

Aufgabe 3 (Syntax und Semantik):
(5 + 1.5 + 1.5 = 8 Punkte)

- a) Die Menge der syntaktisch korrekten **SASP** Programme wird durch die Grammatik $G_2 = (\{A, B, S_2\}, \{., :-, p, q, r, s\}, P_2, S_2)$ definiert, wobei P_2 genau die folgenden Produktionsregeln enthält:

$$\begin{aligned} S_2 &\rightarrow A. \\ S_2 &\rightarrow A. S_2 \\ A &\rightarrow B \\ A &\rightarrow B :- B \\ B &\rightarrow p \\ B &\rightarrow q \\ B &\rightarrow r \\ B &\rightarrow s \end{aligned}$$

Die Semantik $\mathcal{W}(\mathcal{P})$ eines syntaktisch korrekten **SASP** Programms \mathcal{P} ist wie folgt definiert, wobei \mathcal{P}' ebenfalls ein syntaktisch korrektes **SASP** Programm ist und $x, y \in \{p, q, r, s\}$:

$$\begin{aligned} \mathcal{W}(x.) &= \{x\} \\ \mathcal{W}(x :- y.) &= \emptyset \\ \mathcal{W}(\mathcal{P}' x.) &= \mathcal{W}(\mathcal{P}') \cup \{x\} \\ \mathcal{W}(\mathcal{P}' x :- y.) &= \begin{cases} \mathcal{W}(\mathcal{P}') \cup \{x\} & \text{falls } y \in \mathcal{W}(\mathcal{P}') \\ \mathcal{W}(\mathcal{P}') & \text{sonst} \end{cases} \end{aligned}$$

Für alle **SASP** Programme \mathcal{P} gilt also $\mathcal{W}(\mathcal{P}) \subseteq \{p, q, r, s\}$.

Geben Sie für die folgenden Ausdrücke an, ob es sich um ein syntaktisch korrektes **SASP** Programm handelt und welche Semantik es hat.

- i) $p. q. r :- p$ ii) $r. q :- p :- r.$ iii) $r. q :- s. s. p :- s.$ iv) $t :- s.$

- b) Beweisen oder widerlegen Sie: Zwei Ausdrücke einer Sprache mit unterschiedlicher Syntax haben auch immer eine unterschiedliche Semantik.
- c) Beweisen oder widerlegen Sie: Zwei Ausdrücke einer Sprache mit unterschiedlicher Semantik haben auch immer eine unterschiedliche Syntax.

Aufgabe 5 (Formale Sprachen und Grammatiken): (3.5 + 2 + 2 = 7.5 Punkte)

Gegeben sei die folgende Sprache:

$$L_2 = \{w \in \{a, b\}^* \mid \text{auf jedes } a \text{ folgen direkt mindestens zwei } b \text{ oder auf jedes } b \text{ folgt direkt mindestens ein } a\}$$

Die folgenden Wörter sind beispielsweise in der Sprache enthalten:

b $abbbbabb$ $aaaabaaaaba$ ε

Folgende Wörter sind nicht Bestandteil der Sprache:

bab $bbbbbbba$ $ababab$ $abbaba$

- Geben Sie eine kontextfreie Grammatik an, welche die Sprache L_2 erzeugt.
- Geben Sie eine Grammatik in EBNF an, die L_2 definiert. Ihre Grammatik darf nur aus einer Regel bestehen und diese Regel darf nicht rekursiv sein (d. h. das Nichtterminalsymbol auf der linken Seite darf rechts nicht auftreten).
Um die Lesbarkeit zu erhöhen, dürfen Sie Anführungszeichen um Terminalsymbole weglassen.
- Geben Sie ein Syntaxdiagramm ohne Nichtterminalsymbole an, das die Sprache L_2 definiert.

Aufgabe 7 (Zweierkomplement):
(5 + 6 = 11 Punkte)

- a) Welche Zahlen repräsentieren die folgenden Bitfolgen im 10-Bit Zweierkomplement?

0011001100 1001001001 1000000000 1110101101 0010110100

- b) Die zwei folgenden Java-Ausdrücke werten jeweils zu `true` aus. Geben Sie dafür jeweils eine Begründung.

1) `2_000_000_000 + 1_000_000_000 + 1_294_967_296 == 0`

2) `-(2_000_000_000 + 147_000_000 + 483_000 + 648) < 0`

Hinweise:

- In Java dürfen in einer Zahl beliebig viele `_` zwischen den Ziffern auftreten. Dies verändert den Wert der Zahl nicht, sondern es wird nur verwendet, um die Lesbarkeit der Zahl zu erhöhen. `2_000_000_000` ist also nur eine andere Schreibweise für die Zahl 2000000000.

Aufgabe 9 (Casting):

(12 Punkte)

Bestimmen Sie den Typ und das Ergebnis der folgenden Java-Ausdrücke und begründen Sie Ihre Antwort. Sollte der Ausdruck nicht typkorrekt sein, begründen Sie, worin der Fehler besteht.

Dabei seien die Variablen `x`, `y` und `z` wie folgt deklariert: `int x = 1; int y = 121; int z = 126;`

- a) `2147483647 + x`
- b) `2147483647 + 'x'`
- c) `2147483647 + "x"`
- d) `(byte) (3 * z) == 'z' && true`
- e) `2147483648 + 3f`
- f) `(byte) 256 * 2147483648L`
- g) `x + "y" - z`
- h) `y != 'y' ? 2.0 : 1`

Aufgabe 11 (Programmierung):

(11.5 Punkte)

Implementieren Sie einen einfachen Zinseszins-Rechner in Java. Das Programm soll den Benutzer zunächst nach dem Startbetrag und dem Zinssatz in Prozent fragen. Beide Werte sollen als `double` eingelesen werden. Anschließend soll der Benutzer gefragt werden, was er berechnen möchte. Dabei stehen ein *Ziel* (das die Dauer berechnet, bis ein Zielbetrag erreicht wird) oder die *Zeit* (die den nach einer gewissen Zeit angesparten Betrag berechnet) zur Auswahl. Der Benutzer soll also entweder *Ziel* oder *Zeit* als String eingeben. Gibt der Benutzer keines von beidem ein, soll Ihr Programm eine Fehlermeldung anzeigen und sich beenden.

Ziel Bei dieser Auswahl soll der Benutzer nach einem Zielbetrag als `double` gefragt werden. Anschließend soll das Programm *mit Hilfe einer geeigneten Schleife* berechnen, wie viele Jahre es braucht, bis der Zielbetrag erreicht oder überschritten wurde.

Das Programm soll dann eine Ausgabe der Form "Es dauert 2 Jahre bei einem Zinssatz von 5%, um von 100 auf den Betrag 110 zu sparen. Nach dieser Zeit hat man 110.25." liefern. Die Werte in diesem Beispiel sollen vom Programm natürlich durch die Benutzereingaben und die berechneten Werte ersetzt werden.

Zeit Bei dieser Auswahl soll das Programm den Benutzer nach einer Anzahl von Jahren als `int`-Wert fragen. Anschließend berechnet das Programm *mit einer geeigneten Schleife* wie viel man nach dieser Zeit gespart hat. Die Werte werden dann in der Form "Bei einem Zinssatz von 5% und einem Startbetrag von 100 hat man nach 2 Jahren 110.25 gespart." ausgegeben.

Ein Beispiellauf des Programms könnte also so aussehen:

```
Bitte geben Sie den Startbetrag ein.
100
Bitte geben Sie den Zinssatz als Prozentwert ein.
5
Bitte waehlen Sie aus:
Ziel: Berechnet die Zeit, bis ein gegebener Betrag angespart wurde.
Zeit: Berechnet den Betrag, der nach einer gegebenen Zeit angespart wurde.
Ziel
Bitte geben Sie den Zielbetrag ein.
110
Es dauert 2 Jahre bei einem Zinssatz von 5%, um von 100 auf den Betrag 110
zu sparen. Nach dieser Zeit hat man 110.25.
```

Hinweise:

- Um zwei Strings `str1` und `str2` auf Gleichheit zu testen, verwenden Sie `str1.equals(str2)` (und nicht `str1 == str2`).
- Beim Zinseszins wird jedes Jahr der aktuelle Betrag um den Zinssatz erhöht. Im obigen Beispiel wird der Betrag im ersten Jahr um 5 ($= 5\% \cdot 100$) erhöht. Im zweiten Jahr dann um 5.25 ($= 5\% \cdot 105$). Im dritten Jahr würden dann 5.5125 ($= 5\% \cdot 110.25$) hinzu kommen.
- Verwenden Sie die Klasse `SimpleIO` zum Einlesen und Ausgeben von Werten. Legen Sie die bereitgestellte Datei `SimpleIO.java` einfach im gleichen Verzeichnis wie ihre Lösung ab. Dann findet Java diese automatisch.

Aufgabe 12 (Intro, Deck 0 und Deck 1):

(Codescape)

Schließen Sie das Intro und das Tutorial zum Spiel Codescape ab und lösen Sie die Missionen von Deck 0 und Deck 1.

Ihre Lösung für die Codescape Missionen wird nur dann für die Zulassung gezählt, wenn sie Ihre Lösung vor der einheitlichen Codescape Deadline am Samstag, den 16.01.2021, um 23:59 Uhr abschicken.