

Java Aufgaben zu Arrays

Aufgabe 1: Befüllen von Arrays

Legen Sie jeweils ein Array eines beliebigen, ganzzahligen Datentyps der Größe 10 an. Füllen Sie das Array mit

- a) den Zahlen von 1 bis 10 (in dieser Reihenfolge) mittels einer Schleife.
- b) den Zahlen von 10 bis 1 (in dieser Reihenfolge) mittels einer Schleife.
- c) Zufallszahlen mittels einer Schleife.

Geben Sie das Array nach jeder Füllung wieder aus.

Hinweis: Für die Erzeugung der Zufallszahlen verwenden Sie Methoden der Klasse `Random` oder `Math`.

Aufgabe 2: Befüllen von Arrays und Berechnungen

Legen Sie ein Array eines beliebigen, ganzzahligen Datentyps mit beliebiger Größe an. Füllen Sie das Array mit Zufallszahlen.

Führen Sie dann folgende Berechnungen durch:

1. berechnen Sie die Summe aller Zahlen in dem Array
2. berechnen Sie den Mittelwert aller Zahlen im Array und geben diesen aus
3. finden Sie die kleinste Zahl im Array
4. finden Sie die größte Zahl im Array

Hinweis: Verwenden Sie Methoden der Klasse `Math`.

Aufgabe 3: Befüllen von Arrays und Berechnungen

Legen Sie ein Array eines beliebigen, ganzzahligen Datentyps mit der Größe 10 an. Füllen Sie das Array mit Zufallszahlen.

Lassen Sie das Array ausgeben.

Vertauschen Sie alle Zahlen im Array miteinander, d.h. die erste Zahl wird zur letzten Zahl, die zweite Zahl wird zur vorletzten Zahl, usw. Führen Sie den Tausch so durch, dass Sie kein weiteres Array anlegen müssen.

Aufgabe 4: Suchen im Array

Schreiben Sie eine Anwendung, die 100.000 Zufallszahlen zwischen 0 und 50.000 in ein Array schreibt. Der Anwender soll eine zu suchende Zahl eingeben und die Anwendung soll im Array nach der Zahl suchen. Fangen Sie fehlerhafte Eingaben der gesuchten Zahl auf die obere und untere Zahlengrenze ab.

Es wird ausgegeben: „Ja, die Zahl ist im Array“ oder „Nein, die Zahl ist nicht im Array“.

Programmieren Sie die Suche als eine sequentielle Suche, d.h. fangen Sie beim ersten Array-Element an und vergleichen es mit dem gesuchten Wert. Falls es nicht das gesuchte Element ist, gehen Sie zum nächsten Array-Element usw.

Zusätze:

Geben Sie die Stelle (d.h. den Index) aus, an dem die Zahl gefunden wurde.

Brechen Sie die Schleife ab, wenn die gesuchte Zahl im Array gefunden wurde.

Alternativ: Suchen Sie alle Vorkommen einer Zahl im Array, bzw. fragen Sie den Nutzer, ob nach weiteren Vorkommen gesucht werden soll, falls die gesuchte Zahl gefunden wurde.

Aufgabe 5: Wechselgeld ausgeben

Schreiben Sie ein Programm, welches für z.B. einen Getränkeautomaten die Art und Anzahl der Münzen ermittelt, die für die Auszahlung des Wechselgeldes benötigt wird.

Ein Beispiel:

Ein Kunde kauft ein Getränk für 35 Cent, bezahlt aber durch den Einwurf einer 1 Euro Münze. Das Restgeld entspricht 65 Cent und muss ausgezahlt werden. Idealerweise erfolgt dies in folgenden Münzen:

1 x 50 Cent

1 x 10 Cent

1 x 5 Cent

Gehen Sie davon aus, dass Sie für das Programm einen idealen Automaten mit unendlich vielen Münzen jeder Sorte zur Verfügung haben. Erstellen Sie das Programm so, dass es vom Anwender direkt den Restgeldbetrag in Cent abfragt und dann die Folge der zu zahlenden Münzen ausgibt.

Die Ausgabe könnte so aussehen:

Bitte geben Sie einen Betrag in Cent ein: 78

1 x 50 Cent

1 x 20 Cent

1 x 5 Cent

1 x 2 Cent

1 x 1 Cent

Bitte geben Sie einen Betrag in Cent ein: 127

2 x 50 Cent

1 x 20 Cent

1 x 5 Cent

1 x 2 Cent

Aufgabe 6: Lottozahlen und Häufigkeiten

Schreiben Sie ein Programm, das für das Lottospiel „6 aus 49“ einen Tipp vorschlägt, das heißt, sechs zufällige Zahlen in einem Array speichert und anschließend ausgibt. Die Zahlen müssen nicht sortiert sein.

Bei einem Lottotipp darf keine Zahl doppelt vorkommen.

Aufgabe 7: Lottozahlen und Häufigkeiten

Erstellen Sie ein int-Array mit mindestens 5000 Elementen. Füllen Sie dieses Array mit Zufallszahlen mit Zahlen zwischen 0 und 9. Legen Sie ein weiteres Array an, in dem Sie die Häufigkeit der Zahlen von 0 bis 9 des ersten Arrays zählen (d.h. wie oft kommt z.B. die 0 in dem ersten Array vor, wie oft die 1, usw.). Geben Sie anschließend den Inhalt des zweiten Arrays auf dem Bildschirm aus.

Beispielhafte Ausgabe:

Häufigkeit von 0 ist 497

Häufigkeit von 1 ist 510

Häufigkeit von 2 ist 511

Häufigkeit von 3 ist 509

Häufigkeit von 4 ist 525

Häufigkeit von 5 ist 502

Häufigkeit von 6 ist 471

Häufigkeit von 7 ist 514

Häufigkeit von 8 ist 484

Häufigkeit von 9 ist 477

Aufgabe 8: Passwortgenerator

Für eine Benutzerverwaltung sollen Passwörter nach folgendem Prinzip erstellt werden. Die Passwörter bestehen aus acht zufällig gewählten Zeichen, die Groß-, Kleinbuchstaben oder Ziffern sein können. Beispiel: w2hoH7a!

Das Passwort muss dabei wie folgt zusammengesetzt sein:

5 Buchstaben (mindestens ein Großbuchstabe) 2 Ziffern 1 Sonderzeichen

Die Reihenfolge spielt keine Rolle. Speichern Sie das Passwort in einem Array und geben Sie es anschließend aus. Die nötigen ASCII-Zeichen für die Aufgabe finden Sie im Anhang

Tabelle für druckbare ASCII-Zeichen

Die Zahlen 32–126 sind Zeichen zugeordnet, für die Tasten auf der Tastatur vorhanden sind. 127 stellt den Befehl zum Löschen dar.

Dezimalzahl	Zeichen	Dezimalzahl	Zeichen
32	Leerzeichen	80	P
33	!	81	Q
34	"	82	R
35	#	83	S
36	\$	84	T
37	%	85	U
38	&	86	V
39	'	87	W
40	(88	X
41)	89	Y
42	*	90	Z
43	+	91	[
44	,	92	\
45	-	93]
46	.	94	^
47	/	95	_
48	0	96	`
49	1	97	a
50	2	98	b
51	3	99	c
52	4	100	d
53	5	101	e
54	6	102	f
55	7	103	g
56	8	104	h
57	9	105	i
58	:	106	j
59	;	107	k
60	<	108	l
61	=	109	m
62	>	110	n
63	?	111	o
64	@	112	p
65	A	113	q
66	B	114	r
67	C	115	s
68	D	116	t
69	E	117	u
70	F	118	v
71	G	119	w
72	H	120	x
73	I	121	y
74	J	122	z
75	K	123	{
76	L	124	
77	M	125	}
78	N	126	~
79	O	127	Löschen