Informatik I
Prof. Dr. Florin Manea
Stefan Siemer

Sommersemester 2023
Georg-August-Universität Göttingen
Institut für Informatik

Übungsblatt 07

E-Learning

Absolvieren Sie die Tests bis Di., 06.06., 10 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)

ILIAS Sterne-Aufgaben – 17 Punkte

Absolvieren Sie die Tests *Informatik I - ILIAS 07 Teil 4, Teil 5* und *Teil 6*.
(17 Punkte)

Übung

Abgabe bis Di., 06.06., 10 Uhr.

Markdown und AsciiMath

Die Lösungen dieser Aufgaben werden in Ihrer StudIP-Übungsgruppe mittels des VIP's Moduls oder in Absprache mit eurem Tutor/Tutorin getätigt.

Aufgabe 1 – 18 Punkte

Java. Reverse Engineering

Betrachten Sie folgende Methode `foo`.

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

Aufgaben

1. Vervollständigen Sie die Methode `foo`, indem Sie den Methodenkopf komplettieren und die Variablendeklarationen ergänzen.
   (4 Punkte)

2. Ist für jeden `int`-Wert `x` die Abbruchbedingung der Schleife irgendwann erfüllt? Kurze Argumentation.
   (6 Punkte)

3. Beschreiben Sie die Funktionalität der Methode `foo` unter der Bedingung, dass `x` ein `int`-Wert mit `x > 0` ist. D.h. welche Eigenschaften von `x` können aus dem Rückgabewert von `foo` 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, durch Zwischenspeichern der Ziffern in einem `String`, in richtiger Reihenfolge (von links nach rechts) aus. (Vergleiche Skript Kapitel 2.1.2)

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

1. Skizzieren 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)` in einer geeigneten Form (z.B in AsciiMath, Tabellen, oder mit Einrückungen ...). Versuchen Sie die Formattierung so zu wählen, dass es auch zum rechten Seitenrand auf eine Seite passt.

   Hinweis: Sie können sich mit entsprechenden Ausgaben eines ausführbaren Java-Programs Teile des Ablaufprotokoll generieren. .
   (8 Punkte)

Praktische Übung

Abgabe der Prüfsumme Di., 06.06., 10 Uhr.
Testat Mi., 07.06. bis Di., 13.06.2022.

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen.

- Erstellen Sie ein Archiv, dass **alle Dateien** enthält, die Sie beim Testat vorstellen möchten.
- Beim Testat werden nur Dateien aus einem Archiv testiert, dessen Prüfsumme **exakt** der von Ihnen übermittelten Prüfsumme entspricht.
- Berechnen Sie die Prüfsumme des Archivs mit dem **`sha1sum`** Befehl.
- Um die praktische Übung testieren zu lassen, **müssen** Sie einen Termin in der zugehörigen Rechnerübung reservieren. Ein Testat ohne Termin ist nicht möglich. Testate zu einer anderen praktischen Übung können nur in Ausnahmefällen nach Rücksprache mit Stefan Siemer durchgeführt werden.
- Übermitteln Sie die Prüfsumme durch Absolvieren des in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegten Tests *Informatik I - Testat 07.*
- Öffnen Sie die Terminvergabe für diese praktische Übung und lassen Sie den von Ihnen reservierten Termin anzeigen.
- Entpacken Sie das Archiv erst nachdem der Tutor die Prüfsumme kontrolliert hat.

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 $d_0 =$ `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 $c_0 =$ `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 Aufgabeteil 1, sodass 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 ein 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 Aufgabeteil 2, sodass 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**evy-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, sondern das übergebene Feld so bearbeitet werden, dass es die `F-L-R`-Zeichenkette enthält.

  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 vorigen Praktischen Übung oder mit dem Java-Archiv `drawcurve.jar`, das in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien→Übung→uebung07-data* hinterlegt ist.