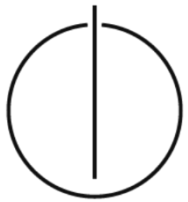


TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK
Praktikum: Grundlagen der Programmierung

Programmierpraktikum

Woche 03 (10.11.2016)

Stefan Berktold
s.berktold@tum.de

Präsenzaufgaben

Heutige Übersicht

Aufgabe 3.1 ***Syntaxbäume***

Quellcode in Syntaxbaum umwandeln (MiniJava Grammatik)

Aufgabe 3.2 ***Kontrollflussdiagramme***

Quellcode → Kontrollflussgraph

Aufgabe 3.3 ***Programmieraufgabe (MiniJava)***

Brettspiel: „Schlangenspiel“ (mit Vorgabe)

Hausaufgaben

Abgabe: 14.11.2016 05:00 Uhr

Aufgabe 3.4 ***Syntaxbäume***

Quellcode in Syntaxbaum umwandeln (MiniJava Grammatik)

Aufgabe 3.5 ***Kontrollflussdiagramme***

Quellcode → Kontrollflussgraph

Aufgabe 3.6 ***Programmieraufgabe (MiniJava)***

Kartenspiel: „17 und 4“

Aufgabe 3.1

Syntaxbaum

Zeichnen Sie für das folgende MiniJava-Programm den Syntaxbaum. Dazu steht Ihnen die Grammatik von MiniJava aus der Vorlesung zur Verfügung.

Eine Zusammenfassung dieser Grammatik finden Sie auf Moodle.

```
int prod, x, n;
x = read();
if (0 < x) {
    prod = 1;
    n = 0;
    while (prod <= x) {
        n = n + 1;
        prod = prod * (-n);
    }
    write(prod);
} else {
    write(n);
}
```

Aufgabe 3.1

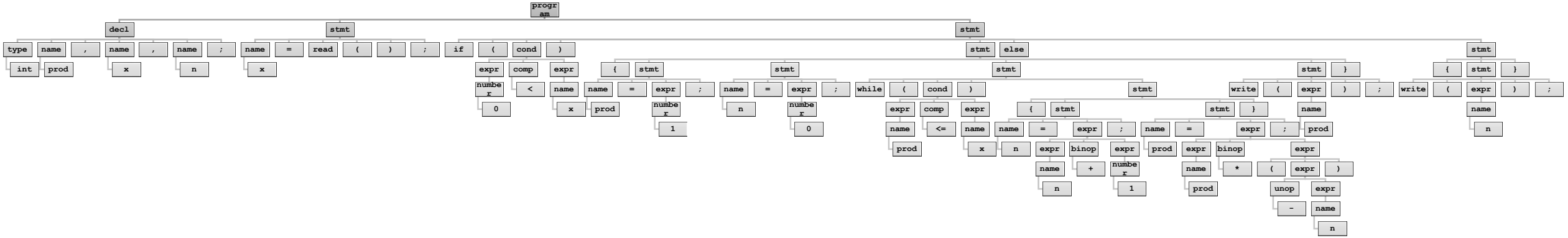
Syntaxbaum

```
int prod, x, n;
x = read();
if (0 < x) {
    prod = 1;
    n = 0;
    while (prod <= x) {
        n = n + 1;
        prod = prod * (-n);
    }
    write(prod);
} else {
    write(n);
}
```

<program>	::=	<decl>* <stmt>*
<decl>	::=	<type> <name> (, <name>)*;
<stmt>	::=	;
	|	{ <stmt>* }
	|	<name> = <expr> ;
	|	<name> = read();
	|	write(<expr>);
	|	if (<cond>) <stmt>
	|	if (<cond>) <stmt> else <stmt>
	|	while (<cond>) <stmt>
<expr>	::=	<number>
	|	<name>
	|	(<expr>)
	|	<unop> <expr>
	|	<expr> <binop> <expr>
<cond>	::=	true
	|	false
	|	(<cond>)
	|	<expr> <comp> <expr>
	|	<bunop> <cond>
	|	<cond> <bbinop> <cond>
<comp>	::=	== | != | <= | < | >= | >
<unop>	::=	-
<binop>	::=	- | + | * | / | %
<bbinop>	::=	&& | ||
<bunop>	::=	!
<type>	::=	int
<name>	::=	letter (letter | digit)*
<number>	::=	digit digit*

Aufgabe 3.1 – Lösung

Syntaxbaum



Aufgabe 3.2

Kontrollflussdiagramm

Zeichnen Sie den Kontrollflussgraphen für das folgende Java-Fragment:

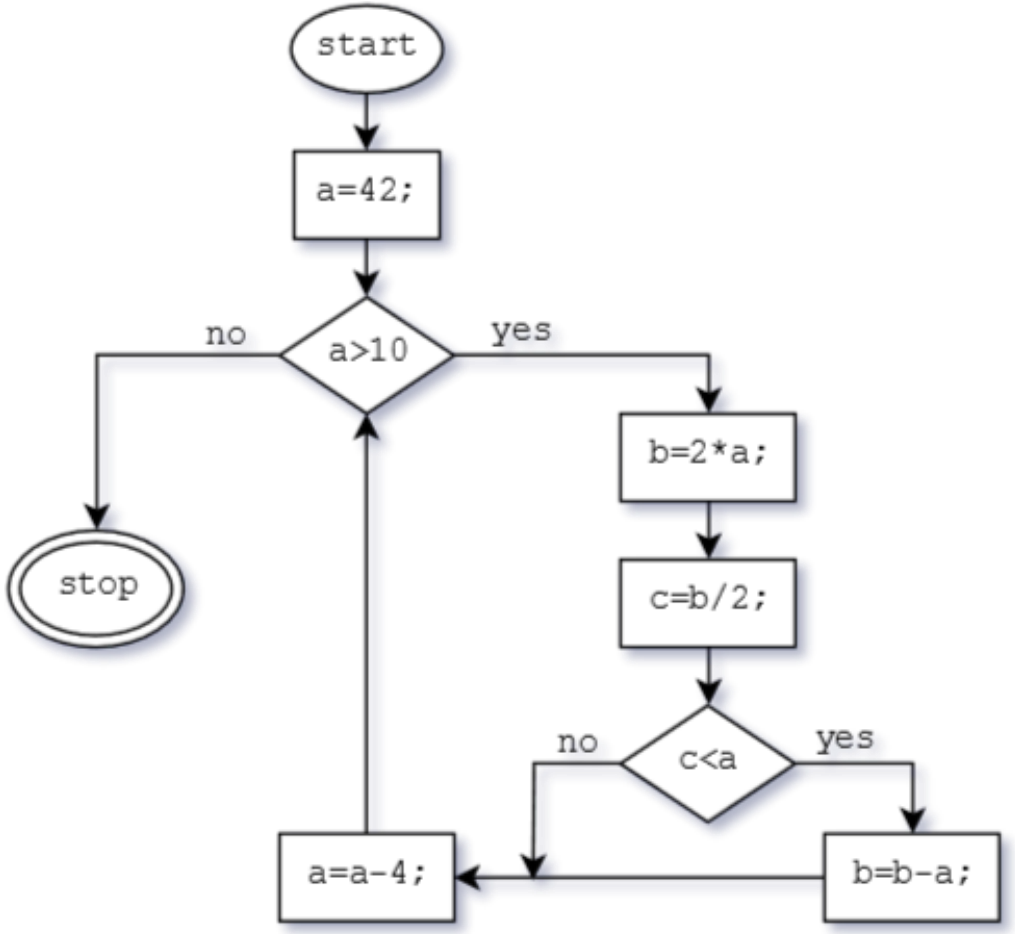
```
int b, c;
for (int a = 42; a > 10; a = a - 4) {
  b = 2 * a;
  c = b / 2;
  if (c < a) {
    b = b - a;
  }
}
```

Hinweis: Zum Zeichnen können Sie z.B. das Programm *LibreOffice Draw*, welches Sie unter http://www.libreoffice.org/ finden, oder die Webseite http://draw.io verwenden. Speichern Sie Ihren Kontrollflussgraphen im *pdf*-Format ab.

Aufgabe 3.2 – Lösung

Kontrollflussdiagramm

```
int b, c;
for (int a = 42; a > 10; a = a - 4) {
  b = 2 * a;
  c = b / 2;
  if (c < a) {
    b = b - a;
  }
}
```



Aufgabe 3.3

Schlangenspiel (Auszug)

In dieser Aufgabe sollen Sie eine Variante des Schlangenspiels für zwei Spieler programmieren. Erstellen Sie dazu ein Programm namens `Schlangenspiel.java`. […]

Implementieren Sie ihre Lösung Schritt für Schritt:

1. Beschränken Sie sich zuerst auf einen Spieler und ein leeres Spielfeld. Das Spiel besteht am Anfang nur aus Würfeln bis die 35 überschritten wird.
2. Beachten Sie nun Leitern und Schlangen – nach dem Würfeln prüfen Sie, ob das Zielfeld leer ist.
3. Beachten Sie nun auch Ketten von Leitern und Schlangen – bewegen Sie einen Spielstein nach dem Würfeln so lange, bis er weder auf einer Schlange noch auf einer Leiter landet.
4. Erweitern Sie Ihr Spiel nun um den zweiten Spieler.

Hinweis: Verwenden Sie die Klasse `Spielfeld`, um das Spielfeld zu zeichnen. Ersetzen Sie dazu `extends MiniJava` durch `extends Spielfeld`. […]

Aufgabe 3.3

Schlangenspiel (Auszug)

Die **Grundidee** des Spiels ist folgende:

- Das Spiel hat die Felder 0 bis 35.
- Jeder Spieler besitzt einen Spielstein.
- Beide Spielsteine starten auf Feld 0
- Es dürfen zwei Spielsteine gleichzeitig auf einem Feld stehen. Im Wechsel wird gewürfelt. Der Spielstein des entsprechenden Spielers wird um die entsprechende Augenzahl vorgerückt.
- Von allen Feldern, die durch 5 teilbar sind, führt eine Leiter nach oben. Wer ein solches Feld erreicht, kommt sofort 3 Felder weiter.
- Erreicht man ein Schlangenfeld (jedes Feld, das durch 7 teilbar ist), rutscht man automatisch um 4 Felder zurück.
- Die Felder 0 und 35 sind weder Leiter- noch Schlangenfelder.
- Leitern und Schlangen treten in Aktion, wenn ein Spielstein dieses Feld erreicht. Es ist daher möglich, Ketten von Schlangen und/oder Leitern zu benutzen.
- Wer zuerst das Feld 35 erreicht oder überschreitet, gewinnt das Spiel.
- Am Ende wird der Sieger ausgegeben.

Geben Sie jeweils das Würfelergebnis und das neue Spielfeld aus.

Aufgabe 3.3 – Lösung

Schlangenspiel (Auszug)

Einen Lösungsvorschlag finden Sie unter http://tutor16.stecrz.de (PW: tutor16).

Die offiziellen Lösungen zu allen Präsenzaufgaben sind voraussichtlich ab

11.11.2016 20:00 Uhr

auf Moodle verfügbar (und dürfen nicht vorab ausgehändigt werden).