Prof. Dr. Franz Schweiggert Institut für Angewandte Informationsverarbeitung 27. Juni 2008
Christoph Ott Blatt 9

Einführung in die Programmierung (SS 2008)

Abgabetermin: 03. Juli 2008

1 Zehn kleine Jägermeister (3P)

In den drei Beispielprogrammen *Jaegermeister1.java*, *Jaegermeister2.java* und *Jaegermeister3.java* wird die Methode *textausgabe* jeweils rekursiv aufgerufen. Erklärt Eurem Tutor was die Programme jeweils am Bildschirm ausgeben und wie diese Ausgabe zu Stande kommt.

2 Überraschungprogramm (2P)

Erklärt Eurem Tutor nun was das folgende Überraschungprogramm *WasAuchImmer.java* eigentlich tut und wie es die Rekursion nutzt.

3 Einfache Rekursionen programmieren (5P)

In der Vorlesung habt Ihr ein Programm gesehen, das in einer rekursiven Methode die Zahlen von *1* bis n aufsummiert hat und somit dem Programm *RekursiveSumme.java* sehr ähnlich war.

3.1 Rekursives Multiplizieren (2P)

Schreibt nun eine weitere rekursive Methode, die die Zahlen von *1* bis n multipliziert.

3.2 Fibonacci-Zahlen (2P)

Fibonacci-Zahlen sind folgendermaßen definiert:

- Die 0. Fibonacci-Zahl hat den Wert 0.
- Die 1. Fibonacci-Zahl hat den Wert 1.
- Größere Fibonacci-Zahlen lassen sich als Summe ihrer beiden Vorgänger darstellen.

Schreibt eine rekursive Methode, die die n-te Fibonacci-Zahl berechnet. Was Fibonacci-Zahlen mit der Vermehrung von Kaninchen zu tun haben, erfahrt Ihr in den Übungen.

3.3 Test der Fibonacci-Zahlen (1P)

An unserem Institut ist in letzter Zeit ein Tool entwickelt worden, das Programme automatisch auf Korrektheit testet. Reicht nun Euer Programm zur Berechnung der Fibonacci-Zahlen auf unserem Rechner mit dem Namen *turing*[1] mit folgendem Befehl ein:

```
submit prog 9 Rekursion.java
```

Wenige Sekunden später solltet Ihr detaillierte Informationen (auch per E-Mail) über das Testergebnis Eures Programms erhalten. Damit das funktionieren kann muss Euer Programm folgenden Ansprüchen genügen:

- Das Programm muss *Rekursion.java* heißen.
- Das Programm darf keinen Text ausgeben, sondern lediglich mit *Urc.getInt()* eine Zahl einlesen und das Ergebnis mit *Write.Int()* ausgeben. Abschließend muss Euer Programm mit *Write.Ln()* ein *newline* ausgeben. [2]
- Ihr könnt darüber hinaus den *submit*-Befehl um das Argument *team* erweitert. Dies setzt jedoch voraus, dass sich in Eurem aktuellen Verzeichnis eine Datei *team* befindet, in der sich in jeder Zeile genau ein Login eines an Eurem Team beteiligten Mitglieds befindet. Maximale Teamgröße hierbei ist 2. Beispielsweise könnten Herr Schweiggert und ich gemeinsam mittels

  ```
  submit prog 9 Rekursion.java team
  ```

  ein Programm einreichen, wenn die Datei *team* folgendermaßen aussieht:

  ```
  chrissch
  swg
  ```

Da dieses Tool im kommenden Sommer in größerem Stil eingesetzt werden soll, bitte ich Euch abschließend mir mitzuteilen, ob Ihr mit dem Tool klarkommt und ob irgendwelche Probleme auftreten. Danke.
Viel Erfolg!

[1] Wenn Ihr auf einem beliebigen unserer Rechner seid, so kommt Ihr mit *ssh turing* auf eben diese Maschine.
[2] Das Beispielprogramm *RekursiveSumme.java* hält sich an diese Vorgaben zu Ein- und Ausgabe und kann deshalb als Anhaltspunkt dienen.