

Programmierpraktikum: Sommersemester 2023

Aufgaben Tag 2

Erinnerung: Um für die Aufgaben Punkte zu erhalten, müssen Sie eine Form von Fehlerbehandlung haben. Insbesondere sollte Ihr Programm mit unerwarteten Eingaben umgehen können.

Aufgabe 1: He speaks Parsertongue!

20 Punkte

Heute wollen wir unseren `Interpreter` fertigstellen. Verwenden Sie Ihre Lösung von gestern und bearbeiten Sie diese weiter.

- a) Implementieren Sie nun boolesche Ausdrücke in Ihrer Methode. Ein boolescher Ausdruck kann wie folgt aussehen:

```
<booleanExpression> = <boolean> ^ <boolean>
<booleanExpression> = <boolean> v <boolean>
<booleanExpression> = ¬<boolean>
<booleanExpression> = <number> < <number>
<booleanExpression> = <number> > <number>
<booleanExpression> = <number> = <number>
```

Verwenden Sie das folgende Programm, um Ihre Implementierung zu testen:

```
var lineColor = #000000ff
var bool = 1 < 2
var degree = 0
var degree = degree + 90
draw circle lineColor #f0daaaaa 2 40 20 40 40 degree
draw triangle lineColor #f0daaaaa 2 40 20 10 20 degree
draw quad lineColor #00ffffaa 2 20 20 5 20 degree
```

- b) Kommen wir nun zum Kontrollfluss. Wir wollen eine Struktur für `if-then-else` einführen. Die Struktur hat die folgende Form, wobei `else`-Fälle optional sein sollen:

```
if <booleanExpression>
then
<statements>
else
<statements>
fi
```

Implementieren Sie `if-then-else` wie beschrieben. Verwenden Sie das folgende Programm, um Ihre Implementierung zu testen:

```
var lineColor = #000000ff
var bool = 1 < 2
var degree = 0
if bool
then
draw circle lineColor #00ffffaa 2 30 20 10 20 degree
fi
var degree = degree + 90
if 2 = 1
then
draw circle lineColor #f0daaaaa 2 40 20 40 40 degree
else
```

```
draw triangle lineColor #f0daaaaa 2 40 20 10 20 degree
fi
draw quad lineColor #00ffffaa 2 20 20 5 20 degree
```

- c) Unsere Programme sind bisher nicht sonderlich spannend. Aus diesem Grund wollen wir ein neues Konzept einführen, welches Schleifen und Funktionen ersetzen soll: Sprungmarken. Sprungmarken kennen Sie eventuell bereits aus der *technischen Informatik*: Sie werden verwendet, um Zeilen im Code zu markieren, an die später zurückgekehrt werden soll. Eine Sprungmarke wird wie folgt definiert:

4

<name>:

Verwenden Sie das folgende Programm, um Ihre Implementierung zu testen:

```
var lineColor = #000000ff
var bool = 1 < 2
var degree = 0
if bool
then
draw circle lineColor #00ffffaa 2 30 20 10 20 degree
fi
Fleepowder:
var degree = degree + 90
if degree < 180
then
draw circle lineColor #f0daaaaa 2 40 20 40 40 degree
else
draw triangle lineColor #f0daaaaa 2 40 20 10 20 degree
fi
draw quad lineColor #00ffffaa 2 20 20 5 20 degree
```

- d) Zu guter Letzt müssen wir natürlich noch einen Befehl einführen, damit die Sprungmarke auch verwendet werden kann. Implementieren Sie hierfür den unbedingten Sprung (`jmp`). Der Befehl soll wie folgt aussehen und springt direkt zur Sprungmarke <name>:

4

jmp <name>

Verwenden Sie das folgende Programm, um Ihre Implementierung zu testen:

```
var lineColor = #000000ff
var bool = 1 < 2
var degree = 0
if bool
then
draw circle lineColor #00ffffaa 2 30 20 10 20 degree
fi
Fleepowder:
var degree = degree + 90
if degree < 180
then
draw circle lineColor #f0daaaaa 2 40 20 40 40 degree
else
draw triangle lineColor #f0daaaaa 2 40 20 10 20 degree
fi
draw quad lineColor #00ffffaa 2 20 20 5 * 2 20 degree
if degree < 180
then
jmp Fleepowder
fi
```

Aufgabe 2: Vipera Evanesca!

| |
|----------|
| 5 Punkte |
|----------|

Im Ilias finden Sie in unserem Aufgabenblock eine Umfrage. Diese Umfrage ist für das Bestehen unseres Blockes verpflichtend. Achten Sie daher bitte darauf, dass Sie die Umfrage vor dem Ende des zweiten Tages unseres Blockes ausfüllen.

Hinweis: Sie erhalten die vollen Punkte lediglich, wenn Sie die Umfrage komplett ausgefüllt haben. Sie erhalten keine anteiligen Punkte.