Programmierkurs I (Inf., W-Info.), Übungsblatt 5

Claus/Weicker, Wintersemester 03/04 — Abgabetermin: 27.11.2003, 23:59 Uhr

Aufgabe 1: Sortieren (mittel) — 5 Punkte

Schreiben Sie ein Programm `sortieren.adb`, welches den Benutzer eine Sequenz von Zahlen eingeben lässt und diese dann in sortierter Reihenfolge wieder ausgibt, wobei die Ausgabe mit der kleinsten eingegebenen Zahl anfangen und mit der größten enden soll. Die Anzahl soll unbegrenzt sein.

```
Wieviele Zahlen geben Sie ein? 11
Zahlen: 77 7 81 4 23 5 4 99 1 45 34
Ergebnis: 1 4 4 5 7 23 34 45 77 81 99
```

Hinweis: Das Sortierverfahren braucht nicht effizient zu sein. Wenn Ihnen nichts Besseres einfällt, benutzen Sie folgende Methode: Suchen Sie zunächst das Minimum unter den eingegebenen Zahlen und geben es aus, dann das Minimum unter den restlichen Zahlen usw., bis Sie alle Zahlen haben.

Aufgabe 2: Löschen in einem Text (schwer) — 5 Punkte

Schreiben Sie ein Programm `textloeschen.adb`, welches zunächst zwei Zeichenketten s und t einliest. Anschließend sollen alle in s vorkommenden Zeichenketten t gelöscht und der modifizierte Zeichenstring ausgegeben werden.

```
Geben Sie einen Text ein: abcdefggggh
Geben Sie den Loeschtext ein: ggg
Ergebnis: abcdefh

Geben Sie einen Text ein: abcdefggh
Geben Sie den Loeschtext ein: ggg
Ergebnis: abcdefggh
```

Benutzen Sie den Datentyp `Unbounded_String` zur Speicherung der Texte. In diesem Programm sollten Sie ausschließlich mit den Funktionen `Get_Line`, `Length`, `Element` und `Replace_Element` auskommen.

Aufgabe 3: Galgenraten (mittel bis schwer) — 10 Punkte

Bei „Galgenraten“ handelt es sich um ein Spiel, welches wie folgt abläuft: Ein Spieler (der Fragesteller) denkt sich ein Wort aus und verrät dem anderen Spieler (dem Rater) die Länge des Wortes. Der Rater muss nun dieses Wort raten, indem er dem Fragesteller einzelne Buchstaben nennt. Nennt er einen Buchstaben, der im Wort vorkommt, verrät ihm der Fragesteller, an welchen Stellen im Wort der Buchstabe vorkommt. Nennt der Rater einen Buchstaben, der nicht im Wort vorkommt, so zählt dies als Fehlversuch. Der Fragesteller gewinnt das Spiel, sobald der Rater 10 Fehlversuche hatte. Der Rater gewinnt das Spiel, sobald er alle Buchstaben erraten hat, falls er dies mit weniger als 10 Fehlversuchen schafft.
Implementieren Sie dieses Spiel als Programm `galgenraten.adb`, wobei der Rechner die Rolle des Fragestellers übernimmt. Als erstes wird eine Eingabedatei abgefragt, die das zu erratende Wort enthält. Das Wort soll nur aus den Großbuchstaben `A` bis `Z` bestehen. Am Anfang soll die Länge des Worts mit einer entsprechende Zahl von Strichen (`-`) angezeigt werden. Nach jedem erratenen Buchstaben soll die Ausgabe erneuert werden, wobei die bereits erratenen Buchstaben dort angezeigt werden, wo sie vorkommen. Aus Gründen der Benutzerfreundlichkeit sollen auch Kleinbuchstaben eingegeben werden können. Wird ein anderes Zeichen als ein Buchstabe oder ein Buchstabe ein zweites Mal eingegeben, erfolgt eine entsprechende Mitteilung und die Eingabe zählt nicht als Fehlversuch. Anstatt des üblichen Galgens, der abhängig von der Anzahl der Fehlversuche gezeichnet wird, werden nur bei jeder unpassenden Eingabe die Anzahl der Fehlversuche ausgegeben. Nach 10 Fehlversuchen bricht das Spiel ab.

Beispiel: Wenn die Datei `galgenwort.txt` das Wort `PROGRAMMIERKURS` enthält, läuft ein Beispieldialog wie folgt ab.

```
Geben Sie die Datei mit dem Ratewort ein: galgenwort.txt
---------------
Ihr nächster Buchstabe? m
------MM-------
Ihr nächster Buchstabe? R
-R--R-MM--R--R-
Ihr nächster Buchstabe? Z
Falsch (1 von 10 Fehlversuchen)
Ihr nächster Buchstabe? M
Bitte geben Sie nur unbenutzte Buchstaben ein!
Ihr nächster Buchstabe? %
Bitte geben Sie nur unbenutzte Buchstaben ein!
Ihr nächster Buchstabe? O
-RO-R-MM--R--R-
⋮
Ihr nächster Buchstabe? X
Falsch (10 von 10 Fehlversuchen)
Sie haben verloren!
```

Wird das Wort mit höchstens 9 Fehlversuchen erraten, lautet die letzte Ausgabe wie folgt:

```
PROGRAMMIERKURS
Sie haben gewonnen!
```

Die Datei ist auf der Web-Seite des Programmierkukrses erhältlich.

Hinweise

- Zur Abgabe wird das eClaus-System verwendet:
  `http://eclaus.informatik.uni-stuttgart.de`
- Die Abgabe zu jeder Teilaufgabe besteht aus einem kompilierbaren Ada-Quelldatei. In jeder Aufgabe wird ein Dateiname vorgegeben. Verwenden Sie diesen bitte für das Hauptprogramm. Auch sind in der Aufgabe Angaben zu Ein- und Ausgabetexten sowie zur Formatierung der Ausgabe enthalten. Bitte folgen Sie diesen Angaben so genau wie möglich.
- Beachten Sie beim Programmieren bitte die folgenden Hinweise („kleine Programmierrichtlinie“).
  - Halten Sie einzelne Bestandteile überschaubar, z.B. indem Sie nur eine Anweisung pro Zeile schreiben, sowie pro Zeile höchstens 80 Zeichen, höchstens 40 Zeilen pro Prozedur/Funktion, nicht mehr als 800 Zeilen pro Datei und höchstens 5 Parameter bei Prozeduren/Funktionen benutzen.
  - Bezeichner sollen selbsterklärend und maximal 15 Zeichen lang sein. Bezeichner enthalten nur Buchstaben, den Bindestrich oder den Unterstrich.
  - Durch Einrückung um 2 Zeichen soll die logische Gliederung eines Programms verdeutlicht werden. Beispielsweise wird der Anweisungsteil einer IF-Verzweigung eingerückt, während „`end if;`“ nicht mehr eingerückt wird. Auch auf der nächsten Zeile fortgesetzte Zeilen werden um 2 Zeichen eingerückt.
  - Zu Beginn des Programms muss in Kommentaren das Konzept und die Lösungsidee des Programms ausführlich erläutert werden.
  - Auch im Programmtext sind Kommentare einzufügen, um den Code zu erläutern.
- Beachten Sie, dass jede Abgabe individuell vom jeweiligen Studierenden erstellt werden muss. Werden von den Tutoren Plagiate erkannt, d.h. exakte oder leicht modifizierte Kopien, werden für die Aufgabe keine Punkte vergeben. Falls Sie die Lösung der Aufgaben vor der Abgabe

in Gruppen besprechen, achten Sie darauf, dort nur das generelle Konzept abzuklären und die Programmierung jedem selbst zu überlassen.

- Bei Fragen wenden Sie sich bitte an Ihren Tutor oder an die Übungsleitung: `Karsten.Weicker@fmi.uni-stuttgart.de` oder Tel. 7816-337
- Weitere Veranstaltungshinweise einschließlich der Übungsblätter finden Sie unter: `http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ws03-04/ada95/`