

Übungsblatt 2

E-Learning

Absolvieren Sie die Tests bis Di., 06.11., 23:55 Uhr.

Die Tests sind in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegt.

Sie können einen Test **nur einmal durchlaufen**. Sobald Sie einen Test starten steht Ihnen nur eine **begrenzte Zeit** zu Verfügung, um den Test zu bearbeiten.

Alle Punkte, die Sie beim Test erreichen, werden ihnen angerechnet.

ILIAS 4-Minuten-Aufgaben – 12 Punkte

Absolvieren sie die Tests

- *Informatik I - ILIAS 02 Teil 1,*
- *Informatik I - ILIAS 02 Teil 2,*
- *Informatik I - ILIAS 02 Teil 3.*

(12 Punkte)

Hinweis

Wenn Sie den Test einmal vollständig durchlaufen haben kommen Sie auf die Seite *Testergebnisse*. Starten Sie den Test erneut aus Stud.IP, ist jetzt auch eine Schaltfläche *Testergebnisse anzeigen* vorhanden, die auf diese Seite führt.

Auf der Seite *Testergebnisse* können Sie sich unter *Übersicht der Testdurchläufe* zu jedem Testdurchlauf *Details anzeigen* lassen.

In der Auflistung der Aufgaben führt der Titel einer Aufgabe zu einer **Musterlösung** für die jeweilige Aufgabe.

LON-CAPA – 42 Punkte

Maschinensprache, Ablaufprotokoll, Trace-Tabelle

Absolvieren Sie im Test *Informatik I - LON-CAPA* die *Übung 02*.

(42 Punkte)

Übung

Abgabe bis Di., 06.11., 18 Uhr.

Werfen Sie Ihre Lösung in den Zettelkästen Ihrer Gruppenübung. Für die Übungen im Nordbereich stehen die Zettelkästen im Sockelgeschoß (Ebene -1) **oder** auf dem Flur vor dem Seminarraum auf Ebene 0 des Instituts für Informatik.

Wenn Ihre Übung im Südbereich stattfindet, klären Sie mit Ihrem Tutor wo die Lösungen abzugeben sind.

Achten Sie darauf, dass Ihr **Name**, Ihre **Gruppe** und Ihre **Matrikelnummer** auf **jedem** Blatt stehen!

Falls Ihre Lösung mehrere Blätter umfasst, heften Sie diese bitte zusammen.

Aufgabe 1 – 11 Punkte

Algorithmus

Gesucht ist ein Algorithmus zum Quadrieren positiver ganzer Zahlen.

Problemspezifikation

- Gegeben: Eine positive ganze Zahl x
- Gesucht: Das Quadrat x^2
- Erlaubte Operationen: Großes Eins-plus-Eins (nur Addition $a + b$ für `int`-Werte a, b und $a, b \geq 0$).

Idee

Das Quadrat einer positiven ganzen Zahl x ist die Summe der ersten x ungeraden positiven ganzen Zahlen beginnend bei Eins.

Aufgabe

Entwickeln und formulieren Sie, nach dem Vorbild der klassischen und modernen Version des Euklidischen Algorithmus aus Kapitel 1.2.2 des Skripts, einen passenden Algorithmus, präzisiert als Java-Code (Fragment).

(11 Punkte)

Praktische Übung

Abgabe der Prüfsumme bis Di., 06.11., 23:55 Uhr.

~~Testat von Mi., 07.11., 8-10 Uhr bis Mo., 12.11., 18-20 Uhr.~~

Testat von Di., 13.11., 8-10 Uhr bis Mo., 19.11., 18-20 Uhr.

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen, insbesondere in den Rechnerübungen **Mo., 12.11., 8-10 Uhr und 18-20 Uhr**, in denen **keine Testate** stattfinden.

Lesen Sie die **Hinweise zu den praktischen Übungen**, die in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien* → *Übungsblätter* hinterlegt sind.

- Erstellen Sie ein Archiv, dass **alle Dateien** enthält, die Sie beim Testat vorstellen möchten.
- Beim Testat werden nur Dateien aus einem Archiv testiert, dessen Prüfsumme **exakt** der von Ihnen übermittelten Prüfsumme entspricht.
- Berechnen Sie die Prüfsumme des Archivs mit dem **sha1sum** Befehl.
- Um die praktische Übung testieren zu lassen **müssen** Sie einen Termin in der zugehörigen Rechnerübung reservieren. Ein Testat ohne Termin ist nicht möglich. Testate zu einer anderen praktischen Übung können nur in Ausnahmefälle nach Rücksprache mit Herrn Brosenne durchgeführt werden.
- Übermitteln Sie die Prüfsumme durch Absolvieren des in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegten Tests *Informatik I - ILIAS 02 Testat*.
- Öffnen Sie die Terminvergabe für diese praktische Übung und lassen Sie den von Ihnen reservierten Termin anzeigen.
- Entpacken Sie das Archiv erst nachdem der Tutor die Prüfsumme kontrolliert hat.

Aufgabe 1 – 10 Punkte

Remote Login

Es besteht, per SSH (*secure shell*), die Möglichkeit der entfernten Anmeldung (*remote login*) in den Rechnerpool des Instituts für Informatik über folgenden Anmeldeserver.

`shell.stud.informatik.uni-goettingen.de`

Für die Anmeldung ist eine Zwei-Faktor-Authentifizierung nötig, informieren Sie sich darüber in folgenden Quellen.

<https://user.informatik.uni-goettingen.de/~damm/info1/aktuell/2FA.html>

<https://doc.informatik.uni-goettingen.de/wiki/index.php/Shell>

Führen Sie eine Anmeldung an `shell.stud.informatik.uni-goettingen.de` vor.

(10 Punkte)

Aufgabe 2 – 8 Punkte

Automatische Trace-Tabelle

Laden Sie die, in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien*→*Übungsblätter*→*Daten* hinterlegten, Quelltexte der Klassen `EuklidKlassisch` und `EuklidModern`, für die Lösung des ggT-Problems mit den Euklidischen Algorithmen aus der Vorlesung, auf Ihren Rechner.

EuklidKlassisch

<code>// ggT(x,y)</code>	<code>// K1</code>
<code>while (y != 0) {</code>	<code>// K2</code>
<code> if (x > y)</code>	<code>// K3</code>
<code> x = x - y;</code>	<code>// K4</code>
<code> else</code>	
<code> y = y - x;</code>	<code>// K5</code>
<code>}</code>	
<code>System.out.println(x);</code>	<code>// K6</code>

EuklidModern

<code>// ggT(x,y)</code>	<code>// M1</code>
<code>int r;</code>	
<code>while (y != 0) {</code>	<code>// M2</code>
<code> r = x % y;</code>	<code>// M3</code>
<code> x = y;</code>	<code>// M4</code>
<code> y = r;</code>	<code>// M5</code>
<code>}</code>	
<code>System.out.println(x);</code>	<code>// M6</code>

Versehen Sie die Quelltexte der Klassen `EuklidKlassisch` und `EuklidModern` mit `System.out.println`-Anweisungen, so dass eine einfache Trace-Tabelle (Ablauf-Protokoll), entsprechend der nachfolgenden Beispiele, *von selbst* entsteht. Berücksichtigt werden müssen nur die mit `//K` und `//M` gekennzeichneten Zeilen.

Schreiben Sie mehrere Trace-Tabellen für `EuklidKlassisch` und `EuklidModern` mit Hilfe der Ausgabeumleitung in Textdateien.

(8 Punkte)

Beispiel

```
> java EuklidKlassisch 15 35
| line | x   | y   | comment
| K1   | 15  | 35  | ggT(15, 35)
| K2   | 15  | 35  | while(35 != 0)
| K3   | 15  | 35  |   if(15 > 35)
| K5   | 15  | 20  |       y = 35 - 15
| K2   | 15  | 20  |   while(20 != 0)
| K3   | 15  | 20  |     if(15 > 20)
| K5   | 15  | 5   |         y = 20 - 15
| K2   | 15  | 5   |   while(5 != 0)
| K3   | 15  | 5   |     if(15 > 5)
| K4   | 10  | 5   |         x = 15 - 5
| K2   | 10  | 5   |   while(5 != 0)
| K3   | 10  | 5   |     if(10 > 5)
| K4   | 5   | 5   |         x = 10 - 5
| K2   | 5   | 5   |   while(5 != 0)
| K3   | 5   | 5   |     if(5 > 5)
| K5   | 5   | 0   |         y = 5 - 5
| K2   | 5   | 0   |   while(0 != 0)
| K6   | 5   | 0   | = 5
```

Beispiel

```
> java EuklidModern 15 35
| line | x   | y   | r   | comment
| M1   | 15  | 35  | -   | ggT(15, 35)
| M2   | 15  | 35  | -   | while(35 != 0)
| M3   | 15  | 35  | 15  |   r = 15 % 35
| M4   | 35  | 35  | 15  |   x = 35
| M5   | 35  | 15  | 15  |   y = 15
| M2   | 35  | 15  | 15  | while(15 != 0)
| M3   | 35  | 15  | 5   |   r = 35 % 15
| M4   | 15  | 15  | 5   |   x = 15
| M5   | 15  | 5   | 5   |   y = 5
| M2   | 15  | 5   | 5   | while(5 != 0)
| M3   | 15  | 5   | 0   |   r = 15 % 5
| M4   | 5   | 5   | 0   |   x = 5
| M5   | 5   | 0   | 0   |   y = 0
| M2   | 5   | 0   | 0   | while(0 != 0)
| M6   | 5   | 0   | -   | = 5
```

Hinweis

Zu Formatierung der Aufgabe können Sie das Tabulatorzeichen `\t` verwenden.

Aufgabe 3 – 17 Punkte

Einfache und iterierte Quersumme in Java

Die Quersumme einer Zahl ist die Summe ihrer Ziffern.

Beispiel. Die Quersumme von 99 ist 18 ($9 + 9 = 18$).

Die iterierte Quersumme erhält man, indem man so lange wiederholt die Quersumme bildet, bis das Ergebnis einstellig ist.

Beispiel. Die iterierte Quersumme von 99 ist 9 ($99 \rightarrow 9 + 9 = 18 \rightarrow 1 + 8 = 9$).

1. Schreiben Sie ein Java-Applikation, die eine natürliche Zahl auf der Kommandozeile übergeben bekommt, die Quersumme dieser Zahl berechnet und ausgibt.

Lassen Sie die Quersumme der Zahlen 42, 3678 und 6947859 berechnen.

(7 Punkte)

2. Schreiben Sie auf Grundlage der Applikation aus Aufgabenteil 1 eine Java-Applikation, die für eine natürliche Zahl die iterierte Quersumme einer Zahl berechnet und ausgibt.

Testen Sie Ihr Programm mit den Zahlen 24, 8763 und 9587496.

(10 Punkte)

Hinweis

Zu jedem Aufgabenteil wird eine eigene übersetzbare und lauffähige Applikation erwartet.