

**Einführung in die theoretische Informatik**  
Sommersemester 2019 – 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 Punkt

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*)

1 Punkt

In dieser Aufgabe implementieren Sie einige Methoden, um zu prüfen, welchen Typ eine Produktion hat. Mehr Informationen hierzu finden Sie in der Aufgabenstellung in TUMjudge (‘Problem text’).

- Lesen Sie sich die PDF-Angabe zu der Aufgabe ‘ProductionType’ durch (zu finden unter ‘course/problems’ auf TUMjudge)
- Laden Sie das Codegerüst für die erste Hausaufgabe auf [https://www21.in.tum.de/teaching/theo/SS19/materials/ha/01/templates\\_ha01.tar.gz](https://www21.in.tum.de/teaching/theo/SS19/materials/ha/01/templates_ha01.tar.gz) herunter.
- Betrachten Sie die Klasse `Production`. Sie sollten die Attribute, insbesondere die dafür gewählten Datentypen, verstehen, bevor Sie fortfahren.
- Implementieren Sie alle mit `TODO` markierten Methode, also `getType`. Die Methode soll für eine gegebene Produktion deren Typ als Zahl zwischen 0 und 3 zurückgeben.
- Natürlich sollen Sie den *genauesten* Typ einer Produktion zurückgeben, d.h. z.B. für eine Produktion von Typ 3 müssen Sie 3 zurückgeben, obwohl sie auch die Typen 0, 1, und 2 hat.
- Laden Sie dann für Problem B (ProductionType) ihre Datei `ProductionType.java` hoch.