

EINFACHER PLOTTER, TEIL 2: ABSTRACT SYNTAX TREE

In diesem Aufgabenteil geht es darum, aus einer Funktion, die in Form eines Abstract Syntax Tree (dt. abstrakter Syntaxbaum) gegeben ist, einen Plot zu erstellen. Dazu muss verstanden werden, wie eine Funktion in dieser Form aussieht und wie sie sich verhält.

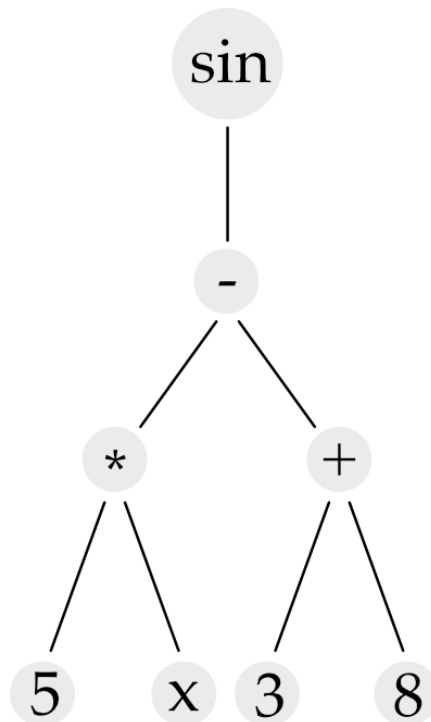
Falls du irgendwo Probleme hast, zögere bitte nicht, um Hilfe zu fragen! Wenn du irgendwo hängst, helfen wir dir gerne weiter! Melde dich bei einem der zuständigen Betreuer*innen: Lorenz Braun, Fabian Sand, Robin Fritz, Furkan Keserci, Julia Kisela. Wir sind über Discord, Mattermost und E-Mail erreichbar. Wen könnt ihr für Hilfe ansprechen? (E-Mail, Mattermost, Discord)

- Robin Fritz: frro1020@h-ka.de, @robin.fritz, Snobo#4754
- Julia Kisela: kiju1012@h-ka.de, @julia.kisela, lohikäärme#3336
- Furkan Keserci: kefu1011@h-ka.de, @furkan, Killnexx#9999
- Lorenz Braun: brlo1014@h-ka.de, @lorenzbraun, MrRaptorious#7632
- Fabian Sand: safa1017@h-ka.de, @safa1017, Fabian S#7613

LERNMATERIAL

Ein abstrakter Syntaxbaum stellt in unserem Fall evaluierbare Ausdrücke dar. Diese heißen Expressions. $3 + 5$ ist z. B. eine Expression und kann zu einer 8 ausgewertet werden. Genauer besteht sie aus einer NumberExpression (der 3), einer zweiten NumberExpression (der 5) und einer OperatorExpression (das + mit seinen beiden Argumenten links und rechts.)

Hier ein Beispiel, wie sich eine Funktion als Abstract Syntax Tree verhält: $f(x) = \sin(5 * x - (3 + 8))$



Hier ein Link, wo es noch mal einfach erklärt wird: <https://www.peterbeard.co/blog/post/syntax-trees-for-mathematical-expressions/>.

So würde das Beispiel in Code-Form aussehen:

```

Expression e = new UnaryFunctionExpression("sin",
    new OperatorExpression(OperatorType.MINUS,
        new OperatorExpression(OperatorType.TIMES,
            new NumberExpression(5),
            new XExpression(),
        new OperatorExpression(OperatorType.PLUS,
            new NumberExpression(3),
            new NumberExpression(8))
        )
    )
);
  
```

Listing 1: AST-Beispiel in Code-Form

VERSTÄNDNISFRAGEN

Bei der Abgabe müssen diese Fragen beantwortet werden können, mache dir also am besten Notizen.

- Schau dir das Beispiel in der Infix-Schreibweise an: $5 * x + (\log(x) - (3 + 5))$. Wie würdest du diesen Ausdruck mit der Hand ausrechnen?
- Zeichne einen Abstract Syntax Tree für den Beispielausdruck in der Infix-Schreibweise. Orientiere dich dabei an dem Beispiel. Du solltest auch erklären können, wie du darauf gekommen bist.
- Benenne die Expressions in deinem Syntaxbaum nach NumberExpression, OperatorExpression, UnaryFunctionExpression und XExpression. Was evaluieren die Teile jeweils?
- Werte deinen Syntaxbaum aus, wie wird er gelesen? In welcher Reihenfolge wird ausgewertet?
- Wie würde dein Beispielbaum als Code aussehen? (Schreibe es als Beispiel am Ende in deinen fertigen Code)

AUFGABENSTELLUNG

Wichtiger Hinweis: Im bereitgestellten Code finden sich Dokumentationskommentare, welche Aufschluss auf die Funktionsweise oder die zu implementierende Funktionalität geben. Bitte deshalb den Code und die Kommentare aufmerksam studieren! Bitte verändere gegebene Klassennamen und Methodennamen nicht, da diese für die Tests verwendet werden!

Um nun statt der hardgecodeten Funktionen den Syntaxbaum nutzen zu können, müssen neue Klassen implementiert werden, die von Expression erben.

Dazu findest du das Paket expressions im Ilias. Lade dieses und füge es in dein Projekt ein (in den src-Ordner hineinlegen). Die alte Klasse Expressions ist hier enthalten. Um deinen Code testen zu können, haben wir dir einen Testordner zur Verfügung gestellt. Lade diesen im Ilias herunter und füge ihn in dein Projekt ein (neben den src-Ordner legen). Die Tests benutzen das Framework JUnit, wie du dieses in Eclipse einrichtest, findest du in einem separaten Dokument in Ilias (Dokument JUnit Konfiguration). Bitte beachte, dass für deine erfolgreiche Abgabe alle Tests erfolgreich durchlaufen müssen (grüner Haken erscheint).

Es sollen die Tochter-Klassen XExpression, OperatorExpression, UnaryFunctionExpression und NumberExpression fertig implementiert werden. Jede dieser Klassen hat eine eval-Methode, die die jeweilige Expression auswertet sowie einen Konstruktor. Das Gerüst der Klassen ist schon gegeben, du musst aber noch die eval-Methode vervollständigen. Das enum OperatorType sagt dir, welche Operationen zu implementieren sind.

Hierbei ist zu beachten, dass die ausgewertete Expression rekursiv auszuwerten ist. (Wenn dir dieser Schritt nicht klar ist, schau noch einmal die Beispiele an). D.h. dass z. B. der Ausdruck $(\text{Expression1} + \text{Expression2}).\text{eval}$ zu $\text{Expression1}.\text{eval} + \text{Expression2}.\text{eval}$ ausgewertet wird. Damit wird die eval-Methode rekursiv durch den AST durchgereicht, genau wie auch im Beispiel.

In der Grid-Klasse muss nun an die auszuführende Stelle ein Syntaxbaum geschrieben werden. Ein Beispiel sieht so aus:

```
Expression e = new OperatorExpression(OperatorType.  
    TIMES, new NumberExpression(-1), new  
    UnaryFunctionExpression("exp", new XExpression()));
```

Füge bitte die Funktionen $\sin(x)$, $\log(5 * (3 + x))$, $x^2 + 3$ in der richtigen Schreibweise in deine Grid-Klasse ein.

CHECKLISTE

- eval-Methode in XExpression
- eval-Methode in OperatorExpression
- eval-Methode in UnaryFunctionExpression
- Exception in UnaryFunctionExpression
- eval-Methode in NumberExpression
- Beispiel aus Verständnisfragen ebenfalls geplottet
- Die zu zeichnenden Funktionen sind richtig im Plot abgebildet

---Pflichtaufgaben enden hier---

Falls dir die Aufgaben bisher sehr leicht fielen, gibt es hier noch eine Knobelaufgabe. (Keine Pflicht!)

Implementiere weitere Funktionalitäten in der Klasse Grid:

- Methode zum Zoomen mit dem Mausrad, passt auch Beschriftung an
- Methode zum Verschieben des Koordinatensystems mit der Maus, passt auch Beschriftung an
- Breitere Linien bei den geplotteten Funktionen
- Die Zoom-Methode verhält sich erwartungsgemäß, d. h. die Stelle, an der sich die Maus befindet, bleibt fest, der Rest skaliert ausgehend von dieser Stelle

Wenn du hier Probleme hast, aber es trotzdem gerne machen würdest oder vielleicht gar nicht weißt, wo du anfangen sollst, melde dich bei Furkan, z. B. über Matternmost oder den Discord-Server.