

# Arithmetische Ausdrücke in Rust - Dokumentation

Mario Occhinegro (74661)  
Michael Kirmizakis (75592)  
Jonas Heck (67889)

## Inhaltsverzeichnis

|          |                                   |          |
|----------|-----------------------------------|----------|
| <b>1</b> | <b>Datatype</b>                   | <b>1</b> |
| <b>2</b> | <b>Expression Enum</b>            | <b>1</b> |
| <b>3</b> | <b>Evaluieren von Expressions</b> | <b>1</b> |
| <b>4</b> | <b>Parsen von Expressions</b>     | <b>1</b> |
| 4.1      | Parsen von Tokens . . . . .       | 1        |
| <b>5</b> | <b>Expression Typcheck</b>        | <b>1</b> |

## 1 Datatype

Einfaches Enum zur Bestimmung der Art des Datentyps. Wir haben Integer oder boolsche Werte

## 2 Expression Enum

Das Expression Enum beschreibt alle möglichen Ausdrücke, die wir verarbeiten können. Eine Variante jeweils für die Zahlen von 1-9. Sowie jeweils eine Variante für true und false. Für die Operatoren Plus, Mult, Or und And gibt auch jeweils eine Variante. Interessant ist hier, dass diese hier wieder Unterexpressionen verwalten. Ein mal für Rechts und ein mal für Links.

## 3 Evaluieren von Expressions

## 4 Parsen von Expressions

Die Funktion, die das Parsen von Expression zuständig ist funktioniert wie folgt. Sie bekommt einen String und ein Präzedenzlevel, um sich in der Rekursion zu merken was die Präzedenz der Parent Expression ist. Die Ausgabe ist dann ein Expressionenum, das den eingegebene Expression beschreibt. Innerhalb des Rumpfes bewegen wir uns von Links nach Rechts durch den übergebenen String. Jeder Aufruf fängt damit an, dass wir den Charakter auf dem wir uns befinden parsen. Dies ist immer eine Zahl, ein boolean oder ein geklammerter Ausdruck. Diese Expression merken wir uns als linken Teilbaum. Danach betrachten wir falls vorhanden, den nächsten Charakter, bei dem es sich in unserer Logik um einen Operator handeln muss, da wir keine unnären Operatoren unterstützen. Falls nicht vorhanden geben wir die linke Seite als Expression zurück und sind fertig. Für den Fall, dass es sich aber wirklich um einen Operator handelt, bestimmen wir zu aller erst dessen Präzedenz. Falls die Übergeordnete Präzedenz der Parent Expression höher ist, so brechen wir an dieser Stelle ab und geben den bis dato errechneten Baum zurück. Das ist notwendig, da falls die Präzedenz des übergeordneten Operator größer ist, der momentane Charakter teil der Parent-Expression ist. Falls für den anderen Fall setzen wir den Operator in die Wurzel unseres Baum.

## 5 Expression Typcheck

Diese Methode prüft den Datentyp eines Ausdrucks und gibt ein Ergebnis zurück. implementiert die methode typecheck für den expressiontyp. wenn der ausdruck eine zahl zwischen 0 und 9 ist, wird der datentyp tint zurückgegeben. wenn der ausdruck etrue oder efalse ist, wird der datentyp tbool zurückgegeben. wenn der ausdruck eine addition, multiplikation, logische oder oder logische undoperation ist, werden die datentypen der linken und rechten operanden überprüft. Wenn beide Operanden den richtigen Typ haben, wird TInt bzw. TBool zurückgegeben, andernfalls wird None zurückgegeben.