

Programmieren in Java

1. Ermitteln Sie die Anzahl der Elemente für den Fall:
 - a) a sei vom Typ `boolean[]`
 - b) b sei ein `ArrayList`
 - 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:

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

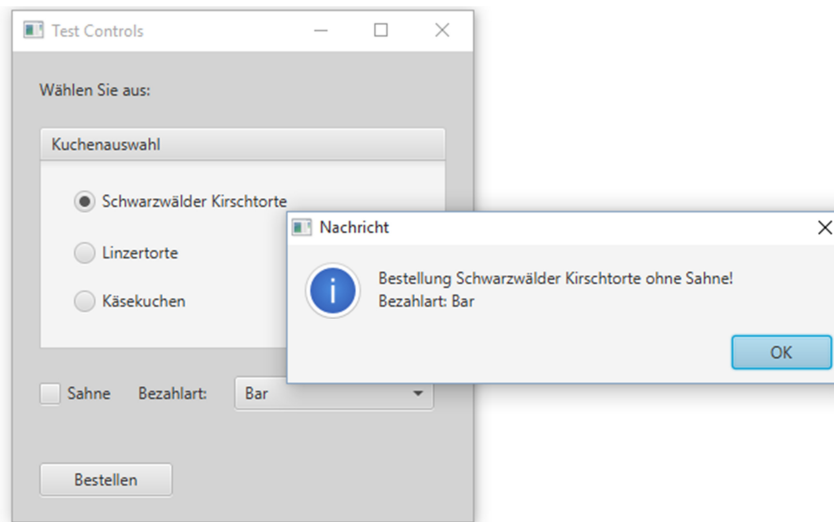
Finden Sie eine Formel für den Umfang. Schreiben Sie ihre Lösung in Form eines Java-Programms. Überprüfen Sie die Lösung anhand einfacher Polygone. Finden Sie für beide Fragestellungen auch eine rekursive Lösung.

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. Die ToggleGroup der RadioButtons, die CheckBox zur Sahneauswahl und die ComboBox für die Bezahlart liegen als Variablen vor. Zeigen Sie in einem Codefragment die Auswertung der Bestellung zur Übergabe an den Dialog auf.

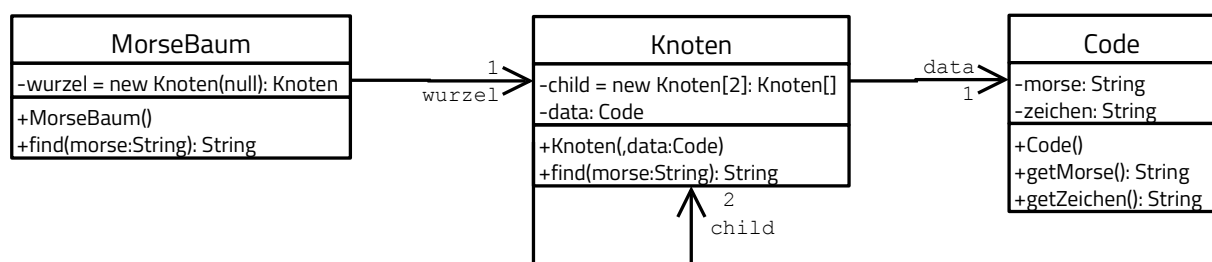


6. Ein Algorithmus braucht 5s um 1000 Datenelemente zu verarbeiten. Füllen Sie folgende Tabelle aus, die die Zunahme der Laufzeit in Abhängigkeit von der Komplexität des Algorithmus aufzeigt.

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

Finden Sie zu jeder Komplexitätsklasse ein Beispiel.

7. Der Code des Morsealphabets kann in der Struktur eines binären Baums gespeichert werden. Folgendes Klassendiagramm wurde bereits implementiert. Schreiben Sie Java-Code zur Methode „find“ der Klasse Knoten. Wie viele Suchaufrufe werden maximal gebraucht, um ein Morsezeichen zu finden?



T -	M - -	O - - -	CH - - - -
		G - - .	Ö - - - .
		K - . -	Q - - . -
	N - .	D - . .	Z - - . .
		X - . . -	Y - . - -
		B - . . .	C - . - .
E .	A . -	W . - -	J . - - -
		R . - .	P . - - .
		U . . -	Ä . - . -
	I . .	S . . .	L . - . .
			Ü . . - -
			F . . - .
			V . . . -
			H