Programmieren und Software-Engineering II
Übung 0

Name: _____ **Klasse:** _____ **Datum:** _____

Lernziele:

- Wiederholung

1. **Arrayprüfung – eindimensionales Array**
   Für ein Array soll festgestellt werden, ob jede Zahl größer ist, als das doppelte der vorherigen. Erstelle eine Methode, die diese Funktion bietet.
   Prüfe das Programm für folgende beiden Arrays:
   arr1 = {1,5,14,27,60}
   arr2 = {1,10,100,1000,10000}

2. **Zweidimensionale Arrays**
   Erstelle alle notwendigen Methoden, für das angegebene Hauptprogramm. Achte dabei darauf, dass du keine Änderungen am Hauptprogramm vornehmen darfst, insbesondere auch keine Änderungen der Parameter und Rückgabewerte. Du kannst aber beliebig noch zusätzliche eigene Methoden verwenden.

   Punkteverteilung:
   fillTwoDim
   outTwoDim
   shiftLeft
   deletePrim

```
public static void main(String[] args) {

    int[][] twoDim = new int [10][10];
    int lower = 1;
    int upper = 50;

    //Fuellen des Arrays mit Zufallszahlen zwischen 1 und 50
    fillTwoDim(twoDim,lower,upper);

    //Ausgeben des Arrays
    System.out.println("\n Zweidimensionales Array");
    outTwoDim(twoDim);

   //Verschieben der Spalten nach links, d.h. in jeder Zeile wird der erste
   //Wert an die letzte Stelle gesetzt, alle anderen wandern eine Position
   //nach links
    shiftLeft(twoDim);

    //Ausgabe
    System.out.println("Erste Spalte wurde zur letzten");
    outTwoDim(twoDim);

    //Loeschen aller Primzahlen im Array durch Ersetzen mit 0
    int count = deletePrim(twoDim);

    System.out.println("Es wurden " + count + " Primzahlen geloescht");

    //Ausgabe
    System.out.println("Array ohne Primzahlen");
    outTwoDim(twoDim);
```

3. **Vergleich von eindimensionalen Arrays**
   Das Vergleichen, oder Pattern-Matching, ist in der Informatik eine wesentliche Grundfunktion. Dabei können für den Vergleich verschiedene Kriterien gelten. Implementiere ein Java-Programm,

das für jeden der folgenden Vergleiche eine Methode hat. Der Rückgabewert ist immer boolean und gibt an, ob der Vergleich positiv verlaufen ist oder nicht. Teste deine Methoden mit sinnvollen Testdaten.

Vergleich1
Die beiden eindimensionalen Arrays gelten als gleich, wenn an jeder Stelle des arr1 der gleiche Wert wie in arr2 steht. Natürlich müssen die beiden auch gleich lang sein.

Vergleich2
Die beiden eindimensionalen Arrays gelten als gleich, wenn jede Zahl, die in arr1 vorkommt, auch in arr2 vorkommt, egal an welcher Stelle und wie oft. Allerdings muss die Bedingung auch umgekehrt stimmen, d. h. auch jede Zahl, die in arr2 vorkommt, muss in arr1 vorkommen. Die Arrays können dabei unterschiedlich lang sein.

4. Anwendungsbeispiel

Schreibe ein Java-Programm, das das Wartezimmer eines Arztes mit Hilfe eines eindimensionalen int-Arrays simuliert.

Dieses Wartezimmer hat 25 Sesseln, die durch ein entsprechend langes Array simuliert werden. Im ersten Schritt werden diese mit 60%iger Wahrscheinlichkeit mit Patienten besetzt werden. Jeder Patient wird dabei durch seine Priorität (=Wichtigkeit der Behandlung) dargestellt. Die Prioritäten sind dabei zufällig zwischen 10 und 1 zu wählen, 0 bedeutet, dass der Sessel unbelegt ist.

Patienten mit niedriger Priorität müssen zuerst behandelt werden, solche mit hoher können warten.

Sortiere daher das Array aufsteigend, die leeren Sesseln (0-Priorität) müssen jedoch am Ende des Arrays sein.

Zusätzlich müssen noch folgende Aktionen abgebildet werden:

- ein Patient wird behandelt: Das heißt das erste Element des Arrays wird gelöscht, alle anderen rutschen um eine Position nach vorne. Die Priorität des zu behandelnden Patienten wird als Rückgabewert verwendet.
- ein neuer Patient kommt: Wenn nicht das gesamte Array bereits belegt ist, wird ein neues Element wird an der richtigen Stelle im sortierten Arrays eingefügt.

Implementiere zur Lösung des Problems zum Beispiel zumindest folgende Methoden:
- public static void fill (int[] p_arr, int p_up, int p_low)
- public static void sort(int[] p_arr)
- public static int delete(int[] p_arr)
- public static void fillIn(int[] p_arr, int value)

Lösungshinweis:

Um eine 60%ige Wahrscheinlichkeit zu ermitteln kann folgendes Prinzip angewendet werden:

- Erstellen einer Zufallszahl zwischen 1 und 100
- Ist die Zufallszahl kleiner-gleich 60 so wird der Stuhl besetzt.
- Ist die Zufallszahl größer als 60 so bleibt der Stuhl leer.