

Übung 6

Abgabe bis Dienstag, 10.12., 18:00 Uhr.

Werfen Sie Ihre Lösung in den Zettelkästen Ihrer Gruppenübung. Für die Übungen im Nordbereich stehen die Zettelkästen im Sockelgeschoß (Ebene -1) 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.

LON-CAPA – 34 Punkte

IEEE 754, Dezimal-/Binärbruch, Schleifen

Absolvieren Sie in den, in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegten, Test *Informatik I - LON-CAPA* die *Übung 06*. Beachten Sie folgende Rahmenbedingungen.

- Sie müssen den Test bis 20:00 Uhr am Abgabetag dieses Übungszettels absolviert haben.
- Sie können den Test nur einmal durchlaufen.
- Sobald Sie den Test starten bleiben Ihnen 3 Stunden Zeit den Test zu bearbeiten.
- Alle Punkte, die Sie beim Test erreichen, werden ihnen angerechnet.

(34 Punkte)

Aufgabe 1 – 8 Punkte

Java. Type Casting

Betrachten Sie folgende ausführbare Klasse.

```
public class TypeCasting {
    public static void main(String[] args) {
        short a=73;
        short b=219;
        short c=(short) (a+b);
        int d=(byte) c;
        short e=-3;
        int f=(char) e;
        int g=1;
        int h=4;
        double i=g/h*4.0;

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(d);
        System.out.println(e);
        System.out.println(f);
        System.out.println(g);
        System.out.println(h);
        System.out.println(i);
    }
}
```

1. Welche Ausgabe produziert die `main`-Methode und warum.
(6 Punkte)
2. Warum verlangt der Java-Compiler in Zeile 5 ein *type casting*.
(2 Punkte)

Aufgabe 2 – 33 Punkte

Binärcode

Es sei $a = (a_0, a_1, \dots, a_{n-1})$ eine n -stellige Binärziffernfolge mit $a_i \in \{0, 1\}$. Die Ableitung von a ist die n -stellige Binärziffernfolge $b = (b_0, b_1, \dots, b_{n-1})$ mit $b_i \in \{0, 1\}$, deren Ziffern folgendermaßen entstehen.

$$\text{Für } i = 0, \dots, n-1 \text{ gilt} \quad b_i = \begin{cases} 0 & \text{falls } i = 0 \text{ und } a_i = 0 \\ 0 & \text{falls } i > 0 \text{ und } a_i = a_{i-1} \\ 1 & \text{sonst} \end{cases}$$

1. Was stellt die Ableitung da? D.h. wenn die Ableitung gegeben ist, welche Aussagen kann man dann über die abgeleitete Binärziffernfolge treffen?
(5 Punkte)

2. Geben Sie Java-Code einer rekursiven `void`-Methode an, die die Ableitung einer Binärziffernfolge bestimmt.

Unter den Parametern der Methode befindet sich genau ein Feld mit Binärziffern. Dieses Feld wird mit jedem Methodenaufruf weitergereicht und während der Abarbeitung des Algorithmus mit der Ableitung überschrieben. Die Methode verwendet, neben dem übergebenen Feld, kein weiteres Feld.

Wie sieht ein Aufruf der Methode aus?

(10 Punkte)

Hinweis. Durchlaufen Sie das Feld vom a_{n-1} nach a_0 .

3. Formulieren Sie eine Vorschrift für die Umkehrung der Ableitung, d.h. eine Vorschrift, die angibt, wie man aus einer gegebenen Ableitung die abgeleitete Binärziffernfolge bestimmt.
(8 Punkte)

4. Geben Sie Java-Code einer iterativen `void`-Methode an, die die Umkehrung einer Ableitung bestimmt, entsprechend der Vorschrift aus Aufgabenteil 3.

Die Methode arbeitet auf der Eingabe, d.h. die übergebene Binärziffernfolge wird während der Abarbeitung des Algorithmus mit der Umkehrung überschrieben. Die Methode verwendet, neben der Eingabe, kein weiteres Feld.

(10 Punkte)

Hinweis. Durchlaufen Sie das Feld vom a_0 nach a_{n-1} .

Praktische Übung F

Testat bis Donnerstag, 13.12., 20.00 Uhr.

Die Punkte, die Sie erreicht haben werden auf diesem Blatt notiert, geben Sie es beim Tutor Ihrer Gruppenübung ab, sobald diese feststeht.

Lassen Sie Ihre Lösung während einer **Rechnerübung** testieren, dazu **müssen** Sie sich über Stud.IP einen **Termin reservieren**. Den Link zur Terminvergabe finden Sie in den Ankündigungen der Stud.IP-Veranstaltung Informatik I. **Pro Tag ist aber nur eine begrenzte Anzahl an Testaten möglich.**

Testieren einer Aufgabe bedeutet, dass Sie dem Tutor die Aufgabe Schritt für Schritt demonstrieren können und auch in der Lage sind auf die Lösung bzw. den Lösungsweg bezogene Fragen zu beantworten.

Bereiten Sie sich auf das Testat vor. Das Testat eine Aufgabe kann vorzeitig beendet und ein entsprechender Punktabzug vorgenommen werden, wenn Sie z.B. auf in der Aufgabenstellung formulierte Fragen keine Antworten vorbereitet haben. Beginnen Sie z.B. erst während des Testats mit der Bearbeitung der Aufgabe wird das Testat abgebrochen und Sie bekommen für die Aufgabe keine Punkte.

Hilfe zum Bearbeiten der praktischen Übung können Sie grundsätzlich jeden Tag in den Rechnerübungen (18-20 Uhr im Rechner-Pool) bekommen.

Studentin/Student

Name, Vorname	
Matrikelnummer	
Stud.IP-Benutzername	

Testat

Aufgabe F.1	25	
-------------	----	--

Datum, Name, Unterschrift

Benutzung

```
> java Hint
15
FLFLFRFLFLFRFRF
```

- Dem eigentlichen Zeichnen geht eine Simulation voran, die die Größe des Polygonzugs ermittelt, d.h. die maximale/minimale x-/y-Koordinaten bestimmt, und die Skalierung des angezeigten Koordinatensystems so festlegt, dass der komplette Polygonzug sichtbar ist.
- Der Polygonzug wird mit Hilfe der Methoden aus der im Skript vorgestellten Klasse `StdDraw` gezeichnet.
- Versehen Sie die Klasse mit ausführlichen Kommentaren, die den Programmablauf erläutern.

(18 Punkte)

3. Testen Sie die ausführbare Klasse aus Aufgabenteil 2. mit dem Java-Archiv `dragon.jar`, das in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien* → *Vorlesung* hinterlegt ist.

(4 Punkte)

Hinweis

Benutzen Sie den *Manualbrowser* `man`, um zu ermitteln wie man mit dem *Java Application Launcher* `java` ein Java-Archiv ausführt.