lt;Zuweisungsanweisung&gt;</i>	::=	<i>&lt;Zuweisung&gt;</i> ';' ;
<i>&lt;Zuweisung&gt;</i>	::=	<i>&lt;Variablenbezeichner&gt;</i> '=' <i>&lt;Ausdruck&gt;</i>
<i>&lt;Verbundanweisung&gt;</i>	::=	'{' <i>&lt;Anweisungssequenz&gt;</i> '}'
<i>&lt;Anweisungssequenz&gt;</i>	::=	( <i>&lt;Anweisung&gt;</i> )*
<i>&lt;BedingteAnweisung&gt;</i>	::=	'if' '(' <i>&lt;Bedingung&gt;</i> ')' <i>&lt;Anweisung&gt;</i> [ 'else' <i>&lt;Anweisung&gt;</i> ]
<i>&lt;for-Anweisung&gt;</i>	::=	'for' '(' [ <i>&lt;Typ&gt;</i> ] <i>&lt;Zuweisung&gt;</i> ';' <i>&lt;Bedingung&gt;</i> ';' <i>&lt;Zuweisung&gt;</i> ')' <i>&lt;Anweisung&gt;</i>
<i>&lt;while-Anweisung&gt;</i>	::=	'while' '(' <i>&lt;Bedingung&gt;</i> ')' <i>&lt;Anweisung&gt;</i>
<i>&lt;Aufrufanweisung&gt;</i>	::=	<i>&lt;Methodenaufruf&gt;</i> ';' ;
<i>&lt;Rückgabeeanweisung&gt;</i>	::=	'return' <i>&lt;Ausdruck&gt;</i> ';' ;

Einige Nicht-Terminals der Grammatik, wie z.B. *<Zahl>*, *<Zeichenkette>* und *<...bezeichner>* werden an dieser Stelle nicht exakt beschrieben, sollten aber intuitiv verständlich sein.

Zur eindeutigen Abgrenzung müssen viele der Terminals und Nicht-Terminals durch mindestens ein Trennzeichen (Leerzeichen oder Zeilenumbruch) voneinander getrennt sein. Diese Leerzeichen sind aus Gründen der Übersichtlichkeit in den Produktionen weggelassen.

Als Faustregel gilt dabei: Folgen zwei alphanumerische (d.h. nur aus Buchstaben und Ziffern bestehende) Terminals unmittelbar aufeinander, so sind diese durch Trennzeichen voneinander abzugrenzen.

Die Grammatik wird im Laufe der Übungen nach und nach um weitere Elemente, also Java-Programmkonstrukte, erweitert werden, und auf späteren Arbeitsblättern bereitgestellt.