

# Programmierübungen

Wintersemester 2007/2008

## 1. Übungsblatt

22. Oktober 2007

Abgabe bis Freitag, 9. November 2007

Die Abgabe Ihrer Bearbeitung können Sie im eClaus-System durchführen. Quelltexte, die Teil der Aufgabenstellung sind, werden auf der Webseite zur Veranstaltung zur Verfügung gestellt. 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.

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>

### Aufgabe 1.1: Schleifen

(6 Punkte)

Schreiben Sie drei Varianten eines Ada-Programms, mit diesem identischen Ablauf:

1. Das Programm gibt den Text „Größte Zahl: “ aus.
2. Der Benutzer gibt eine natürliche Zahl  $n$  (inkl. 0) ein.
3. Das Programm gibt zwei Listen aller ganzen Zahlen im Bereich  $1 \dots n$  aus, die eine aufsteigend, die andere absteigend sortiert. Wurde 0 eingegeben, so soll eine leere Liste ausgegeben werden.

Verwenden Sie ...

a) eine **while**-Schleife (`count_while.adb`),

b) eine **loop**-Schleife (`count_loop.adb`),

c) eine **for**-Schleife (`count_for.adb`),

Beispiel:

```
Größte Zahl: 8
1, 2, 3, 4, 5, 6, 7, 8
8, 7, 6, 5, 4, 3, 2, 1
```

### Aufgabe 1.2: Text Ein-/ Ausgabe

(3 Punkte)

Schreiben Sie ein Ada Programm `replace.adb`, das den Benutzer zur Eingabe einer Zeile Text auffordert, in dem Text alle Leerzeichen ‚ ‘ durch Unterstriche ‚\_‘ ersetzt und den resultierenden Text ausgibt. Sie können davon ausgehen, dass der Benutzer nur Texte eingibt, die kürzer als 256 Zeichen sind.

Beispiel:

```
Ein: Hier steht ein Text.
Aus: Hier_steht_ein_Text.
```

### Aufgabe 1.3: Stellenwertsysteme

(5 Punkte)

Aus der Vorlesung Einführung in die Informatik kennen Sie Stellenwertsysteme. Schreiben Sie ein Programm, das eine ganze Zahl aus einem Stellenwertsystem in ein anderes umwandeln kann, mit folgendem Ablauf:

1. Das Programm gibt den Text „Basis der Eingabe: “ aus.
2. Der Benutzer gibt eine Dezimalzahl im Bereich 2 – 36 ein. Ist die eingegebene Zahl außerhalb dieses Bereichs, so darf das Programm unkontrolliert abbrechen.
3. Das Programm gibt den Text „Eingabe: “ aus.
4. Der Benutzer gibt eine Zahl in dem Stellenwertsystem der Basis aus Schritt 2 ein. Als Ziffern dienen die arabischen Ziffern ‚0‘–‚9‘ und die Buchstaben ‚A‘–‚Z‘ sowie ‚a‘–‚z‘ nach Wahl des Benutzers. Groß- und Kleinbuchstaben dürfen auch gemischt werden. Nur diese Zeichen, sowie ein führendes ‚-‘ sind zulässige Eingabezeichen. Sollte die Eingabe andere Zeichen enthalten, so gibt das Programm die Meldung „Eingabe nicht verstanden.“ aus und bricht ab.
5. Das Programm gibt den Text „Basis der Ausgabe: “ aus.
6. Der Benutzer gibt eine Dezimalzahl im Bereich 2 – 36 ein. Ist die eingegebene Zahl außerhalb dieses Bereichs, so darf das Programm unkontrolliert abbrechen.
7. Das Programm gibt die in Schritt 4 eingegebene Zahl als Zahl im Stellenwertsystem zur Basis aus Schritt 6 aus.

Nennen Sie Ihr Programm „Convert“, die Quelltext-Datei des Hauptprogramms folglich „convert.adb“. Ihr Programm soll mindestens die Zahlen im Bereich –Integer‘Last .. Integer‘Last verarbeiten können. Stellen Sie sicher, dass Zwischenergebnisse Ihrer Berechnungen keine Überläufe verursachen (nicht zu groß oder zu klein werden).

Beispielabläufe:

```
Basis der Eingabe: 16
Eingabe: -2A3f
Basis der Ausgabe: 3
Ausgabe: -112211120
```

```
Basis der Eingabe: 36
Eingabe: ABC
Basis der Ausgabe: 10
Ausgabe: 13368
```

```
Basis der Eingabe: 2
Eingabe: x
Eingabe nicht verstanden.
```

### Aufgabe 1.4: Taschenrechner

(6 Punkte)

Laden Sie von der Programmierübungs-Webseite die Datei `calculator.adb` herunter. Ersetzen Sie die „-- TODO: ...“ Kommentare, so dass ein Programm mit folgender Funktionalität entsteht. Verwenden Sie die Funktion `Calculate`:

1. Das Programm gibt aus: „linker Operand: “
2. Der Benutzer gibt eine ganze Zahl ein
3. Das Programm gibt aus: „rechter Operand: “
4. Der Benutzer gibt eine ganze Zahl ein
5. Das Programm gibt aus: „Operation: “
6. Der Benutzer gibt eines der Zeichen `+, -, *, /` ein
7. Das Programm gibt das Resultat der Berechnung aus
8. Ist das Resultat 0, so beendet das Programm. Andernfalls wird das Resultat als linker Operand weiterverwendet und es wird mit Schritt 3 fortgefahren.

Sollte eine Berechnung zu einem Überlauf oder zu Division durch 0 führen, so darf Ihr Programm unkontrolliert abstürzen