

Übungsblatt 07

E-Learning

Absolvieren Sie die Tests bis Di., 11.12., 23:55 Uhr.

Die Tests sind in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegt.

Sie können einen Test **nur einmal durchlaufen**. Sobald Sie einen Test starten steht Ihnen nur eine **begrenzte Zeit** zu Verfügung, um den Test zu bearbeiten.

Alle Punkte, die Sie beim Test erreichen, werden ihnen angerechnet.

ILIAS 4-Minuten-Aufgaben – 12 Punkte

Absolvieren sie die Tests *Informatik I - ILIAS 07 Teil 1*, *Teil 2* und *Teil 3*.
(12 Punkte)

Übung

Abgabe bis Di., 11.12., 18 Uhr.

Werfen Sie Ihre Lösung in den Zettelkästen Ihrer Gruppenübung. Für die Übungen im Nordbereich stehen die Zettelkästen im Sockelgeschoß (Ebene -1) **oder** auf dem Flur vor dem Seminarraum auf Ebene 0 des Instituts für Informatik.

Wenn Ihre Übung im Südbereich stattfindet, klären Sie mit Ihrem Tutor wo die Lösungen abzugeben sind.

Achten Sie darauf, dass Ihr **Name**, Ihre **Gruppe** und Ihre **Matrikelnummer** auf **jedem** Blatt stehen!

Falls Ihre Lösung mehrere Blätter umfasst, heften Sie diese bitte zusammen.

Aufgabe 1 – 18 Punkte

Java. Reverse Engineering

Betrachten Sie folgendes Programmfragment.

```
b=true;
t=2;
while ((t*t <= x) && b) {
    b=((x%t) != 0);
    t++;
}
res = (b && (x !=1));
```

1. Ergänzen Sie die Variablendeklarationen.
(4 Punkte)
2. Warum ist für jede beliebige ganze Zahl x die Abbruchbedingung der Schleife irgendwann erfüllt?
(6 Punkte)
3. Beschreiben Sie die Funktion des Programmfragments unter der Bedingung, dass x eine natürliche Zahl mit $x > 0$ ist? D.h. welche Eigenschaften von x können aus dem Wert von **res** gefolgert werden?
(8 Punkte)

Aufgabe 2 – 18 Punkte

Umkehrung des Horner-Schemas

Der folgende Quelltext bestimmt die q -adische Darstellung der Dezimalzahl z und gibt die Ziffern in richtiger Reihenfolge (von links nach rechts) aus, durch Zwischenspeichern der Ziffern in einem `String` (vergleiche Skript Abschnitt 2.2-18).

```
String s = "";
do {
    s = (z % q) + s;
    z = z / q;
} while (z != 0);
System.out.println(s);
```

1. Schreiben Sie eine Java-Methode `hornerUmkehrungRekursiv(int z, int q)`, die die Umkehrung des Horner-Schemas benutzt und die Ziffern in richtiger Reihenfolge (von links nach rechts) ausgibt, ohne Zwischenspeichern der Ziffern in einem `String`. Nutzen Sie dazu die Idee rekursiv erst die q -adische Darstellung von z/q und dann die nachfolgende Ziffer auszugeben.
(10 Punkte)
2. Erstellen Sie ein rekursives Ablaufprotokoll für den Aufruf der Methode `hornerUmkehrungRekursiv(13, 2)`, setzen Sie anstelle des Rückgabewerts der Methode die bis zu diesem Zeitpunkt erfolgt Ausgabe.
(8 Punkte)

Aufgabe 3 – 17 Punkte

Java und Rekursion

Betrachten Sie folgende Applikation.

```
class Foo {  
    public static String ba(int n) {  
        String s = "";  
        int p;  
        for (p=2; p*p <= n; p++) {  
            if (n%p == 0) {  
                s = p + "*" + ba(n/p);  
                break; // verlassen der for - Schleife  
            }  
        }  
        if (p*p > n)  
            s = s + n;  
        return s;  
    }  
  
    public static void main(String[] args) {  
        int n = Integer.parseInt(args[0]);  
        System.out.println(ba(n));  
    }  
}
```

Angenommen `Foo.java` ist übersetzt und wird mit `java Foo 210` aufgerufen.

Erstellen Sie ein rekursives Ablaufprotokoll für den Aufruf der Methode `ba`. Sehen Sie für jeden rekursiven Aufruf eine Spalte für die aktuell ausgeführte Zeile und eine Spalte für jede lokale Variable vor. Tragen Sie in die Spalte für die aktuell ausgeführte Zeile den Java-Code dieser Zeile ein.

Wie sieht die Ausgabe auf der Konsole aus?

(17 Punkte)

Praktische Übung

Abgabe der Prüfsumme bis Di., 11.12., 23:55 Uhr.

Testat von Mi., 19.12., 8-10 Uhr bis Fr., 21.12., 18-20 Uhr.

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen, insbesondere in den Rechnerübungen am **Dienstag**, in denen **keine Testate** stattfinden.

Hinweise zu den praktischen Übungen, insbesondere zur **Abgabe der Prüfsumme** und zur **Durchführung der Testate**, sind in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien* → *Übungsblätter* hinterlegt.

Aufgabe 1 – 35 Punkte

Fraktale Kurven

Durch Zeichenketten, die nur die Buchstaben F, L und R enthalten, lassen sich fraktale Kurven beschreiben.

- F Bewege den Zeichenstift um die Länge 1 in die Richtung, in die der Zeichenstift orientiert ist und zeichne dabei eine Linie.
- L Drehe den Zeichenstift um 90° nach links.
- R Drehe den Zeichenstift um 90° nach rechts.

Drachenkurve

Die Zeichenkette $d_0 = \text{F}$ repräsentiert die Drachenkurve der Ordnung 0.

Die Zeichenkette d_{i+1} , der Drachenkurve mit Ordnung $i+1$, entsteht aus der Zeichenkette d_i , der Drachenkurve mit Ordnung i , indem an d_i erst R und dann d_i in umgekehrter Reihenfolge angehängt wird, wobei in der Umkehrung von d_i jedes L durch ein R und jedes R durch ein L ersetzt wird.

Beispiel

Ordnung 0: F

Ordnung 1: FRF

Ordnung 2: FRFRFLF

Ordnung 3: FRFRFLFRFRFLFLF

Lévy-C-Kurve

Die Lévy-C-Kurve wird zunächst mit den Zeichen F, +, - gebildet.

- + Drehe den Zeichenstift um 45° nach rechts.
- Drehe den Zeichenstift um 45° nach links.

Die Zeichenkette $c_0 = \text{F}$ repräsentiert die Lévy-C-Kurve der Ordnung 0.

Sei c_i die Zeichenkette der Lévy-C-Kurve mit Ordnung i , dann ist $c_{i+1} = + c_i - - c_i +$ die Zeichenkette der Lévy-C-Kurve mit Ordnung $i + 1$.

Um die Lévy-C-Kurve mit F, L und R zu repräsentiert, werden zuerst die Zeichen vor dem ersten und nach dem letzten F entfernt. Da sich zwischen zwei F immer eine gerade Anzahl sowohl an + als auch an - befindet, können diese durch L und R ersetzt werden (gleiche Anzahl von + und - hebt sich auf).

Beispiel

Ordnung 0: F

Ordnung 1: +F--F+ → FLF

Ordnung 2: ++F--F+---F--F++ → FLFFLF

Ordnung 3: +++F--F+---F--F+---+F--F+---F--F+++ → FLFFLFRFLFFLF

1. Schreiben Sie eine Applikation, die eine Zeichenkette ausgibt, die zufällig aus Buchstaben F, L und R zusammengesetzt wurde.

Setzen Sie folgende Anforderungen um.

- Die Länge der Zeichenkette k wird auf der Kommandozeile übergeben. Negative oder zu große Längen (> 100000) werden zurückgewiesen.
- Die Länge k wird an eine Methode übergeben, die k in einer Zeile ausgibt. In der nächsten Zeile gibt die Methode eine zufällige Anordnung der Buchstaben F, L und R aus, wobei die nötigen Zufallszahlen mit Hilfe der Funktion `StdRandom.uniform`, aus der in der Vorlesung eingeführten Bibliothek `stdlib.jar`, erzeugt werden.
- Versehen Sie die Klasse mit ausführlichen Kommentaren, die den Programmablauf erläutern.

(5 Punkte)

2. Erweitern Sie die Applikation aus Aufgabenteil 1, so dass auch eine Zeichenkette für die Drachenkurve der Ordnung k konstruiert und ausgegeben werden kann.

Setzen Sie folgende Anforderungen um.

- Auf der Kommandozeile wird ein Buchstaben und eine ganze Zahl erwartet. Der Buchstabe (**D**rachenkurve, **R**andom) bestimmt, was für eine Zeichenkette erzeugt wird, die ganze Zahl k ist die Ordnung der Drachenkurve oder die Länge der zufälligen Zeichenkette.
Ungültige Buchstaben, sowie negative oder zu große Ordnungen der Drachenkurve (> 20) und ungültige Längen (siehe Aufgabenteil 1) werden zurückgewiesen.

- Eine Drachenkurve der Ordnung k wird durch eine Zeichenkette der Länge $2^{k+1} - 1$ repräsentiert. Es wird ein Feld von `char` der passenden Länge erzeugt und an eine Methode übergeben, die die Repräsentation der Drachenkurve in dieses Feld hinein schreibt.
- Die Länge der Zeichenkette und die Zeichenkette selbst werden ausgegeben, jeweils in einer Zeile.
- Versehen Sie die Klasse mit ausführlichen Kommentaren, die den Programmablauf erläutern.

(10 Punkte)

3. Erweitern Sie die Applikation aus Aufgabenteil 2, so dass auch eine Zeichenkette für die Lévy-C-Kurve der Ordnung k konstruiert und ausgegeben werden kann.

Setzen Sie folgende Anforderungen um.

- Die Menge der gültigen Buchstaben (**L**évy-C), die auf der Kommandozeile übergeben werden können, wird erweitert.

Für die Ordnung k der Lévy-C-Kurve gelten dieselben Einschränkungen wie für die Ordnung der Drachenkurve.

- Benutzen Sie eine Methode, die in einem Feld von `char` eine Lévy-C-Kurve der Ordnung i übergeben bekommt und ein neues Feld zurückliefert, das eine Lévy-C-Kurve der Ordnung $i + 1$ repräsentiert.
- Benutzen Sie eine Methode, die in einem Feld von `char` eine Lévy-C-Kurve mit den Zeichen **F**, **+** und **-** übergeben bekommt und **+**, **-** wie oben beschrieben durch **L**, **R** ersetzt. Dabei soll kein neues Feld erzeugt und die **F-L-R**-Zeichenkette im übergebenen Feld zurückgeliefert werden.

Die Methode liefert die Länge der **F-L-R**-Zeichenkette zurück.

- Die Länge der **F-L-R**-Zeichenkette und die Zeichenkette selbst werden ausgegeben, jeweils in einer Zeile.
- Versehen Sie die Klasse mit ausführlichen Kommentaren, die den Programmablauf erläutern.

(20 Punkte)

Testen Sie Ihre Applikation mit der Lösung der Praktischen Übung 05 oder mit dem Java-Archiv `drawcurve.jar`, die in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien* → *Übungsblätter* → *Daten* hinterlegt ist.