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**  Sehr coole Visualisierung mit der Skizze! 😊 | **Abzug** |

Quellcode: 9/13

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

Testfälle: 7/10

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

# Beispiel 2: Arithmetische Ausdrücke (präfix) 19 von 30 Lösungsidee: 5/7

|  |  |
| --- | --- |
| **Anmerkung**  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“). | **Abzug**  -2 |

Quellcode: 8/13

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

Testfälle: 6/10

|  |  |
| --- | --- |
| **Anmerkung**  Ist 2 / (2+2) wirklich 3? Und 4 / (4/1) = 1? Wurde einfach so stehen gelassen, ist aber definitiv inkorrekt.  Komplexe Rechnungen nur theoretisch  Fehlermeldungen nicht provoziert bzw. nur theoretisch:   * Unknown character 🡪 Sonderzeichen * Error parsing factor 🡪 zu viele Rechenzeichen * Falsche Syntax (Infix Notation probieren) | **Abzug**  **-1**  **-1**  **-2** |

# Beispiel 3: Rechnen mit Variablen 26 von 40

Lösungsidee: 7/9

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

Quellcode: 13/18

|  |  |
| --- | --- |
| **Anmerkung**  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.  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.  Ansonsten stehen die Parsing-Methoden aber an der richtigen Stelle und der Aufbau ist ansonsten korrekt. | **Abzug**  **-5** |

Testfälle: 6/13

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