

2. Praktikum

Java und Eclipse

Bearbeitungszeitraum: **16.11. - 30.11.20, 9.15 Uhr**

Gesamtpunktzahl: **5 Abgabepunkte + 6 Leistungspunkte**

Praktikumsziel: Im ersten Praktikum sollen Sie sich mit den grundlegenden Werkzeugen vertraut machen, die für alle weiteren Aufgaben zum Einsatz kommen:

- die Entwicklungsumgebung *eclipse*
- der Java-Compiler und der Java-Interpreter

Dabei implementieren Sie in Java einen allerersten (klassischen) Algorithmus.

Voraussetzungen: Account am Institut - damit Sie uneingeschränkt arbeiten können

Account bei Moodle und Registrierung für dieses Praktikum - damit Sie Lösungen dieser und aller weiteren Aufgaben abgeben können

Eigene (lokale) Installation von *eclipse* ODER *remote* Zugang zu den gruenau-Rechnern am Institut.

Aufgabe: Kenntnisse aus der 1. Praktikumswoche (*remote access* und *eclipse*)
Experimentieren Sie mit *eclipse* und dem Java-Compiler/-Interpreter (wie unten erläutert) und implementieren Sie dabei den Euklidischen Algorithmus http://de.wikipedia.org/wiki/Euklidischer_Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier positiver ganzer Zahlen in Java.

FAQ: Am Ende dieser Seite finden Sie die am häufigsten gestellten Fragen sowie (spätere) Konkretisierungen der Aufgabenstellung. Sie sind Teil der Aufgabenstellung und könnten sich während des Bearbeitungszeitraumes ändern, wir werden Sie ggf. darüber im Forum zu dieser Veranstaltung informieren.

Sollten Sie sich die Aufgabenstellung ausdrucken, so achten Sie bitte darauf, dass sich die FAQ weiterhin ändern können.

Senden Sie die Datei **GGT.java** über **Moodle** ein.

Abgabe: Diese Aufgabe ist noch eine Einzelabgabe, d.h. **jeder muss seine Lösung bei Moodle separat abgeben!**

Aufgabe

Erstellen Sie (am besten in einem neuen *eclipse*-Projekt) ein Java-Programm **GGT.java**, welches (in Anlehnung an die Berechnung der Summe zweier Zahlen siehe auch [https://moodle.hu-berlin.de/pluginfile.php/3416401/mod_assign/intro/A01.pdf]) den größten gemeinsamen Teiler von zwei positiven ganzen Zahlen nach dem Euklidischen Algorithmus berechnet und ausgibt.

Das Prinzip des Euklidischen Algorithmus wird auch gegenseitige Wechselwegnahme genannt. Eingangsgrößen seien zwei positive ganze Zahlen a und b . Bei der Berechnung verfährt man nach Euklid wie folgt:

Das Prinzip des euklidischen Algorithmus wird auch <i>gegenseitige Wechselwegnahme</i> genannt. Eingangsgrößen sind zwei natürliche Zahlen a und b . Bei der Berechnung verfährt man nach Euklid wie folgt:	
1.	Setze $m = a$; $n = b$.
2.	Ist $m < n$, so vertausche m und n .
3.	Berechne $r = m - n$.
4.	Setze $m = n$, $n = r$.
5.	Ist $r \neq 0$, so fahre fort mit Schritt 2.
Nach Ablauf des Verfahrens hat man mit m den ggT von a und b gefunden.	

Überzeugen Sie sich durch geeignete Tests von der Korrektheit Ihrer Implementation. Versuchen Sie auch den Algorithmus strukturell so zu formulieren, dass man die obige verbale Beschreibung wiedererkennen kann. Sorgen Sie dafür, dass im Programm geprüft wird, ob die beiden als Argumente übergebenden Zahlen tatsächlich positiv (also > 0) sind. Ansonsten funktioniert der oben beschriebene Ablauf vielleicht gar nicht richtig!? Das macht man in Java beispielsweise so:

```
boolean someCondition = ...;
// diese muessen Sie natuerlich geeignet formulieren
if (someCondition) {
    System.out.println("eine passende Ausschrift");
    System.exit(-1); // Programm sofort beenden
    // negative Parameter bei exit weisen den Rufer
    // auf ein abnormales Ende hin
}
```

Abschließend hier noch ein paar Beispielaufrufe, die zeigen, wie sich Ihr Programm verhalten soll - offenbar müssen Sie sich NICHT darum kümmern, dass tatsächlich zwei Zahlen als Argumente übergeben werden.

```
$ java GGT 12 14
ggT(12, 14) = 2
$ java GGT 36 12
ggT(36, 12) = 12
$ java GGT 36 -12
nur positive ganze Zahlen als Argumente erlaubt
$ java GGT -36 12
nur positive ganze Zahlen als Argumente erlaubt
$ java GGT 666
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
    at GGT.main(GGT.java:6)
$ java GGT 6 sieben
Exception in thread "main" java.lang.NumberFormatException: For input string:
"sieben"
```

```
at
java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Integer.parseInt(Integer.java:492)
at java.lang.Integer.parseInt(Integer.java:527)
at GGT.main(GGT.java:6)
```

ACHTUNG

Um die Korrektur der Aufgaben durch die Tutoren zu erleichtern, gelten (auch bei allen weiteren Aufgaben) die folgenden Regeln:

Eingesandte Java-Programme, die nicht vom Java-(Referenz-)Compiler akzeptiert werden (d.h. Fehler bei der Übersetzung liefern) werden ohne weitere Inspektion mit 0 (Leistungs-) Punkten bewertet! Ob Sie die 5 Abgabepunkte erhalten, entscheiden wir in Abhängigkeit davon, ob Ihre Einsendung genügend mit der Aufgabe zu tun hat ;-)

Halten Sie sich bei der Benennung von Klassen, Funktionen und Dateien immer an die Vorgaben und achten Sie darauf, dass Sie **keine** Packages verwenden (nur *default package*). Alle Abweichungen von diesen Regeln führen zu zusätzlichem Aufwand bei der Bewertung und daher auch zu Punktabzug. Vergessen Sie nicht den abschließenden Test auf der Referenzplattform!

Die vom implementierten Programm erzeugten Ausgaben müssen das geforderte Format exakt einhalten (incl. Zeilenstruktur und Leerzeichen). Soll und Ist werden automatisch verglichen. Sofern Abweichungen festgestellt werden, gibt es Punktabzug, über dessen Höhe dann die Tutoren entscheiden! Die einzig zulässige Abweichung (eine die man bei der Textansicht des Programms gar nicht feststellen kann) besteht in der Codierung der Zeilenenden, die unter Windows mit zwei Zeichen (carriage return [/x0d] + line feed [/x0a]) unter Unix dagegen nur mit einem Zeichen (newline [/x0a]) dargestellt werden.

FAQ

1. **Q:**

Was sind denn Abgabe- und Leistungspunkte?

A:

Wenn Sie die [Praktikumsordnung](#) gelesen haben, sollte klar sein, dass Sie alle Aufgaben abgeben müssen. Die Abgabepunkte erhalten Sie dann, wenn Sie eine erkennbar ernst gemeinte Abgabe eingereicht haben. Die eigentlich in der jeweiligen Aufgabe geforderte Leistung wird dann in Abhängigkeit von der Erfüllung mit Leistungspunkten bewertet.

Bitte schauen Sie vor der Abgabe immer noch einmal online in die FAQs - falls es Korrekturen oder Modifikationen an der Aufgabestellung geben sollte, werden diese dort explizit erläutert!

Das war die 1. (richtige) Praktikumsaufgabe! - Viel Erfolg bei der Bearbeitung

Letzte Änderung 15. November 2020

Inhaltliche Fragen/Korrekturen bitte an: [Ahrens](#)