Java programmieren: Konsolen-Aufgaben

Aufgabe 1: Gefüllte Rechtecke zeichnen

Schreiben Sie ein Programm, das ein durch Sternchen gefülltes Rechteck zeichnet. Der Benutzer soll Breite und Höhe eingeben können:

```
Geben Sie die Breite des Rechtecks ein: 20
Geben Sie die Höhe des Rechtecks ein:   5
********************
********************
********************
********************
********************
```

Der Anfang ist bereits gemacht:

```
public class SternchenRechteckGefuellt {

	public static void main(String[] args) throws IOException {
		final BufferedReader konsolenEingabe = new BufferedReader(
				new InputStreamReader(System.in));
		System.out.print("Geben Sie die Breite des Rechtecks ein: ");
		final int breite=Integer.parseInt(konsolenEingabe.readLine());
		System.out.print("Geben Sie die Höhe des Rechtecks ein:   ");
		final int hoehe = Integer.parseInt(konsolenEingabe.readLine());

		// hier fehlt das Hauptprogramm
	}

	// hier können Sie eigene Methoden ergänzen

}
```

Ergänzen Sie das Programm (Hauptprogramm sowie benötigte Methoden), damit das Programm ein gefülltes Rechteck zeichnet.

Aufgabe 2: Rechteck-Umrandung zeichnen

Schreiben Sie ein Programm, das ein durch Sternchen umrandetes Rechteck zeichnet. Der Benutzer soll Breite und Höhe eingeben können:

```
Geben Sie die Breite des Rechtecks ein: 20
Geben Sie die Höhe des Rechtecks ein:   4
********************
*                  *
*                  *
********************
```

Sie können dazu das gleiche Gerüst wie in Aufgabe 1 verwenden.

Hinweis: Ein umrandetes Rechteck braucht für die Darstellung mindestens eine Breite und Höhe von 2 Zeichen. Ihr Programm sollte die Benutzereingaben daraufhin überprüfen.

Sie können Ihr Programm auch variieren: Horizontale Linien mit – darstellen, vertikale mit |. Und wenn Sie möchten, etwas anspruchsvoller, Ecken mit +.

Aufgabe 3: Unterschiede zwischen zwei Texteingaben zählen

Ein häufiges Problem ist der Vergleich von zwei Dateien: Wo unterscheiden Sie sich? Betrachten wir hier die Aufgabe, die Anzahl Unterschiede zwischen zwei eingegebenen Texten zu zählen. Der Anfang ist wiederum bereits gemacht:

```
public class AnzahlUnterschiede {

	public static void main(String[] args) throws IOException {
		final BufferedReader konsolenEingabe = new BufferedReader(
				new InputStreamReader(System.in));
		final String eingabe1 = konsolenEingabe.readLine();
		final String eingabe2 = konsolenEingabe.readLine();

		int anzahlUnterschiede = 0;

		// hier fehlt das Hauptprogramm

		System.out.println("Anzahl Unterschiede: " +
				anzahlUnterschiede);
	}

}
```

Ergänzen Sie das Hauptprogramm, so dass die Anzahl Unterschiede zwischen eingabe1 und eingabe2 ermittelt wird.

Sie können eine erste Version Ihres Programmes unter der Annahme schreiben, dass beide Eingaben gleich lang sind.

Erweitern Sie dann Ihr Programm, so dass es auch mit Eingaben unterschiedlicher Längen zurecht kommt.

Aufgabe 4: Berechnungen auf eingegebenen Zahlen

Schreiben Sie ein Programm, das eine Liste von Zahlen – getrennt durch einzelne Leerzeichen – entgegennimmt und die Summe sowie den Mittelwert der Zahlen ausgibt. Der Anfang ist wiederum bereits gemacht; der Befehl *eingabe.split(" ")* trennt die Eingabe bei den Leerzeichen auf und gibt die einzelnen Bestandteile in einem Array zurück:

```
public class ZahlenAddieren {

	public static void main(String[] args) throws IOException {
		final BufferedReader konsolenEingabe = new BufferedReader(
				new InputStreamReader(System.in));
		final String eingabe = konsolenEingabe.readLine();
		final String[] zahlen = eingabe.split(" ");

		// hier fehlt das Hauptprogramm
	}

}
```

Wenn Sie möchten, können Sie die Zahlen auch einzeln einlesen und zum Beispiel bei einer Eingabe von 99999 oder so die Eingabe abbrechen.

Aufgabe 5: Doppellautanalyse

Schreiben Sie ein Programm, dass die Anzahl Diphthonge (siehe auch http://de.wikipedia.org/wiki/Diphthong) in der Eingabe zählt. Wir betrachten hier die Diphthonge au, ei, ai. Das Gerüst des Programms ist wie bei den anderen Aufgaben gegeben:

```
public class DiphthongeZaehlen {

	public static void main(String[] args) throws IOException {
		final BufferedReader konsolenEingabe = new BufferedReader(
				new InputStreamReader(System.in));
		final String eingabe = konsolenEingabe.readLine();

		int anzahlDiphtonge = 0;

		System.out.println("Anzahl Diphthonge: " + anzahlDiphtonge);
	}


}
```

Aufgabe 6: Vokale filtern

Schreiben Sie ein Programm, das alle Vokale aus der Eingabe filtert, also nur Nicht-Vokale wieder ausgibt; die Anzahl gefilterter Vokale soll ebenfalls ausgegeben werden:

```
Wie erleben die Menschen in den Strassen Kairos den Kampf um ihre Freiheit?
Eine junge Lehrerin erzählt tagebuchartig von ihren Besuchen auf dem
Tahrir-Platz.
W rlbn d Mnschn n dn Strssn Krs dn Kmpf m hr Frht? En ng Lhrrn rzählt
tgbchrtg vn hrn Bschn f dm Thrr-Pltz.
Anzahl weggelassener Vokale: 51
```

Das Gerüst des Programms ist wie bei den anderen Aufgaben gegeben:

```
public class VokaleWeglassen {

	public static void main(String[] args) throws IOException {
		final BufferedReader konsolenEingabe = new BufferedReader(
				new InputStreamReader(System.in));
		final String eingabe = konsolenEingabe.readLine();

		int anzahlVokale = 0;



		System.out.println("Anzahl weggelassener Vokale: " +
				anzahlVokale);
	}

}
```

Aufgabe 7: Leerzeichen komprimieren

Schreiben Sie ein Programm, dass mehrere aufeinanderfolgende Leerzeichen durch ein einzelnes Leerzeichen ersetzt:

```
Hier    hat es viel Platz. Und  hier        noch mehr.
Hier hat es viel Platz. Und hier noch mehr.
```

Es gibt diverse Möglichkeiten, diese Aufgabe zu lösen. Ein Lösungsansatz verwendet eine Boole'sche Variable – nennen wir sie warVorherEinLeerzeichen – damit gespeichert werden kann, ob das vorgängige Zeichen in der Eingabe ein Leerzeichen war.

Überlegen Sie sich auf jeden Fall zuerst auf Papier, wie sie diese Aufgabe lösen würden.

Das Gerüst des Programms ist wie bei den anderen Aufgaben gegeben:

```
public class LeerzeichenReduzieren {

	public static void main(String[] args) throws IOException {
		final BufferedReader konsolenEingabe = new BufferedReader(
				new InputStreamReader(System.in));
		final String eingabe = konsolenEingabe.readLine();

		boolean warVorherEinLeerzeichen = false;



	}

}
```