Praktikum Grundlagen der Programmierung

Prof. Dr. Harald Räcke, J. Kranz, A. Reuss

WS 2017/18

Übungsblatt 3

Abgabefrist: 13.11.2017 um 05:00 Uhr

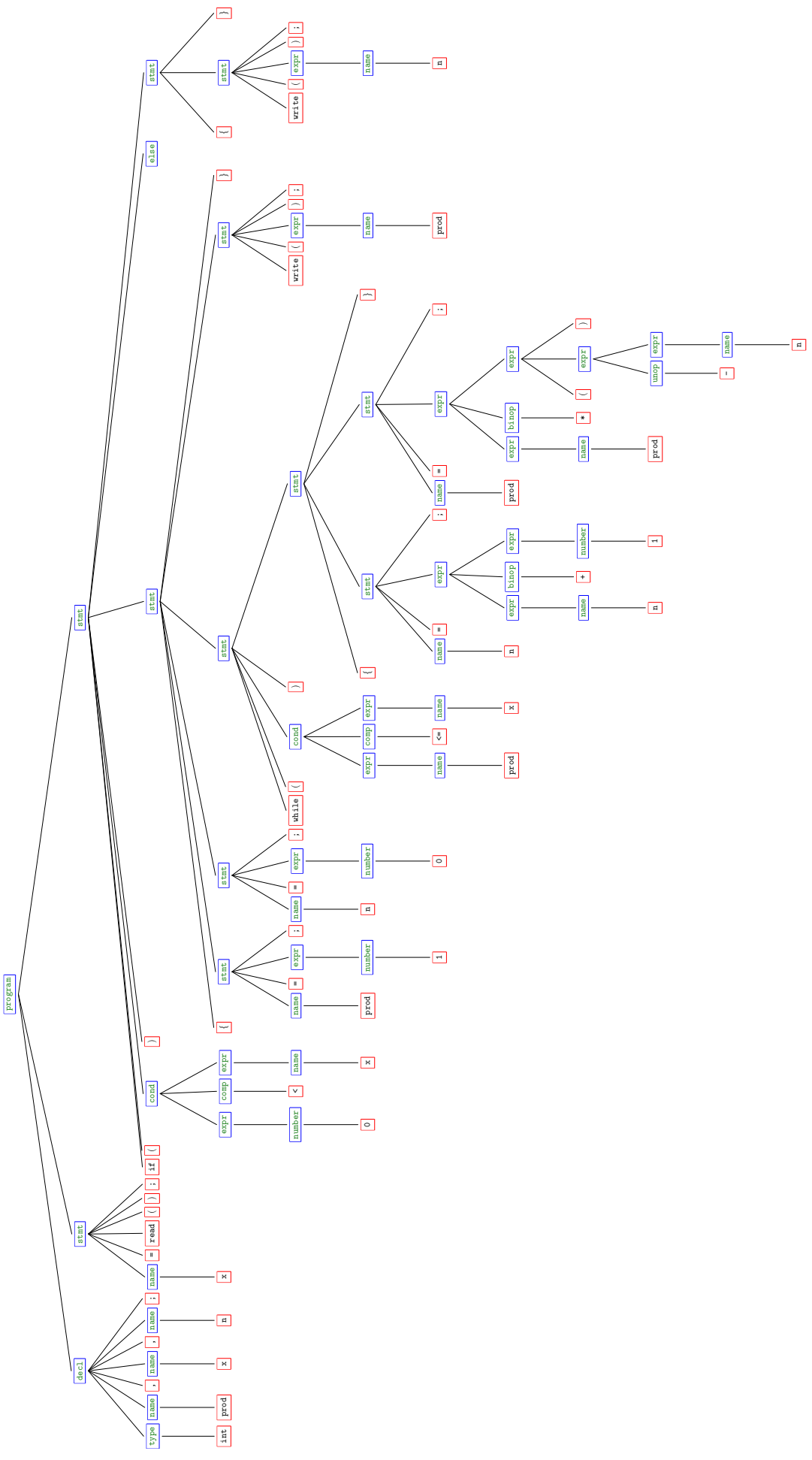
Σ Punkte: 20

Aufgabe 3.1 (P) Syntaxbaum

Zeichnen Sie für das folgende `MiniJava`-Programm den Syntaxbaum. Dazu steht Ihnen die Grammatik von `MiniJava` aus der Vorlesung zur Verfügung (eine Zusammenfassung finden Sie auch auf Moodle).

```
1  int prod, x, n;
2  x = read();
3  if (0 < x) {
4    prod = 1;
5    n = 0;
6    while (prod <= x) {
7      n = n + 1;
8      prod = prod * (-n);
9    }
10   write(prod);
11 } else {
12   write(n);
13 }
```

Lösungsvorschlag 3.1


program
decl
type
int
name
prod
,
name
x
,
name
n
;
stmt
name
x
=
read
(
)
;
stmt
if
(
cond
expr
name
x
comp
<
expr
number
0
)
{
stmt
while
(
cond
expr
name
x
comp
<
expr
name
prod
)
stmt
{
stmt
name
prod
=
expr
number
1
;
stmt
name
n
=
expr
number
0
;
}
stmt
write
(
expr
name
prod
)
;
else
stmt
{
stmt
write
(
expr
name
n
)
;
stmt
{
stmt
name
n
=
expr
name
n
-
expr
number
1
;
stmt
name
prod
=
expr
name
prod
*
expr
(
expr
name
n
-
number
1
)
;
}
}

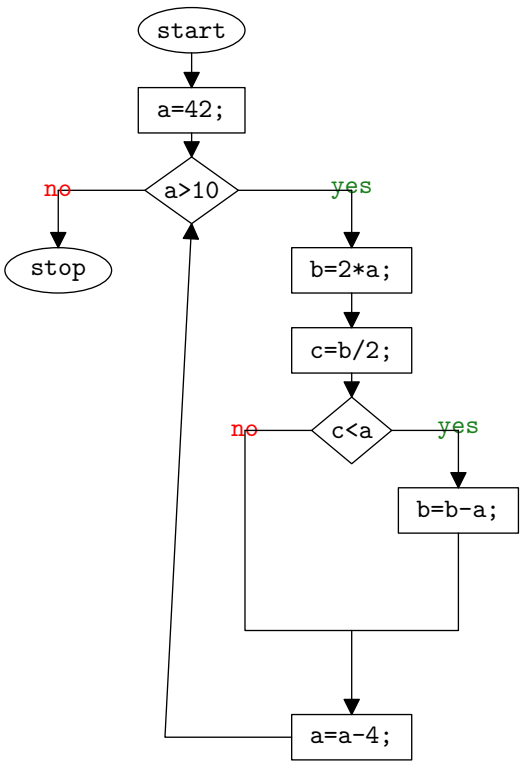
Aufgabe 3.2 (P) Kontrollflussdiagramm

Zeichnen Sie den Kontrollflussgraphen für das folgende `MiniJava`-Programm:

```
1  int a, b, c;
2  a = 42;
3  while (a > 10) {
4    b = 2*a;
5    c = b / 2;
6    if (c < a) {
7      b=b-a;
8    }
9    a = a - 4;
10 }
```

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

Lösungsvorschlag 3.2



Ein Semikolon nach einer Zuweisung oder nach einem `write`-Aufruf ist optional.

Aufgabe 3.3 (P) Kleinster Paul, größte Pauline

Schreiben Sie ein `MiniJava`-Programm namens `MinMax.java`, das in einem Array von ganzen Zahlen den kleinsten und den größten Wert findet. Das Programm soll sich wie folgt verhalten:

- Zunächst fragt das Programm ab, wie viele Zahlen in das Array eingegeben werden sollen.
- Dann werden die Zahlen eingegeben und in einem Array gespeichert.
- Anschließend wird das Array durchsucht und in **einem Durchgang** soll sowohl die kleinste als auch die größte Zahl gefunden werden.
- Schließlich sollen die kleinste und die größte Zahl ausgegeben werden.

Lösungsvorschlag 3.3

Die Lösung befindet sich in der Datei `MinMax.java`.

Aufgabe 3.4 (P) Pascal und sein Dreieck

Das Pascalsche Dreieck wird Schritt für Schritt, beginnend mit Zeile 0, aufgebaut. Dazu berechnet man die n-te Zeile aus der $(n-1)$-ten Zeile wie folgt:

- Die Anzahl der Elemente von Zeile n ist $n+1$.
- Die erste und letzte Zahl jeder Zeile ist stets die 1.
- Das i-te Element der Zeile n entspricht der Summe des i-ten und des $(i-1)$-ten Elements der Zeile $(n-1)$.

Beispiel: Pascalsches Dreieck mit 5 Zeilen:

Zeilenindex n					
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

Schreiben Sie ein `MiniJava`-Programm, welche das Pascalsche Dreieck berechnet. Das Programm soll dabei zunächst die Anzahl der Zeilen des Dreiecks vom Benutzer einlesen; gibt die Benutzerin eine negative Anzahl Zeilen ein, soll sie zur Wiederholung der Eingabe aufgefordert werden. Anschließend soll das Pascalsche Dreieck in einem Array berechnet und schließlich ausgegeben werden.

Hinweis: Für die Anzahl der Einträge e im Dreieck mit m Zeilen gilt:

$$e = \sum_{i=1}^{m}(i) = \frac{m*(m+1)}{2}$$

Lösungsvorschlag 3.4

Die Lösung befindet sich in der Datei `Pascal.java`.

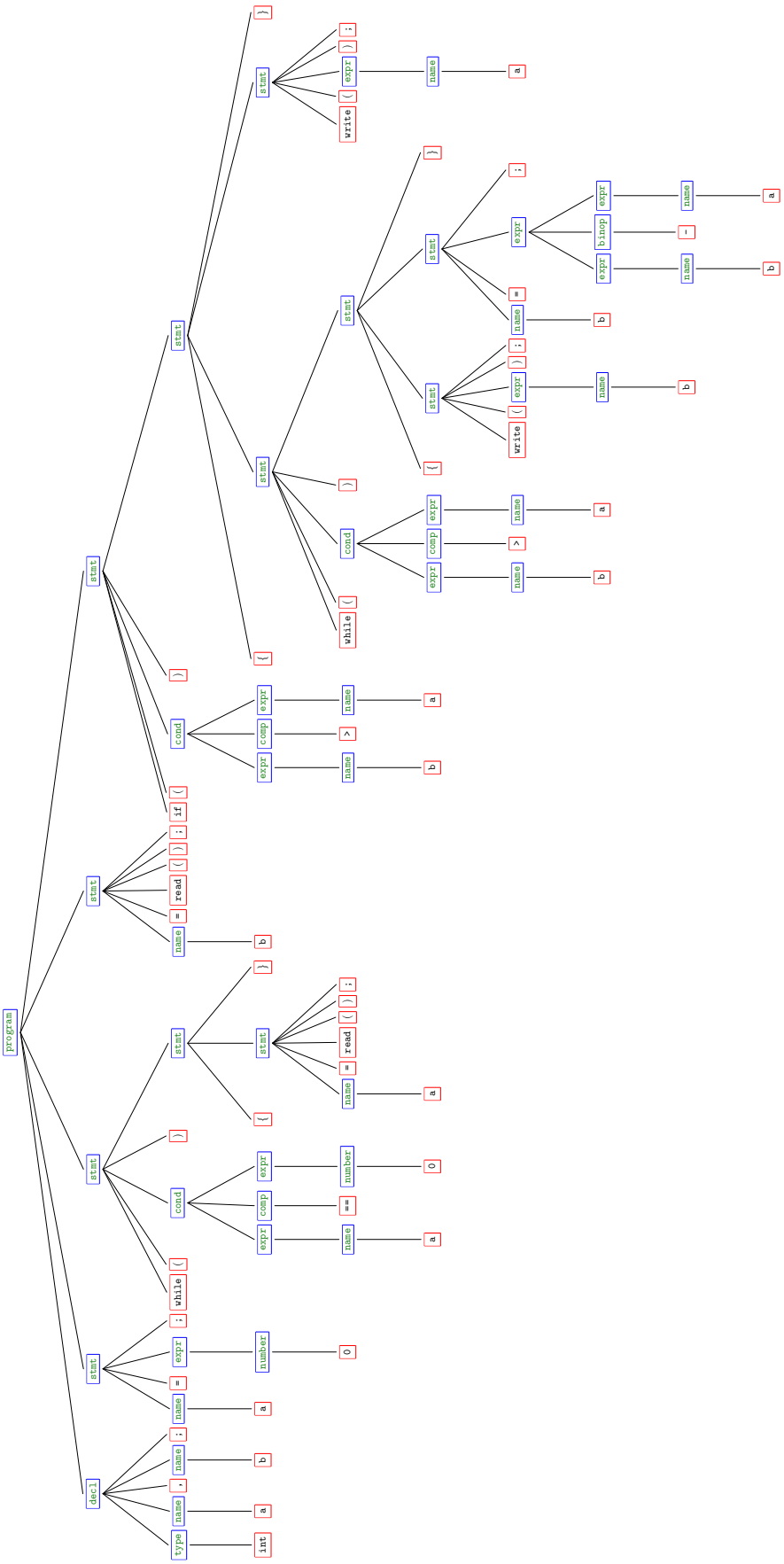
Die Hausaufgabenabgabe erfolgt über Moodle. Bitte geben Sie Ihren Code als UTF8-kodierte (ohne BOM) Textdatei(en) mit der Dateiendung `.java` ab. Geben Sie **keine** Projektdateien Ihrer Entwicklungsumgebung ab. Geben Sie **keinen** kompilierten Code ab (`.class`-Dateien). Geben Sie Ihren Code **nicht** als Archiv (z.B. als `.zip`-Datei) ab. Nutzen Sie **keine** Packages. Achten Sie darauf, dass Ihr Code kompiliert. Auf diesem Blatt gibt auch Aufgaben, zu denen die Lösung kein Programm ist. Geben Sie diese Aufgaben in einer einzelnen, korrekt orientierten und gut lesbaren PDF-Datei ab. Hausaufgaben, die sich nicht im vorgegebenen Format befinden, werden nur mit Punktabzug oder gar nicht bewertet.

Aufgabe 3.5 (H) Syntaxbaum [5 Punkte]

Zeichnen Sie für das folgende `MiniJava`-Programm den Syntaxbaum. Dazu steht Ihnen die Grammatik von `MiniJava` aus der Vorlesung zur Verfügung (eine Zusammenfassung finden Sie auch auf Moodle).

```
1  int a, b;
2
3  a = 0;
4  while (a == 0) {
5    a = read();
6  }
7  b = read();
8  if (b > a) {
9    while (b > a) {
10     write(b);
11     b = b - a;
12   }
13   write(a);
14 }
```

Lösungsvorschlag 3.5



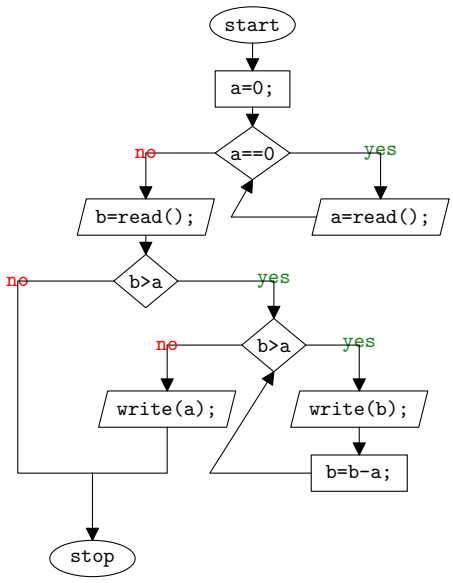
Aufgabe 3.6 (H) Kontrollflussdiagramm [5 Punkte]

Zeichnen Sie den Kontrollflussgraphen für folgendes `MiniJava`-Programm:

```
1  int a, b;
2
3  a = 0;
4  while (a == 0) {
5    a = read();
6  }
7  b = read();
8  if (b > a) {
9    while (b > a) {
10     write(b);
11     b = b - a;
12   }
13   write(a);
14 }
```

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

Lösungsvorschlag 3.6



Korrekturbemerkung: ½ Punkt Abzug für jeden Fehler

Aufgabe 3.7 (H) Feldverarbeitung [5 Punkte]

In dieser Aufgabe werden einfache Operationen auf Arrays geübt. Lesen Sie dazu zunächst ein Array a vom Benutzer ein. Der Benutzer soll dazu nach einer gewünschten Größe n für das Array gefragt werden und anschließend nacheinander die Elemente des Arrays eingeben. Das Array muss mindestens 2 Elemente enthalten; gibt die Benutzerin eine kleinere Größe ein, soll sie auf ihren Irrtum hingewiesen und um erneute Eingabe gebeten werden. Nutzen Sie diesen Teil des Programms für die folgenden Teilaufgaben.

1. Schreiben Sie ein Programm, welches die Elemente an geraden Indices addiert und von der Summe die Elemente an ungeraden Indices subtrahiert.

2. Schreiben Sie ein Programm, welches das zweitgrößte Element des Arrays findet und ausgibt. Gehen Sie davon aus, dass es im Feld keine Wiederholungen von Zahlen gibt.

3. Schreiben Sie ein Programm, welches jeweils zwei benachbarte Elemente addiert und das erste mit der Summe überschreibt. Hat das Array eine ungerade Anzahl Elemente, so bleibt am Ende des Arrays ein einzelnes Element übrig. Dieses soll unverändert bleiben. Das Ergebnis-Feld soll ausgegeben werden.

Lösungsvorschlag 3.7

Die Lösung befindet sich in der Datei `Feldverarbeitung.java`.

Korrekturhinweise:

1. Studenten, die *benachbarte Elemente* derart verstanden haben, dass das Programm in jedem Schritt nur ein Element fortschreitet (vgl. Fragen auf Piazza) erhalten keinen Abzug.

Bewertungsschema:

1. 1 Punkt

2. 2.5 Punkte

3. 1.5 Punkte

Aufgabe 3.8 (H) Palina [5 Punkte]

In dieser Aufgabe sollen Sie ein Programm schreiben, welches für eine Zahl prüft, ob es sich bei der Zahl um ein Palindrom handelt. Eine Zahl ist genau dann Palindrom, wenn sie vorwärts und rückwärts gelesen den gleichen Wert darstellt.

Zunächst soll der Benutzer nach einer positiven Zahl gefragt werden. Wie üblich wird die Anfrage wiederholt, bis die Benutzerin tatsächlich eine positive Zahl eingibt. Die eingegebene Zahl soll anschließend in ein Array von Ziffern überführt werden, auf dem schließlich die Palindrom-Eigenschaft geprüft wird.

Die Zahl soll durch einen `read()`-Aufruf eingelesen werden. Sie darf zu keinem Zeitpunkt in einen String umgewandelt werden. Verwenden Sie ausschließlich MiniJava-Methoden. Verwenden Sie außerdem nicht die Methode `Math.log10(double a)`.

Lösungsvorschlag 3.8

Die Lösung befindet sich in der Datei `Palindrom.java`.

Korrekturhinweise:

- In Piazza wurde die 0 korrekt als Palindrom bezeichnet, ohne jedoch die Studenten noch einmal explizit darauf hinzuweisen, dass diese eigentlich keine erlaubte Eingabe darstellt (@185). Aus diesem Grund soll nichts abgezogen werden, wenn Studenten die 0 als Eingabe zulassen und ausgeben, dass es sich um ein Palindrom handelt.

Bewertungsschema:

1. Verwendung von Strings zur Prüfung der Palindromeigenschaft: Maximal 2 Punkte insgesamt
2. Fehlerhafte Eingabeprüpfung: 0.5 Punkte Abzug
3. Einlesen der Zahl in ein Feld: 2.5 Punkte
4. Testen der Palindromeigenschaft auf dem Feld: 2.5 Punkte