

# Programmierübungen

Wintersemester 2006/2007

## 4. Übungsblatt

17. November 2006

Abgabe bis Samstag, 25. November 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/WS0607/inf-prokurs>

Hinweis: Zur Ein-/Ausgabe in Dateien, zur Verwendung von Kommandozeilen-Argumenten und zur Behandlung von Ausnahmen (exceptions) werden am Montag, 20.11., 8:00 Uhr in V38.01 noch Hinweise gegeben.

### Aufgabe 4.1: Datei-Statistiken (7 Punkte)

Schreiben Sie ein Programm „File\_Stats“, das Text-Dateien analysiert. Ihr Programm gibt für eine analysierte nicht-leere Datei folgende Statistik aus:

```
Analysierte Datei: hangman.ads
Durchschnittliche Zeilenlaenge: 33 Zeichen
Kuerzeste Zeile: 0 Zeichen
Laengste Zeile: 79 Zeichen
Anzahl Zeilen: 175
Anzahl ';' : 12
```

Sollte jedoch die Datei keinen Inhalt besitzen, so erfolgt nur die folgende Ausgabe:

```
Analysierte Datei: empty.txt
Datei ist leer.
```

1. Ihr Programm fragt den Benutzer nach einem Dateinamen, liest die Datei dieses Namens ein und analysiert sie. Behandeln Sie auftretende Fehler (z. B. Datei nicht vorhanden) und geben Sie im Fehlerfall sinnvolle Meldungen aus.
2. Erweitern Sie Ihr Programm aus Teilaufgabe 1, so dass die Namen von Dateien als Kommandozeilen-Argumente übergeben werden können.
  - Wird kein Kommandozeilen-Argument angegeben, so soll weiterhin die Eingabe vom Benutzer verlangt werden.

- Wird mindestens ein Kommandozeilen-Argument angegeben, so soll der Benutzer nicht mehr zur Eingabe eines Namens aufgefordert werden.
- Werden mehrere Kommandozeilen-Argumente angegeben, so sollen die Dateien nacheinander analysiert werden. Es wird für jede Datei eine separate Statistik ausgegeben.

Hinweis: verwenden Sie zur Durchführung der Analyse eine Prozedur (z. B. `Analyse`), der ein Dateiname als Parameter übergeben werden kann.

## Aufgabe 4.2: Abstrakte Datenobjekte (13 Punkte)

Sicher kennen Sie das Galgenraten-Spiel. Darin denkt sich der Spielleiter ein Wort aus und schreibt so viele Unterstriche wie das Wort Buchstaben hat auf ein Papier. Der Spieler darf dann wiederholt einen Buchstaben raten und alle Positionen des Worts, an denen dieser Buchstabe vorkommt, werden aufgedeckt. Rät der Spieler einen Buchstaben, der im gesuchten Wort nicht vorkommt, oder der bereits aufgedeckt wurde, so wird ein Strich an eine Galgenzeichnung angefügt. Sobald der Galgen samt gehängtem Strichmännchen fertiggestellt ist, hat der Spieler verloren. Hat der Spieler jedoch das Wort vollständig aufgedeckt, so hat er gewonnen.

Eine unvollständige Implementierung des Spiels können Sie von der Übungs-Webseite herunterladen. Entpacken Sie das Archiv `hangman.zip`. Darin finden sich das Hauptprogramm Galgenspiel sowie das Paket `Hangman`. Das Paket bietet durch seine Unterprogramme Zugriffsmöglichkeit auf einige globale Variablen. Zur Laufzeit des Programms dient das Paket als Abstraktion für den Spielzustand. Die Unterprogramme des Pakets führen die Spielaktionen durch und lassen nur regelkonforme Veränderungen des Spielzustands zu.

**galgenspiel.adb:** Hauptprogramm des Spiels. Dies können Sie nach Ihren Wünschen verändern. Es dient nur als zusätzliche Erklärung der Funktionsweise, sowie zum Testen des Pakets `Hangman`. Beachten Sie, dass das Hauptprogramm nicht alle Verwendungsmöglichkeiten des Pakets ausreizt und deshalb kein vollständiger Test ist. Richten Sie sich beim Schreiben Ihres Quelltexts nach der Anforderungsspezifikation und nicht nach diesem Beispiel.

**hangman.ads:** Die Spezifikation des Pakets ist das Kernstück der Aufgabe. Sie darf nicht verändert werden. Hier finden Sie eine genaue Anforderungsspezifikation der erwarteten Funktionalität.

**hangman.adb:** Die Implementierung des Pakets soll von Ihnen geschrieben werden. Es sind bereits einige globale Variablen-Deklarationen vorhanden. Diese sind als Hinweis auf eine mögliche Implementierung gedacht. Beachten Sie die mit „TODO“ gekennzeichneten Kommentare. Diese kennzeichnen Stellen an denen auf jeden Fall Hand angelegt werden muss.

Hinweise (mögliches Vorgehen):

- Überlegen Sie sich zunächst, welche verschiedenen Spielzustände auftreten können und wie sich während des Spiels der aktuelle Spielzustand mit Variablen modellieren lässt.

- Dokumentieren Sie möglichst exakt, wie Sie die Variablen verwenden wollen.
- Prüfen Sie für jedes Unterprogramm, welche Veränderungen an den Werten der Variablen durchgeführt werden müssen und ob diese Veränderungen konsistent zu Ihrer Planung durchgeführt werden können.
- Schreiben Sie den Quelltext für die Unterprogramme.
- Testen Sie Ihre Implementierung. Modifizieren Sie auch das Galgenspiel-Programm um weitere Tests durchzuführen.
- Sie dürfen die Deklarationen der globalen Variablen im *Body* des Pakets Hangman verändern oder ersetzen. Nur die Datei `hangman.ads` darf nicht verändert werden.