

# Programmierübungen

Wintersemester 2007/2008

## 12. Übungsblatt

23. Januar 2008

Abgabe bis Freitag, 8. Februar 23:59 Uhr

Die Abgabe Ihrer Bearbeitung können Sie im eClaus-System durchführen. Erarbeiten Sie Lösungsideen zu den Aufgaben möglichst in Kleingruppen. Es wird jedoch von Ihnen erwartet, dass jeder Teilnehmer eine eigene Lösung abgibt. Sollten kopierte Quelltexte abgegeben werden, so werden grundsätzlich alle Kopien mit 0 Punkten bewertet. In den Vortragsfolien der Programmierübungen oder im Skript zur Einführung in die Informatik abgedruckte Quelltexte können verwendet werden, müssen aber der Programmierrichtlinie entsprechend formatiert und kommentiert werden.

Beachten Sie die Programmierrichtlinie und kommentieren Sie Ihren Quelltext! Dokumentieren Sie unbedingt Ihre Lösungsidee in den Quelltext-Kommentaren.

<http://www.iste.uni-stuttgart.de/ps/Lehre/WS0708/inf-prokurs>

*Bitte beachten Sie, dass zum Scheinerwerb unter anderem auf den letzten 4 Übungsblättern 50 % der Punkte erreicht werden müssen. Dies ist das letzte Übungsblatt.*

### Aufgabe 12.1: Große Zahlen

(20 Punkte)

In dieser Aufgabe sollen Sie die Techniken, die Sie in diesem Semester kennen gelernt haben gegeneinander abwägen und selbst zu einer praxistauglichen Lösung kommen.

Implementieren Sie ein Programm `big_factorial`, das die Fakultät seines ersten Kommandozeilen-Arguments berechnet. Als Kommandozeilen-Argument soll jede natürliche Zahl, dargestellt im Dezimalsystem, zugelassen sein. Ihr Programm sollte die Fakultät von 1000 exakt berechnen können.

Dokumentieren Sie unbedingt, welche Datenstrukturen Ihr Programm wozu verwendet und wie die geforderte Funktionalität umgesetzt wurde.

Hinweise:

- Beachten Sie, dass die erwarteten Resultate sehr groß werden können und ohne Genauigkeitsverlust nicht durch einfache Datentypen (`Integer` oder `Long_Integer`) modelliert werden können. Finden Sie eine geeignete Datenstruktur, um Zahlen von beliebiger Länge repräsentieren zu können.
- Erinnern Sie sich zurück an Ihre Grundschulzeit. Dort wurden Ihnen Algorithmen zum schriftlichen Subtrahieren und Multiplizieren präsentiert. Diese Algorithmen können leicht nachprogrammiert werden.
- Geben Sie Speicher für Zahlen, die in Ihrer Berechnung nicht mehr benötigt werden frei. Vermeiden Sie die rekursive Implementierung der Fakultät.
- Stellen Sie sich (je nach gewählter Datenstruktur) auf Rechenzeiten im zweistelligen Sekundenbereich ein. Lassen Sie Ihr Programm eventuell an geeigneten Punkten kurze Ausgaben machen, um ein Feedback zu erhalten, ob die Berechnung noch läuft. Das Endresultat muss von dieser Debug-Ausgabe klar abgetrennt werden und erkennbar sein.

- Legen Sie den Fokus nicht auf die effizienteste Lösung, sondern auf Einfachheit und Verständlichkeit Ihrer Implementierung.

Beispiel Punkteverteilung (Ihre Lösung muss nicht dieser Struktur folgen):

- Wahl einer geeigneten Datenstruktur für große Zahlen: 3 Punkte
- Unterprogramme zum Umwandeln eines Strings in die gewählte Datenstruktur und zurück: 3 Punkte
- Subtraktion große Zahl minus 1: 3 Punkte
- Multiplikation zweier großer Zahlen: 4 Punkte
- Gesamtablauf inkl. Ein-/Ausgabe: 4 Punkte
- Vollständige und frühzeitige Freigabe von nicht mehr benötigtem Speicher: 3 Punkte