Universität der Bundeswehr München
Elektrotechnik und Technische Informatik
Oswald / Baumann

Wintertrimester 2021
Praktikum 04
Abgabe: 08.02.-09.02.2021

Grundlagen der Programmierung

Programmieren mit Feldern

Aufgabe 1: Debuggen

Gegeben ist folgendes Java-Programm. Die Klassenmethode `ersetze` bekommt als Parameter einen Parameter vom Typ `char[]` (1. Parameter) und zwei Parameter vom Typ `char` (2. und 3. Parameter) übergeben. Die Klassenmethode ersetzt im `char`-Array alle Buchstaben, die dem 2. Parameter entsprechen durch die Buchstaben des 3. Parameters. Das Ergebnis wird in einem `char[]` zurückgegeben.

```
import java.util.Arrays;

public class Buchstaben {
    public static void main(String[] args) {
        char[] eingabe = new char[] { 'K', 'a', 's', 's', 'e' };
        char[] ergebnis = ersetze(eingabe, 'K', 'T');

        // Ausgabe der Eingabe
        System.out.println("Eingabe: " + Arrays.toString(eingabe));
        // Ausgabe des Ergebnis
        System.out.println("Ergebnis: " + Arrays.toString(ergebnis));
    }

    // Klassenmethode zum Ersetzen von Buchstaben
    public static char[] ersetze(char[] a, char suche, char ersetze) {
        for (int i = 0; i < a.length; i++) {
            // Ersetzen der Buchstaben
            if (a[i] == suche)
                a[i] = ersetze;
        }
        return a;
    }
}
```

a) Kopieren Sie sich das Programm nach Netbeans und führen Sie das Programm aus. Was fällt Ihnen auf? Warum gibt die Anweisung
`System.out.println("Eingabe: " + Arrays.toString(eingabe));`
`Eingabe: [T, a, s, s, e]` und nicht `Eingabe: [K, a, s, s, e]` aus?
Lassen Sie das Programm schrittweise im Debugger ablaufen.

b) Wie müssen Sie die Klassenmethode `ersetze` verändern, damit die Ausführung der Klasse `Buchstaben` die folgende Ausgabe erzeugt?
`Eingabe: Kasse`
`Ergebnis: Tasse`
Sie dürfen ausschließlich die Klassenmethode `ersetze` anpassen! Sie dürfen **nicht** die Klassenmethode `main` verändern und die Klassenmethoden der Klassen `System` und `Arrays` **nicht** verwenden.

Aufgabe 2: Operationen auf Feldern

In dieser Aufgabe sollen Sie einige Klassenmethoden für Felder implementieren. Es gibt schon einige Klassenmethoden für Felder, die Ihnen Java zur Verfügung stellt. Zur Umsetzung der Klassenmethoden der Teilaufgaben b) - e) dürfen Sie diese Klassenmethoden allerdings nicht verwenden.

a) Schauen Sie sich die Java API der Klasse `Arrays` und `System` an. Mit welchen Klassenmethoden können Sie Felder kopieren? Mit welchen Klassenmethoden können Sie Felder in einen `String` umwandeln? Welche Klassenmethoden für Felder werden Ihnen noch zur Verfügung gestellt?

b) Schreiben Sie eine nicht-rekursive Klassenmethode
`public static double getMaximum(double[] a)`,
die den maximalen Wert, der im Feld vorkommt als Ergebnis zurückliefert.

c) Schreiben Sie eine nicht-rekursive Klassenmethode
`public static double getEvenAverage(int[] a)`,
die den Mittelwert aller im Feld vorkommenden geraden Zahlen berechnet.

d) Schreiben Sie eine nicht-rekursive Klassenmethode
`public static char[] revert(char[] a)`,
die ein neu erzeugtes Feld zurückgibt, in das die Werte aus dem Feld `a` in umgekehrter Reihenfolge kopiert wurden.

e) Schreiben Sie eine nicht-rekursive Klassenmethode
`public static boolean isSorted(long[] a)`,
die feststellt, ob das übergebene Feld aufsteigend sortiert ist.

f) Testen Sie alle Methoden. Welche Bedingungen für die Eingabeparameter müssen erfüllt sein, damit die Klassenmethoden ein gültiges bzw. sinnvolles Ergebnis liefern?

Aufgabe 3: Matrizenmultiplikation

Eine $l \times m$-Matrix A ist eine rechteckige bzw. tabellarische Anordnung von Zahlen, in l-Zeilen und m-Spalten

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,m} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{l,1} & a_{l,2} & \ldots & a_{l,m} \end{pmatrix}$$

In Java können wir diese Matrix in einem zweidimensionalen Feld bzw. Array speichern: `double[][] a = new double[l][m]`. Den Wert $a_{i,j}$ mit $1 \leq i \leq l$ und $1 \leq j \leq m$ speichern wir dann an der Position `a[i-1][j-1]`.

a) Schreiben Sie eine öffentliche Klassenmethode
`public static boolean isMatrix(double[][] a, int l, int m)`
die überprüft, ob die eingegeben Matrix `a` eine gültige `l`×`m`-Matrix ist. Die Klassenmethode soll `false` liefern, wenn `l` oder `m` kleiner gleich `0` ist oder wenn die Matrix `a` keine vollständige `l`×`m`-Matrix ist.

Testen Sie Ihre Klassenmethode, indem Sie sich überlegen, welche Parameter an die Klassenmethode `isMatrix` übergeben werden können. Denken Sie dabei auch an die folgenden Eingabem Möglichkeiten für `a`: `null`, "jagged arrays" oder zu große oder ungültige Werte in den Feldern. Schauen Sie sich dazu die in der Klasse `Double` definierten Konstanten an.

b) Schreiben Sie eine öffentliche Klassenmethode
`public static void printMatrix(double[][] a)`
die eine gültige Matrix in tabellarischer Anordnung ausgibt.

c) In dieser Aufgabe sollen Sie eine öffentliche Klassenmethode
`public static double[][] multMatrices (double[][] a, double[][] b)`
schreiben, die die Matrix A und die Matrix B als Eingabeparameter erhält und das Produkt der beiden Matrizen $(A \cdot B)$ berechnet.

Mathematische ist die Matrizenmultiplikation wie folgt definiert: Zwei Matrizen können nur dann multipliziert werden, wenn A eine $l \times m$-Matrix und B eine $m \times n$-Matrix ist, d.h. wenn die Anzahl der Spalten der ersten Matrix gleich der Anzahl der Zeilen in der zweiten Matrix ist. Das Ergebnis $C = A \cdot B$ ist dann eine $l \times n$-Matrix. Die Werte der Ergebnismatrix ergeben sich wie folgt:

$$c_{i,j} = \sum_{k=1}^{m} a_{i,k} \cdot b_{k,j} \text{ mit } 1 \leq i \leq l \text{ und } 1 \leq j \leq n$$

Die Klassenmethode `multMatrices` soll `null` zurückgeben, wenn die Eingabeparameter fehlerhaft sind bzw. nicht multipliziert werden können.