

SWE3 Übung 06

1 INHALTSVERZEICHNIS

2	Beispiel 1: Infix Parser	2
2.1	Lösungsidee	2
2.2	Code.....	2
2.3	Testfälle	2
3	Beispiel 2: Präfix Parser	3
3.1	Lösungsidee	3
3.2	Code.....	3
3.3	Testfälle	4
4	Beispiel 3: Infix Parser mit Variablen.....	6
4.1	Lösungsidee	6
4.2	Code.....	6
4.3	Testfälle	6

Zeitaufwand: 5h

2 BEISPIEL 1: INFIX PARSER

2.1 LÖSUNGSDIEE

Das Ziel dieser Aufgabe war es einen Parser zu implementieren, der den `pfc::scanner` verwendet um arithmetische Ausdrücke in Infix-Notation von einem Input-Stream oder einem String zu lesen und zu interpretieren. Dafür wurde folgende Grammatik verwendet:

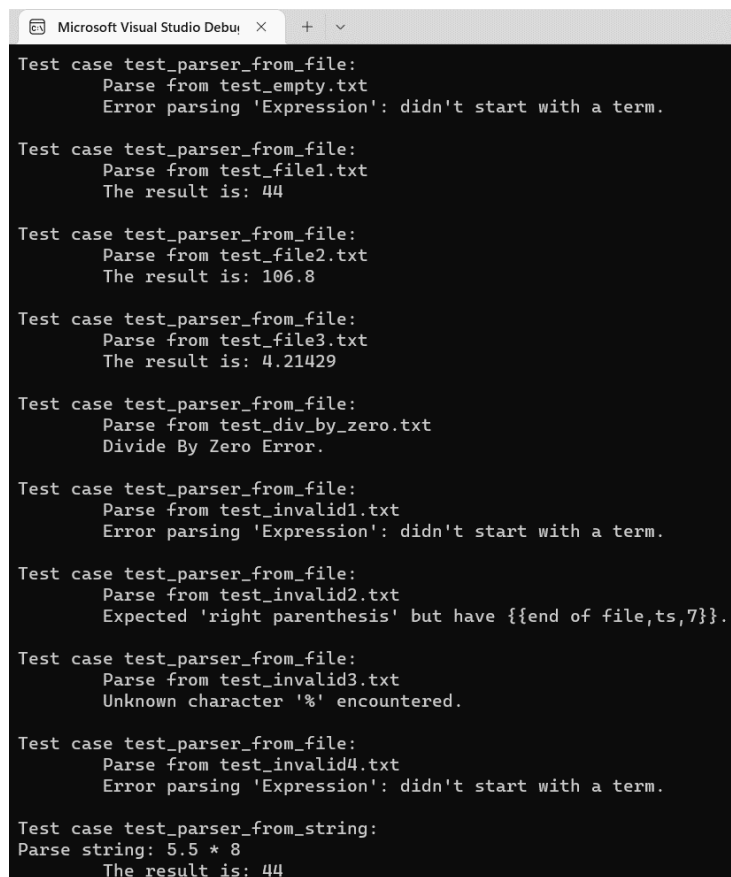
Grammatik der arithmetischen Ausdrücke in EBNF-Schreibweise:

```
Expression = Term { AddOp Term } .
Term       = Factor { MultOp Factor } .
Factor     = [ AddOp ] ( Unsigned | PExpression ) .
Unsigned   = Digit { Digit } .
PExpression = „ ( “ Expression „ ) “ .
AddOp      = „+“ | „-“ .
MultOp     = „.“ | „/“ .
Digit      = „0“ | „1“ | „2“ | „3“ | „4“ | „5“ | „6“ | „7“ | „8“ | „9“ .
```

2.2 CODE

siehe SWE3_Huspek_Ue06

2.3 TESTFÄLLE



```
Microsoft Visual Studio Debug Console
Test case test_parser_from_file:
  Parse from test_empty.txt
  Error parsing 'Expression': didn't start with a term.

Test case test_parser_from_file:
  Parse from test_file1.txt
  The result is: 44

Test case test_parser_from_file:
  Parse from test_file2.txt
  The result is: 106.8

Test case test_parser_from_file:
  Parse from test_file3.txt
  The result is: 4.21429

Test case test_parser_from_file:
  Parse from test_div_by_zero.txt
  Divide By Zero Error.

Test case test_parser_from_file:
  Parse from test_invalid1.txt
  Error parsing 'Expression': didn't start with a term.

Test case test_parser_from_file:
  Parse from test_invalid2.txt
  Expected 'right parenthesis' but have {{end of file,ts,7}}.

Test case test_parser_from_file:
  Parse from test_invalid3.txt
  Unknown character '%' encountered.

Test case test_parser_from_file:
  Parse from test_invalid4.txt
  Error parsing 'Expression': didn't start with a term.

Test case test_parser_from_string:
  Parse string: 5.5 * 8
  The result is: 44
```

```

Microsoft Visual Studio Debu
Test case test_parser_from_string:
Parse string:
    Error parsing 'Expression': didn't start with a term.

Test case test_parser_from_string:
Parse string: ( 13 + ( 4 * 5.5) - 10) / (6 - 6)
    Divide By Zero Error.

Test case test_parser_from_string:
Parse string: 8*16+(4.8-3)-23
    The result is: 106.8

Test case test_parser_from_string:
Parse string: 10 - 3 + (12.5 - 4 + (3 * -5)) / -7 * -3
    The result is: 4.21429

Test case test_parser_from_string:
Parse string: (8 + 4 ( -3 * 2)
    Expected 'right parenthesis' but have {{left parenthesis,ts,9}}.

Test case test_parser_from_string:
Parse string: 5 * (6 % 3)
    Unknown character '%' encountered.

Test case test_parser_from_string:
Parse string: / 8 + 3 4
    Error parsing 'Expression': didn't start with a term.

Test case test_parser_from_string:
Parse string: 12.4 - 8.2 + (8
    Expected 'right parenthesis' but have {{end of file,ts,7}}.

```

3 BEISPIEL 2: PRÄFIX PARSER

3.1 LÖSUNGSDIEE

Das Ziel dieser Aufgabe war es, den Infix Parser in einen Präfix Parser umzuwandeln. Dieser Präfix Parser kann im Gegensatz zum Infix Parser arithmetische Ausdrücke in der Präfix Notation interpretieren und berechnen. Dabei steht der Operator (+, -, *, /) immer zuerst und danach die beiden Operanden (Zahl oder wieder Expression).

Grammatik für den Präfix-Scanner in EBNF-Schreibweise:

Expression = Operation Term Term

Term = (Unsigned | Expression)

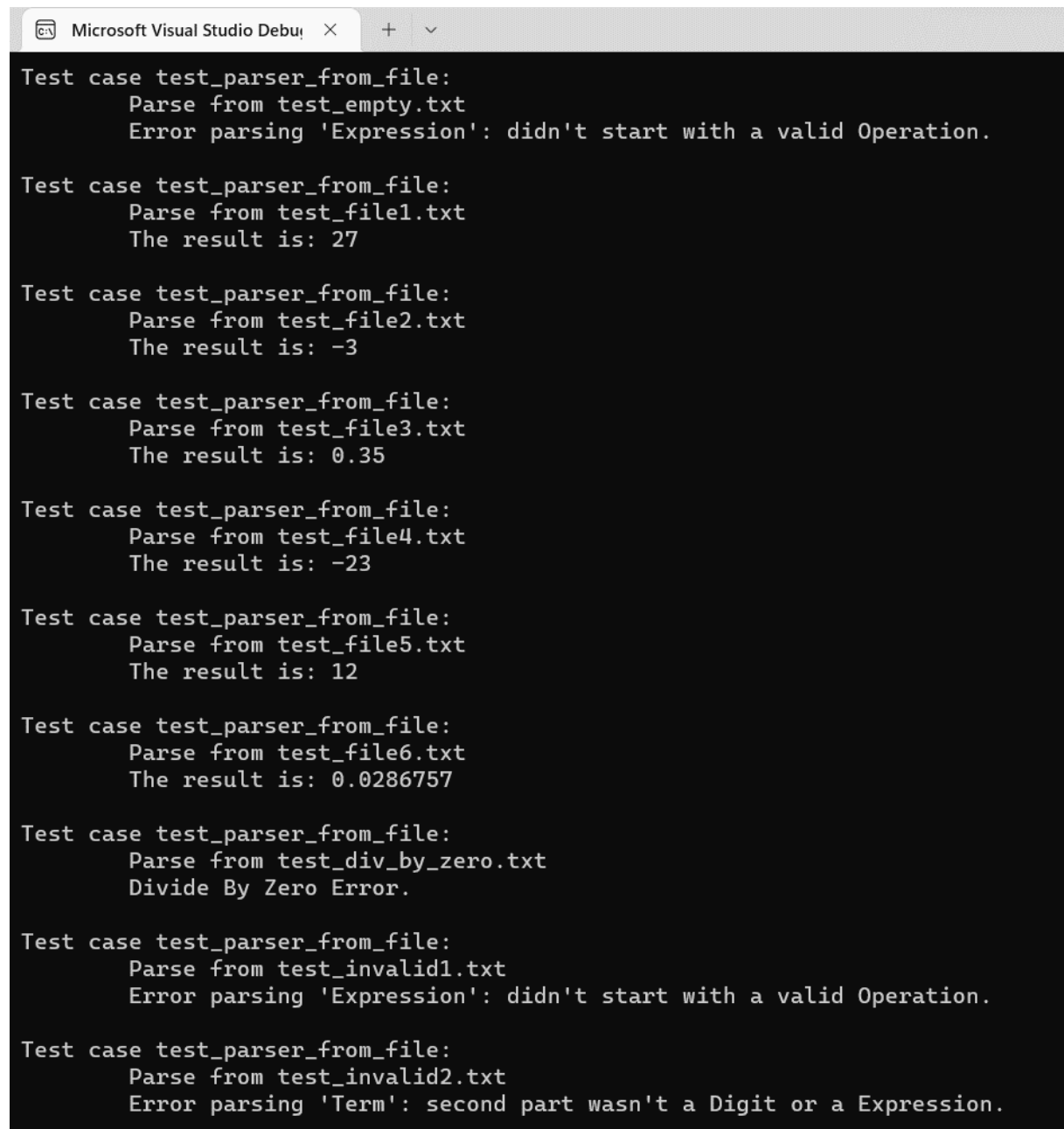
Operation = „+“ | „-“ | „*“ | „/“.

Unsigned = Digit {Digit}.

3.2 CODE

Siehe SWE3_Huspek_Ue06

3.3 TESTFÄLLE



```
Microsoft Visual Studio Debug Console
Test case test_parser_from_file:
  Parse from test_empty.txt
  Error parsing 'Expression': didn't start with a valid Operation.

Test case test_parser_from_file:
  Parse from test_file1.txt
  The result is: 27

Test case test_parser_from_file:
  Parse from test_file2.txt
  The result is: -3

Test case test_parser_from_file:
  Parse from test_file3.txt
  The result is: 0.35

Test case test_parser_from_file:
  Parse from test_file4.txt
  The result is: -23

Test case test_parser_from_file:
  Parse from test_file5.txt
  The result is: 12

Test case test_parser_from_file:
  Parse from test_file6.txt
  The result is: 0.0286757

Test case test_parser_from_file:
  Parse from test_div_by_zero.txt
  Divide By Zero Error.

Test case test_parser_from_file:
  Parse from test_invalid1.txt
  Error parsing 'Expression': didn't start with a valid Operation.

Test case test_parser_from_file:
  Parse from test_invalid2.txt
  Error parsing 'Term': second part wasn't a Digit or a Expression.
```

```
Microsoft Visual Studio Debu  X  +  v

Test case test_parser_from_file:
  Parse from test_invalid3.txt
  Error parsing 'Term': second part wasn't a Digit or a Expression.

Test case test_parser_from_file:
  Parse from test_invalid4.txt
  Error parsing 'Expression': didn't start with a valid Operation.

Test case test_parser_from_string:
Parse string: + 4 23
  The result is: 27

Test case test_parser_from_string:
Parse string:
  Error parsing 'Expression': didn't start with a valid Operation.

Test case test_parser_from_string:
Parse string: * - 5 6 3
  The result is: -3

Test case test_parser_from_string:
Parse string: / - 9 2 * 4 5
  The result is: 0.35

Test case test_parser_from_string:
Parse string: - 7 * 5 6 8
  The result is: -23

Test case test_parser_from_string:
Parse string: * 3 4 - 5 6
  The result is: 12

Test case test_parser_from_string:
Parse string: *-456
  Error parsing 'Term': second part wasn't a Digit or a Expression.

Test case test_parser_from_string:
Parse string: + ( 8 9 - 16 4
  Error parsing 'Term': second part wasn't a Digit or a Expression.

Test case test_parser_from_string:
Parse string: 3 4 +
  Error parsing 'Expression': didn't start with a valid Operation.

Test case test_parser_from_string:
Parse string: / * 3 6 - 5 5
  Divide By Zero Error.

Test case test_parser_from_string:
Parse string: /11+*17.3 22 3
  The result is: 0.0286757

Test case test_parser_from_string:
Parse string: ++1 2.7 * 3 - 5 4
  The result is: 6.7
```

4 BEISPIEL 3: INFIX PARSER MIT VARIABLEN

4.1 LÖSUNGSDIEE

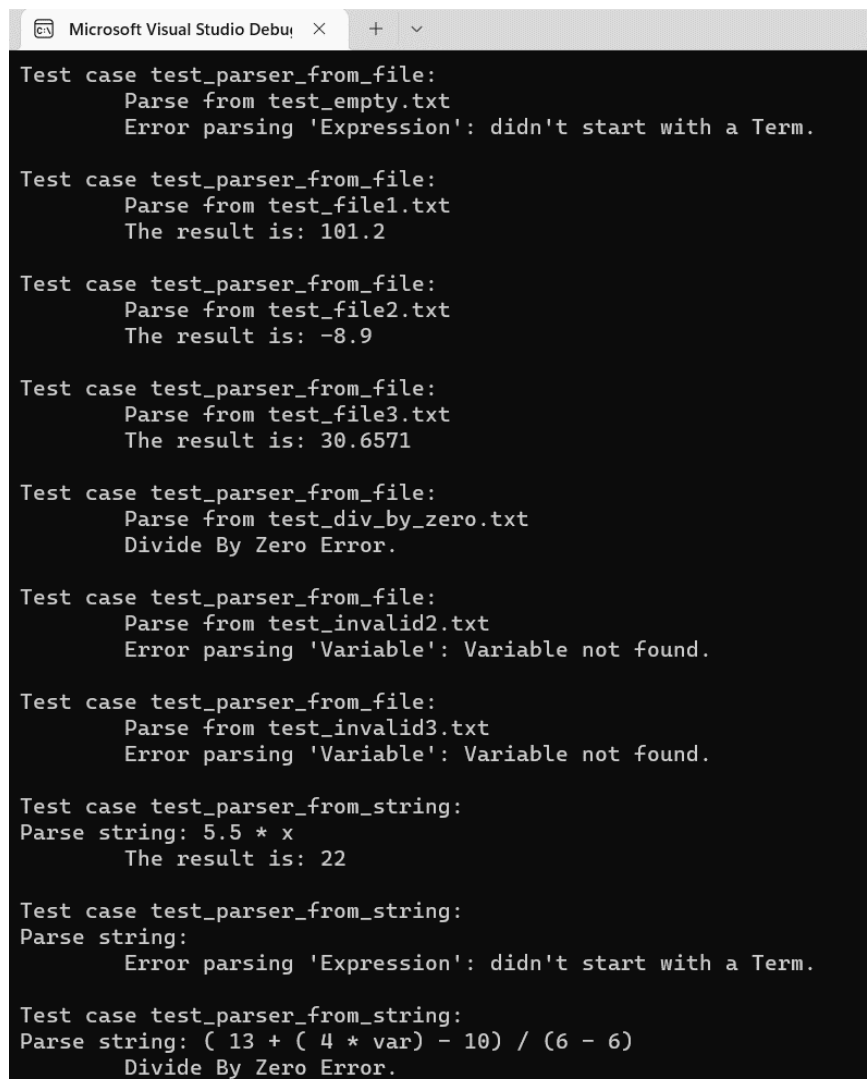
Das Ziel dieser Aufgabe war es den Parser für arithmetische Ausdrücke in Infix-Notation vom ersten Beispiel zu erweitern. Die Erweiterung sollte beinhalten, dass anstatt ‚Unsigned‘ auch zuvor festgelegte Variablen eingesetzt werden dürfen. Diese Variablen werden als Schlüssel-Werte-Paar gespeichert und dem Parser mitübergeben. Wird ein Identifier (Name einer Variable) eingelesen, wird überprüft, ob diese im Variablen-Speicher enthalten ist. Ist dies der Fall, wird ihr Wert geholt und für die Berechnung herangezogen. Befindet sich der eingelesene Identifier nicht im Variablen-Speicher, dann wird eine Exception geworfen.

4.2 CODE

Siehe SWE3_Huspek_Ue06

4.3 TESTFÄLLE

```
std::map<std::string, double> vars = { {"x", 4},  
                                       {"test", -8.9},  
                                       {"i", 22.9},  
                                       {"var", 18.4} };
```



```
Microsoft Visual Studio Debug Console  
Test case test_parser_from_file:  
  Parse from test_empty.txt  
  Error parsing 'Expression': didn't start with a Term.  
  
Test case test_parser_from_file:  
  Parse from test_file1.txt  
  The result is: 101.2  
  
Test case test_parser_from_file:  
  Parse from test_file2.txt  
  The result is: -8.9  
  
Test case test_parser_from_file:  
  Parse from test_file3.txt  
  The result is: 30.6571  
  
Test case test_parser_from_file:  
  Parse from test_div_by_zero.txt  
  Divide By Zero Error.  
  
Test case test_parser_from_file:  
  Parse from test_invalid2.txt  
  Error parsing 'Variable': Variable not found.  
  
Test case test_parser_from_file:  
  Parse from test_invalid3.txt  
  Error parsing 'Variable': Variable not found.  
  
Test case test_parser_from_string:  
  Parse string: 5.5 * x  
  The result is: 22  
  
Test case test_parser_from_string:  
  Parse string:  
  Error parsing 'Expression': didn't start with a Term.  
  
Test case test_parser_from_string:  
  Parse string: ( 13 + ( 4 * var) - 10) / (6 - 6)  
  Divide By Zero Error.
```

```
Test case test_parser_from_string:
Parse string: 8*x+(4.8-3)-i
    The result is: 10.9

Test case test_parser_from_string:
Parse string: 10 - 3 + (12.5 - test + (3 * -5)) / -7 * -3
    The result is: 9.74286

Test case test_parser_from_string:
Parse string: (8 + ( -3 * z))
    Error parsing 'Variable': Variable not found.

Test case test_parser_from_string:
Parse string: x
    The result is: 4

Test case test_parser_from_string:
Parse string: 6 + i * y
    Error parsing 'Variable': Variable not found.
```