

Übung 06

Korrektur: Sebastian Pritz

Name: Selina Adlberger

Punkte: 66

Allgemeine Anmerkungen:

Warum wurde auf 3 verschiedene Solutions für jedes Projekt aufgespalten? -2

Beispiel 1: Arithmetische Ausdrücke (infix)

23 von 30

Lösungsidee: 7/7

Anmerkung	Abzug
Sehr coole Visualisierung mit der Skizze! 😊	

Quellcode: 9/13

Anmerkung	Abzug
Keine Testfunktionsdatei oder Testfunktionen (alles in Main) – in allen Teilen...	-2
Die geworfenen Fehler sollten auch irgendwo wieder abgefangen werden, und nicht zum Programmabsturz führen → Ausgabe bei den Testfällen	-2

Testfälle: 7/10

Anmerkung	Abzug
Keine Division durch 0	-2
Kein Test mit unerlaubtem Zeichen	-1

Beispiel 2: Arithmetische Ausdrücke (präfix)

19 von 30

Lösungsidee: 5/7

Anmerkung	Abzug
Bei Präfix-Ausdrücken wäre noch relevant zu erwähnen, dass keine negativen Zahlen erlaubt sind, da man nicht vom Minus Operator unterscheiden kann. (dafür müsste man im String „nach vorne schauen“).	-2

Quellcode: 8/13

Anmerkung	Abzug
Bei dieser Implementierung ist es aufgrund des Schleifenaufbaus nicht immer möglich das korrekte Ergebnis zu erzielen. Z.B mehrere Operatoren hintereinander ($/ * 4 5 5 \rightarrow$ wäre in Infix $„(4 * 5) / 5“ = 4$, wirft aber Exception) bzw. einzelne Zahlen einsetzen ($„2“ \rightarrow 0$).	-3
Division durch 0 sollte Exception auslösen, und nicht „inf“ ausgeben	-2

Testfälle: 6/10

Anmerkung	Abzug
Ist $2 / (2+2)$ wirklich 3? Und $4 / (4/1) = 1$? Wurde einfach so stehen gelassen, ist aber definitiv inkorrekt.	-1
Komplexe Rechnungen nur theoretisch	-1
Fehlermeldungen nicht provoziert bzw. nur theoretisch:	-2
<ul style="list-style-type: none">Unknown character \rightarrow SonderzeichenError parsing factor \rightarrow zu viele RechenzeichenFalsche Syntax (Infix Notation probieren)	

Lösungsidee: 7/9

Anmerkung	Abzug
Es wäre relevant zu wissen, wo das Parsen der Variablen passiert (Grammatik).	-2

Quellcode: 13/18

Anmerkung	Abzug
<p>Programm wirft bei Verwendung von Variablen sofort Exceptions, egal ob links oder rechts vom Operand, selbst bei simplen Rechnungen → Fehler ist, dass die Methoden von m_scanner nicht „is_string()“ lautet, sondern „is_identifier()“ → gleichnamig zur Methode! Natürlich auch beim Parsen. Zusätzlich dazu, wurde „initialize()“ nirgends aufgerufen. Ansonsten funktioniert das Verwenden von Variablen aber.</p> <p>Abzüge für: Verwendung der falschen Methode, vergessen von initialize, kein Behandeln von Verwendung nicht existenter Variablen und zu guter letzt: Ohne mein Eingreifen hätte dieses Beispiel garnicht funktioniert.</p> <p>Ansonsten stehen die Parsing-Methoden aber an der richtigen Stelle und der Aufbau ist ansonsten korrekt.</p>	-5

Testfälle: 6/13

Anmerkung	Abzug
Testfälle passen, aber halt nur theoretisch. Letzter Test zudem etwas spärlich ausformuliert.	-7