

**Einführung in die theoretische Informatik**  
Sommersemester 2018 – Hausaufgabenblatt Lösungsskizze 1

**Programmierabgabe**

Für die Korrektur Ihrer Programmierabgaben wird TUMjudge(<https://judge.in.tum.de/theo/public/>) verwendet. Auf dem ersten Hausaufgabenblatt werden die für Sie relevanten Funktionen des Webinterfaces erklärt. Weitere Details finden Sie auf den Folien des Praktikums Algorithms for Programming Contests von 2017(<https://www7.in.tum.de/um/courses/theo/ss2018/materials/judge.pdf>).

**AUFGABE 1.1.** (*TUMjudge verstehen*)

1 Punkte

In dieser Aufgabe lernen Sie, den TUMjudge zu verwenden.

- Gehen Sie auf die Website des TUMjudge (<https://judge.in.tum.de/theo/public/>).
- Registrieren Sie sich. Verwenden Sie dazu ihren Rechnerhallen login.
  - Wenn ihre Mailadresse name@in.tum.de ist, dann ist ihr login für den judge “name”
  - Das Passwort ist dasselbe wie beim Rechnerhallen login.
  - Sollten Sie Ihr Passwort oder Ihren login vergessen haben, gehen Sie zum Infopoint (<http://www.in.tum.de/en/current-students/services-facilities-it/infopoint.html>).
- Stellen Sie sicher, dass Sie links oben den TUMjudge Theo ausgewählt haben und loggen Sie sich ein.
- Wählen Sie unter “contests” den Punkt “Theo Homework 1”.
- Laden Sie für Problem A (KnowTheJudge) ein Java-Programm hoch, das den String “gg ez” (ohne Anführungszeichen) auf der Kommandozeile ausgibt.

**Hinweise:**

- Bei Problemen mit TUMjudge können Ihnen die Folien des Praktikums Algorithms for Programming Contests von 2017(<https://www7.in.tum.de/um/courses/theo/ss2018/materials/judge.pdf>) helfen.

**AUFGABE 1.2.** (*Typen von Grammatiken*)

2 Punkte

In dieser Aufgabe implementieren Sie einige Methoden, um zu prüfen, welchen Typ eine Grammatik hat.

- Laden Sie das Codegerüst für die erste Hausaufgabe auf <https://www7.in.tum.de/um/courses/theo/ss2018/homework/> herunter.
- Betrachten Sie die Klassen Production und Grammar. Sie sollten die Attribute, insbesondere die dafür gewählten Datentypen, verstehen, bevor Sie fortfahren.
- Implementieren Sie alle mit TODO markierten Methoden, also `isType0`, `isType1`, `isType2`, und `isType3`. Die Methoden sollen für eine gegebene Grammatik `true` zurück geben, wenn die Grammatik Typ  $n$  hat ( $n \in \{0, 1, 2, 3\}$ ), ansonsten `false`.
- Laden Sie dann für Problem B (IsTypeN) *alle* Dateien hoch, also inclusive Production und Grammar, obwohl sie an diesen nichts ändern müssen.