



# Grundlagen der Informatik 2

## Praktikumsaufgabe 0

Abgabe: 10.05.2010, 12:00

**Hinweise.** Für die Praktikumsaufgaben benötigen Sie die Entwicklungsumgebung Eclipse und die in Eclipse integrierbare Profiling Plattform TPTP. Auf den Pool-Rechnern finden Sie beides bereits vorinstalliert. Hilfe erhalten Sie in den Poolsprechstunden Mo-Fr 10:00-12:00 Uhr und 13:00-15:00 Uhr von unseren Praktikurstutoren.

Um auf Ihrem eigenen System zu Hause oder am Notebook zu arbeiten:

- Richten Sie die Eclipse IDE for Java Developers ein. Erhältlich unter <http://www.eclipse.org/downloads/packages/release/galileo/sr2>.
- Installieren Sie TPTP von <http://www.eclipse.org/tptp/>. Folgen Sie den Hinweisen unter *Installation Guide*.

Import der Quelltext-Vorgaben:

- Laden Sie die Quelltext-Vorgaben für Praktikumsaufgabe 0 aus dem moodle-Portal herunter (Project0.zip).
- Importieren Sie das Projekt in Eclipse. *Hinweis:* File, Import, General, Existing Projects Into Workspace, Select Archive File, Browse, Project0.zip auswählen, Finish.

**Zu implementieren.** Als Beispiel für Laufzeiten wurde in der Vorlesung die Methode `contains` für Integer-Arrays definiert (Foliensatz 3, Seite 11). Eine analoge Implementierung dieser Methode für Listen von Strings ist in `Contains.java` die Methode `contains_lecture`.

1. Eine einfache Optimierung dieser Methode ist, die Suche abubrechen sobald das gesuchte Element gefunden wurde. Implementieren Sie eine solche optimierte Version in der Methode `contains_fast` in derselben Datei! Die zu bearbeitenden Stellen im Code sind mit **TODO:** markiert.
2. Testen Sie Ihre Lösung mit den vorgegebenen Testfällen. *Hinweis:* Öffnen Sie `ContainsTest.java` im Editor. Wählen Sie im Menü: *Run, Run As, JUnit Test*.

3. Die `main`-Methode in `Contains.java` führt 5000 Aufrufe der verschiedenen `contains`-Implementierungen mit zufälligen Strings aus der Liste aus. Starten Sie `main` mit dem Profiler. Im Menü: *Run, Profile As, Java Application*. Wählen Sie im Tab *Monitor* nur den Punkt *Execution Time Analysis* aus und klicken sie auf *Profile*. Wechseln Sie in die Profiling Perspektive, warten Sie bis das Profiling abgeschlossen ist (einige Sekunden bis Minuten) und betrachten Sie anschließend den Report zur *Execution Time Analysis*. Ein Glossar zu den verschiedenen Begriffen in TPTP finden Sie unter <http://www.eclipse.org/tptp/home/documents/resources/glossary.xml>
4. Beantworten Sie nun folgende Fragen zu den Profiling-Ergebnissen im Kopf von `Contains.java` an den mit **TODO:** markierten Stellen:
  - a) Wie verhält sich die gemessene Laufzeit von den Methoden `contains_fast` und `contains_lecture` zu einander? Erklären Sie das Ergebnis!
  - b) Die Methode `contains_java` ruft die in der Java-API vordefinierte `contains`-Funktion für Listen auf. Wie verhält sich die Laufzeit von `contains_java` zu den anderen? Was könnte der Grund sein?

**Abgabe.** Reichen Sie Ihre fertig bearbeitete Datei `Contains.java` bis zum Abgabetermin in moodle ein. Die Abgabe ist in Gruppen von bis zu drei Personen möglich. **Ein** Mitglied der Gruppe reicht die Lösung zusammen mit einem Kommentar der folgenden Form ein: *Die Übung wurde bearbeitet von der Gruppe X, Y, Z; die Abgabe erfolgte durch X.* **Jedes andere** Mitglied der Gruppe reicht *anstelle* der Lösung nur diesen (identischen) Kommentar ein.

Bitte stellen Sie sicher, dass Sie und ggf. Ihre Gruppenmitglieder diese Regeln befolgen; wird weder eine Lösung noch der obenstehende Kommentar eingereicht, so vergibt der Tutor auch keine Punkte!