

Programmieren in Java

1. Ermitteln Sie die Anzahl der Elemente für den Fall:
 - a) a sei vom Typ boolean[]
 - b) b sei ein Vector
 - c) c sei ein String
2. Schreiben Sie eine „for-each“ – Schleife zur Ausgabe aller Datenelemente eines Ganzzahlarrays.
3. Berechnen Sie den Umfang und die Fläche eines beliebigen Polygons dessen n Eckpunkte in zwei Arrays vom Typ double gegeben sind. Für die Fläche gilt:

$$\left| \frac{1}{2} (x_0 y_1 + x_1 y_2 + \dots + x_{n-1} y_0 - y_0 x_1 - y_1 x_2 - \dots - y_{n-1} x_0) \right|$$

Finden Sie eine Formel für den Umfang. Schreiben Sie ihre Lösung in Form eines Java-Programms. Überprüfen Sie die Lösung anhand einfacher Polygone.

4. Implementieren Sie folgenden Algorithmus für beliebige, ungerade n zur Erzeugung eines $n \times n$ magischen Quadrats. Die 1 wird in der Mitte der untersten Zeile platziert. Setzen sie die Zahlen aufsteigend in das nächste Quadrat eine Zeile unterhalb und eine Spalte nach rechts. Falls die Grenzen des Quadrats überschritten werden, so kommt die Zahl auf der anderen Seite wieder herein. Falls das Zielfeld bereits belegt ist, so ist das nächste in der gleichen Spalte eine Zeile oberhalb. Für $n = 5$ ergibt sich folgendes Bild:

11	18	25	2	9
10	12	19	21	3
4	6	13	20	22
23	5	7	14	16
17	24	1	8	15

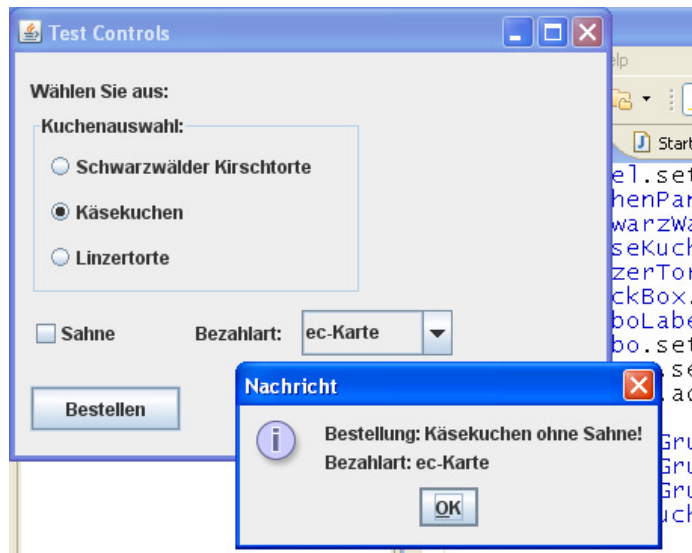
Ein magisches Quadrat liegt vor, wenn die Zeilensummen, die Spaltensummen, die Summe der Hauptdiagonalen und die Summe der Nebendiagonalen den gleichen Wert haben. Schreiben Sie ein Unterprogramm mit Namen isMagic das für eine gegebene Matrix true zurück gibt falls ein magisches Quadrat vorliegt, andernfalls ist der Rückgabewert false.

5. Der Wert eines Pixels liege als int-Variable a im folgenden Format vor.

Alpha	Red	Green	Blue
-------	-----	-------	------

Der Wert der Grünen Farbe soll durch 3 geteilt werden. Alle anderen Komponenten bleiben wie sie sind. Die geänderte Farbe soll in der Variablen a wieder gespeichert werden. Schreiben Sie Java Code.

6. Die Optionsfelder zur Auswahl des Kuchen seien als Variable JRadioButton kuchen[] deklariert und können verwendet. Zeigen Sie in einem Codefragment die Auswertung der Kuchenauswahl zur Übergabe an den Dialog auf.



7. Ein Algorithmus braucht 5s um 1000 Datenelemente zu verarbeiten. Füllen Sie folgende Tabelle aus, die die Zunahme der Laufzeit in Abhängigkeit von der Komplexität des Algorithmus aufzeigt.

	$O(n)$	$O(n^2)$	$O(n^3)$	$O(n \log(n))$	$O(2^n)$
1000	5	5	5	5	5
2000					
3000					
10000					

Finden Sie zu jeder Komplexitätsklasse ein Beispiel.

8. Alle Nachkommen Ihrer Großmutter sollen in einer Baumstruktur gespeichert werden. Bereits im Konstruktor soll ein Knoten für die Großmutter angelegt werden. Die Kinder werden durch Übergabe des Namens des Vorfahren und des eigenen Namens in den Baum eingefügt. Die toString-Methode des Baumes gibt rekursiv alle Daten aus. Zum Beispiel:

```
Baum [wurzel=
Knoten [name=Marie Hausner]
  Knoten [name=Josef Hausner]
    Knoten [name=Knuth Hausner]
    Knoten [name=Dörthe Hausner]
  Knoten [name=Rosie Heitel]
    Knoten [name=Inga Mittel]
      Knoten [name=Friederike Mittel]
      Knoten [name=Luise Mittel]
      Knoten [name=Charlotte Mittel]
    Knoten [name=Joachim Heitel]
  Knoten [name=Werner Hausner]
    Knoten [name=Matthias Hausner]
    Knoten [name=Patrick Hausner] ] ]
```

Zeichnen Sie ein UML-Klassendiagramm zur Lösung des Problems.

Ein Knoten habe die Methode „find(name:String):Knoten“. In dieser Methode soll geprüft werden ob der Knoten selbst oder eines seiner Kinder den Namen „name“ hat. Bei Erfolg wird der Knoten zurückgegeben, ansonsten null.