Prof. Dr. Stefan Brass
Dipl.-Inform. Annett Thüring
Institut für Informatik
MLU Halle-Wittenberg

08. November 2010

Vorlesung "Objektorientierte Programmierung" — 1. Programmiertest (Aufgabe A) —

Hinweise/Regeln:

Vermeiden Sie bitte jedes Verhalten, was als Täuschungsversuch (miss-)verstanden werden könnte. Wir müssten Sie sonst disqualifizieren, d.h. Sie hätten den Programmiertest nicht bestanden. Sie werden sowohl durch das Aufsichtspersonal im Rechnerpool überwacht, als auch auf den Rechnern per Programm/Aufzeichnung, sowie über das Netz. Wenn wir nachträglich in der Aufzeichnung einen Täuschungsversuch erkennen sollten, müssen wir Ihnen auch einen zunächst als bestanden gewerteten Programmiertest wieder aberkennen.

- Bearbeitungsdauer: 60 Minuten (plus eventuell leichte Verlängerung).
- Am Ende gibt es nur "bestanden" und "nicht bestanden", keine Punkte für partiell korrekte Lösungen. Möglicherweise werden am Ende noch einige Minuten Bearbeitungszeit angehängt, wenn mehrere Teilnehmer ganz knapp vor einer fertigen Lösung sind — aber nicht funktionierende Lösungen werden nicht akzeptiert, selbst wenn nur eine Kleinigkeit fehlt.
- Sie dürfen bis zu 3 Blätter "Spickzettel"/"Quick Reference" verwenden, sowie ein Buch (nicht zu groß, es muß noch auf den Tisch passen ohne den Nachbarn zu stören — ein Aktenordner wäre nicht akzeptabel).
- Sie dürfen außerdem ein Blatt leeres Papier und einen Stift verwenden, um sich Notizen zu machen.
- Eigene Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone bitte ausschalten (oder mit der Aufsicht besprechen).
- Sie dürfen nur Terminalfenster für den Compileraufruf und Editor-Fenster offen haben. Selbstverständlich können Sie auch einen Debugger benutzen, sowie `make` und `man`. Dagegen ist z.B. ein Web-Browser nicht erlaubt, und auch kein EMail-Programm (allgemein keine Zugriffe auf das Internet). Ebensowenig dürfen Sie versuchen, Dateien außerhalb Ihres Homeverzeichnisses abzulegen, oder auf Dateien außerhalb Ihres Homeverzeichnisses zuzugreifen (außer Dateien, die der Compiler verwendet).
- Das Programm müssen Sie vollständig neu eintippen, und nicht durch Modifikation von eventuell in Ihrem Homeverzeichnis vorhandenen Dateien erstellen. Für `make` dürfen Sie ein vorhandenes `Makefile` benutzen und bei Bedarf modifizieren.
- Die Homeverzeichnisse werden für Zugriffe von außen gesperrt. Falls Sie spezielle Zugriffsrechte gesetzt hatten, müssen Sie diese nach dem Test selbst wieder herstellen.
- Selbstverständlich dürfen Sie auch Microsoft Visual Studio benutzen. Das abgegebene Programm muss aber unter Linux/g++ laufen. Sie dürfen auch Programme zum Remote Login auf `anubis` verwenden, sowie zum Kopieren von Dateien zwischen Windows und Linux.

- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Sie müssen Mindestanforderungen an den Programmierstil erfüllen, z.B. entsprechend der Programmstruktur einrücken.
- Fragen Sie, wenn Ihnen die Aufgabe nicht klar ist!
- Wenn Sie an einer unverständlichen Fehlermeldung länger festhängen, können Sie probieren, zu fragen. Wir wollen aber nicht zu viele Tipps geben.

Aufgabe (Variante A)

Entwickeln Sie ein Programm, das folgende Frage beantwortet: Angenommen, ich lege 1 Euro mit p Prozent Zinsen an, nach wieviel Jahren habe ich 1000 Euro?

Es soll zunächst der Text

```
Bitte Zinsatz eingeben:
```

ausgegeben werden (am Ende mit einem Leerzeichen, aber ohne Zeilenumbruch). Dann soll der Zinssatz eingelesen werden in eine Variable vom Typ `double` (Fliesskomma-Zahl). Wenn der eingegebene Zahlwert ≤ 0 ist, soll folgende Fehlermeldung ausgegeben werden:

```
Fehler bei der Eingabe!
```

(mit einem Zeilenumbruch am Ende). Das Programm soll sich dann beenden. Sie können davon ausgehen, dass eine syntaktisch korrekte Fließkomma-Zahl eingegeben wurde.

Der vorhandene Geldbetrag soll auch in einer Variablen vom Typ `double` gespeichert werden (wir ignorieren dabei, dass der Kontostand eigentlich immer ein voller Cent-Betrag sein muss). Wenn man x Euro auf dem Konto hat, und p Prozent Zinsen bekommt, erhöht sich der Kontostand nach einem Jahr um $x * (p/100.0)$.

Sie können das Anwachsen des Kontostands in einer Schleife simulieren, und sollen am Ende ausgeben, nach wie vielen Jahren der Geldbetrag ≥ 1000 ist. Es soll ein Text der folgenden Form ausgegeben werden (natürlich mit der richtigen Anzahl Jahre):

```
Es dauert 142 Jahre.
```

(mit einem Zeilenumbruch am Ende).

Beispiele für Tests:

- Bei einer negativen Eingabe muss die vorgegebene Fehlermeldung ausgegeben werden.
- Bei 100% Zinsen hat man nach einem Jahr 2 Euro, nach zwei Jahren 4 Euro, nach drei Jahren 8 Euro, u.s.w., bis man nach zehn Jahren mit 1024 Euro das erste Mal über der Grenze von 1000 Euro liegt. Das Programm muss dann also ausgeben:

  ```
  Es dauert 10 Jahre.
  ```

- Bei 5% Zinsen dauert es 142 Jahre.