

Einführung in die Programmierung

Übungsblatt 5

Hinweise

- Erstellen Sie bitte eine schriftliche Lösung der nachfolgenden Aufgaben und erzeugen Sie daraus eine pdf-Datei. Sie können auch Ihre (lesbare!) handschriftliche Lösung als pdf einscannen. Achten Sie bitte darauf, dass am Beginn der ersten Seite des pdfs Ihr *Name*, Ihre *Matrikelnummer* und der Name Ihres *Tutors* (Laborübungsgruppe) vermerkt ist. Jede Aufgabe, die durch **(Laborübung)** gekennzeichnet ist, beinhaltet als Lösung auch ein Java-Programm als Quelltextdatei. Der Quelltextdateiname muss dem Namen der *Klasse* entsprechen, die die `main`-Methode enthält. Bitte packen Sie Ihr pdf gemeinsam mit allen Java-Quelltextdateien (nur `*.java`, keine `*.class`) in eine zip-Datei, die Sie termingemäß im moodle hochladen.
- Sie bekommen eine Bewertung Ihrer Lösung mit der Information, ob sie ausreichend zum Bestehen des Übungsblattes ist. Im Falle einer nichtausreichenden Lösung haben Sie Gelegenheit, eine verbesserte Version zeitnah, aber spätestens bis 15.07.2016, im moodle nachzureichen. Plagiate werden nicht als ausreichende Lösungen anerkannt.

Aufgabe 1

(Laborübung) Der fiktive Lebensmittelhersteller *ProAdipo* produziert Flaschenketchup. Jede Ketchupflasche soll idealerweise eine Inhaltsmasse von 450 Gramm besitzen. Aus technischen Gründen variiert jedoch die Füllmenge in den einzelnen Flaschen. Der Wiegeautomat hat in einem Produktionszeitraum folgende 12 Inhaltsmassen (in Gramm) gemessen: 451.23, 449.71, 449.85, 450.02, 448.79, 450.11, 447.39, 450.28, 450.94, 448.20, 446.98 und 451.19.

Schreiben Sie ein Java-Programm, das die einzelnen Massen in einem Feld erfasst sowie die minimale Masse, die maximale Masse und die durchschnittliche Masse (arithmetisches Mittel) bestimmt und ausgibt. Flaschen, deren Inhaltsmasse um mehr als 0,5 Prozent unter 450 Gramm liegt, dürfen nicht in den Handel gebracht werden. Erweitern Sie Ihr Java-Programm um eine Methode, die die Maßzahlen auf Einhaltung der Mindestfüllmenge prüft und die Inhaltsmassen zu leichter Flaschen in ein gesondertes Feld kopiert, das dann für weitere Auswertungen bereitgestellt und angezeigt wird.

Aufgabe 2

Auf der Lehrveranstaltungswebseite finden Sie die Datei `Kalenderdatum.java` zum Download. Die entsprechende Klasse `Kalenderdatum` enthält als Attribut ein `Kalenderdatum` durch Angabe von Tag, Monat und Jahr im *Gregorianischen Kalender*. Bei Anlegen eines neuen Objekts dieser Klasse wird über den Konstruktor ein konkretes `Kalenderdatum` erfasst, wobei nur gültige Kalenderdaten ab 01.01.1583 aufgenommen werden.

In der Klasse stehen verschiedene Methoden bereit. Die Methode `bestimmeWochentag` liefert den zugehörigen Wochentag als Zeichenkette (z.B. "Sonntag") auf Basis der *Gaußschen Wochentagsformel*. Darüber hinaus gibt es Methoden, die abtesten, ob das gespeicherte Datum ein bestimmter bundeseinheitlicher gesetzlicher Feiertag ist. Beispielsweise liefert die Methode `istOstersonntag` genau dann den booleschen Wert `true` zurück, wenn das hinterlegte Kalenderdatum mit diesem Feiertag übereinstimmt. Als bundeseinheitliche gesetzliche Feiertage sind in Deutschland seit 1949 festgelegt: Neujahr, Karfreitag, Ostersonntag, Ostermontag, Tag der Arbeit, Christi Himmelfahrt, Pfingstsonntag, Pfingstmontag, Tag der Deutschen Einheit (seit 1990 am 03.10., zwischen 1954 und 1990 am 17.06., vor 1954 kein Feiertag), 1. Weihnachtsfeiertag und 2. Weihnachtsfeiertag. Darüber hinaus enthält die Klasse die Methoden `istGueltig`, `gibTag`, `gibMonat`, `gibJahr` und `toString`.

- **(Laborübung)** Schreiben Sie eine eigene Testklasse, welche die Klasse `Kalenderdatum` verwendet. Finden Sie damit die Wochentage folgender denkwürdiger Daten heraus:
 20.07.1969 Landung des ersten Menschen auf dem Mond
 09.11.1989..... Fall der Berliner Mauer
 03.10.1990 Tag der Deutschen Wiedervereinigung
 01.01.2002 Einführung des Euro-Bargeldes
 Tag Ihrer Geburt (optional – Bitte nur den Wochentag aufschreiben, nicht das Datum)
- **(Laborübung)** Erweitern Sie Ihre Testklasse, so dass folgende Fragen beantwortet werden:
 In welchem Jahr zwischen 2000 und 2016 fielen der *Tag der Arbeit* und *Christi Himmelfahrt* auf das gleiche Kalenderdatum?
 Wieviele bundeseinheitliche gesetzliche Feiertage liegen im Jahr 2016 sonntags?
- **(Laborübung)** Erzeugen Sie in Ihrer Testklasse eine Ausgabe, die die Woche (7 Tage) ab einem vom Nutzer einzugebenden gültigen Startdatum (z.B. 26.12.2016) im Sinne eines Wochenkalenderblattes darstellt und zu jedem Tag den Wochentag sowie enthaltene Feiertage mit anzeigt. Nutzen Sie zur Bereitstellung der Daten die Klasse `Kalenderdatum`.

Aufgabe 3

Zwischen etwa 20 und 25 Prozent der Rechenzeit von Großrechenanlagen entfallen nach einer Studie der Stanford University auf das Ausführen von Sortiervorgängen. Sortieren dient bei weitem nicht nur dazu, Tabellen oder Ergebnisdaten für den Nutzer leicht erschließbar darzustellen, sondern ist Bestandteil zahlreicher Algorithmen wie zum Beispiel als Vorstufe der Suche in Datenbeständen oder bei Algorithmen, die ein Problem dadurch lösen, dass sie eine Vielzahl potentieller Lösungskandidaten raten und dann jeden dieser Lösungskandidaten ob seiner Tauglichkeit und Qualität als Lösung bewerten.

(Laborübung) Implementieren Sie das Sortierverfahren *SelectionSort* in Java. Die zu sortierenden Daten (in unserem Fall ganze Zahlen) werden in einem `long`-Feld bereitgestellt, das entsprechend initialisiert ist. Die Daten sollen *aufsteigend* sortiert werden.

(Laborübung) Testen Sie Ihr Programm in vier verschiedenen Fallstudien:

Fallstudie 1: 17, 11, 8, 2, 15, 4, 8, 10, 9, 14

Fallstudie 2: 17, 15, 14, 11, 10, 9, 8, 8, 4, 2

Fallstudie 3: 17, 2, 4, 8, 8, 9, 10, 11, 14, 15

Fallstudie 4: 2, 4, 8, 8, 9, 10, 11, 14, 15, 17

Bis auf wenige Fälle ist bei *SelectionSort* die Anzahl Vertauschungen bzw. Verschiebungen von Feldelementen deutlich niedriger als bei anderen einfachen Sortierverfahren wie *InsertSort* und *BubbleSort*, hingegen werden aber zumeist mehr Vergleiche vorgenommen. Für welche Anwendungsszenarien bietet sich *SelectionSort* daher an?

Aufgabe 4

Will man natürliche Zahlen sortieren, deren Werte eng beieinander liegen, so bietet sich das Verfahren *BucketSort* an. Es hat die Besonderheit, dass keinerlei Vergleiche zwischen den einzelnen Zahlen notwendig sind und besticht durch Schnelligkeit (nur etwa $2n$ Arbeitsschritte bei n zu sortierenden Zahlen) sowie algorithmische Eleganz, ist aber im praktischen Einsatz bei großer Streubreite zwischen den zu sortierenden Zahlen ungeeignet und benötigt üblicherweise viel Hilfsspeicherplatz. Hinter *BucketSort* steht die Idee des Fachverteils, wie sie traditionell bei der Briefzustellung in Verteilzentren der Post etabliert wurde.

```
public class Bucketsort {
    public static int N; // Anzahl Buckets

    public static void sort (int[] a) { // sortiere Feld a
        int[] b = new int[N]; // N Buckets
        int i, j, k; // Laufvariablen

        for (i=0; i < N; i++) {
            b[i] = 0; // setze alle Buckets auf 0
        }
        for (i=0; i < a.length; i++) { // fuer jedes Feldelement
            b[a[i]]++; // zustaendiges Bucket erhoehen
        }

        k = 0;
        for (i=0; i < N; i++) { // fuer jedes Bucket
            for (j=0; j < b[i]; j++) { // genaess Zaehlerstand
                a[k++] = i; // seinen Index uebernehmen
            }
        }
    }

    public static void main (String[] args) {
        int[] intArr = { 16, 23, 14, 7, 21, 20, 6, 1, 17, 13, 12, 9, 3, 19 };

        N = intArr[0]; // groesstes Elem im Feld finden
        for (int m = 0; m < intArr.length; m++) {
            if (N < intArr[m]) {
                N = intArr[m];
            }
        }
        N++;
        sort(intArr); // BucketSort aufrufen
        for (int m = 0; m < intArr.length; m++) { // sortiertes Feld ausgeben
            System.out.print(intArr[m] + " ");
        }
        System.out.print("\n");
    }
}
```

- **(Laborübung)** Implementieren Sie *BucketSort* in Java unter Benutzung der umseitigen Quelltextvorlage.
- Wieviele Buckets (Fächer) werden zum Sortieren des Feldes `intArr` im Quelltextbeispiel benötigt?
- Welchen Inhalt hat das Bucket-Feld `b`, nachdem es vollständig belegt ist?
- **(Laborübung)** Modifizieren Sie den Quelltext bitte so, dass *absteigend* statt *aufsteigend* sortiert wird, aber ohne eine längere Programmlaufzeit hervorzurufen.