

```
1 // Lottoschein.cpp : Aufgabe zu Schleifen und Datenfeldern.
2 //
3 // Die üblichen Vorab-Anweisungen:
4 #include "stdafx.h"
5 #include <iostream>
6 using namespace std;
7
8 // Für Zufallszahlen ganz geschickt:
9 #include <ctime> // für die Time-Funktion als Startwert für den Zufallszahlengenerator
10
11
12 int main() {
13     // erst mal ganz einfach:
14     // 10000 Zufallszahlen von 1 bis 49 erzeugen:
15     const int MAX = 49; // es gibt die Lottozahlen 1 bis 49 - OBACHT beim INDEX [0] bis [48]
16     int Lottozahl;
17     int LottoArray[MAX]; // Ein Array für 49 Lottozahlen
18
19     // Zusätzlich bei jedem Programmaufruf neue Startwerte kreieren:
20     // für den time-Befehl brauchen wir die o.a. <ctime>-Bibliothek
21     srand(unsigned(time));
22
23     // Anzahl auf 0 zurücksetzen:
24     for (Lottozahl=0; Lottozahl<49; Lottozahl++)
25         LottoArray[Lottozahl] = 0;
26
27     // Jetzt 10000 Ziehungen:
28     for (int Ziehung=0; Ziehung<10000; Ziehung++) {
29         // rand() liefert eine Zufallszahl von 0 bis 4Mrd.
30         // Modulo 49 liefert Rest 0 bis Rest 48 - also gut brauchbar...
31         Lottozahl = rand() % MAX;
32         // Und die Zahl als gezogen markieren:
33         LottoArray[Lottozahl]++;
34     }
35
36     // Jetzt Ergebnis anzeigen:
37     for (Lottozahl=0; Lottozahl<49; Lottozahl++) {
38         cout << "Die Zahl ";
39         if (Lottozahl<9) cout << "0";
40         cout << Lottozahl+1 << " wurde " << LottoArray[Lottozahl] << " mal gezogen
41         \n";
42     }
43     cout << endl;
44
45     system("pause");
46     return 0;
47 }
48
```

```
1 // Lottoschein2.cpp : Aufgabe zu Schleifen und Datenfeldern.
2 //
3 // Die üblichen Vorab-Anweisungen:
4 #include "stdafx.h"
5 #include <iostream>
6 using namespace std;
7
8 // Für Zufallszahlen ganz geschickt:
9 #include <ctime> // für die Time-Funktion als Startwert für den Zufallszahlengenerator
10
11
12 int main() {
13     // OK, jetzt etwas anspruchsvoller:
14     // Lottoschein füllen, dabei aufpassen, dass eine Zahl nicht mehrmals gezogen werden darf!
15
16     const int MAX = 49; // es gibt die Lottozahlen 1 bis 49 - OBACHT beim INDEX [0] bis [48]
17     int Lottozahl;
18     int LottoArray[MAX]; // Ein Array für 49 Lottozahlen
19
20     // Zusätzlich bei jedem Programmaufruf neue Startwerte kreieren:
21     // für den time-Befehl brauchen wir die o.a. <ctime>-Bibliothek
22     srand(unsigned(time));
23
24     // Anzahl auf 0 zurücksetzen:
25     for (Lottozahl=0; Lottozahl<49; Lottozahl++)
26         LottoArray[Lottozahl] = 0;
27
28     // Jetzt ?? wieviel ??? Ziehungen:
29     int AnzahlGezogen = 0;
30     do {
31         // wieder die zufällige Lottozahl von 0 bis 48 erzeugen:
32         Lottozahl = rand() % MAX;
33         // ist aber nur dann brauchbar, wenn die Zahl nicht schon einmal gezogen wurde:
34         if (LottoArray[Lottozahl] == 0) {
35             // OK, ist noch auf 0, wurde also noch nicht gezogen
36             LottoArray[Lottozahl] = 1;
37             AnzahlGezogen++;
38         }
39         // else brauchen wir nicht, weil wir einfach so tun, als sei nichts geschehen...
40     } while (AnzahlGezogen<6);
41
42     // Jetzt Ergebnis anzeigen:
43     // einfach übernommen aus Aufgabe 1:
44     for (Lottozahl=0; Lottozahl<49; Lottozahl++) {
45         cout << "Die Zahl ";
46         if (Lottozahl<9) cout << "0";
47         cout << Lottozahl+1 << " wurde " << LottoArray[Lottozahl] << " mal gezogen
48         \n";
49     }
50     cout << endl;
```

```
51
52 // Das Ergebnis können wir auch etwas schöner darstellen:
53 // mit 7 Zeilen zu je 7 Spalten
54 for (int Zeile=0; Zeile<7; Zeile++) {
55     for (int Spalte=0; Spalte<7; Spalte++) {
56         // das mathematisch anspruchsvollste ist die Berechnung des 1 Index aus 2 ↗
57         // Zahlen ;- )
58         Lottozahl = 7*Zeile + Spalte; // puh, war DAS schwer...
59         // wenn gezogen, malen wir ein _X
60         if (LottoArray[Lottozahl] > 0)
61             cout << " X";
62         else {
63             if (Lottozahl<9) cout << " ";
64             cout << Lottozahl+1;
65         }
66         cout << " "; // für etwas Abstand
67     }
68     cout << endl; // Umbruch in nächste Zeile
69 }
70 cout << endl; // Kosmetik: Leerzeile zum Programmende
71
72 system("pause");
73 return 0;
74 }
```

```
1 // Lottoschein3.cpp : Aufgabe zu Schleifen und Datenfeldern.
2 //
3 // Die üblichen Vorab-Anweisungen:
4 #include "stdafx.h"
5 #include <iostream>
6 using namespace std;
7
8 // Für Zufallszahlen ganz geschickt:
9 #include <ctime> // für die Time-Funktion als Startwert für den Zufallszahlengenerator
10
11
12 int main() {
13
14     // OK, jetzt noch anspruchsvoller:
15     // Lottoschein selbst ausfüllen,
16     // dann Lottozahlen ziehen (dabei aufpassen, dass eine Zahl nicht mehrmals
    gezogen werden darf!)
17     // und Ergebnis "ansprechend" darstellen.
18
19     const int MAX = 49; // es gibt die Lottozahlen 1 bis 49 - OBACHT beim INDEX [0]
    bis [48]
20     int Lottozahl;
21     int LottoArray[MAX]; // Ein Array für 49 Lottozahlen
22     int MeineLottozahlen[MAX]; // ein zusätzliches Array für meinen eigenen Tipp
23
24     // Zusätzlich bei jedem Programmaufruf neue Startwerte kreieren:
25     // für den time-Befehl brauchen wir die o.a. <ctime>-Bibliothek
26     srand(unsigned(time));
27
28     // Anzahl auf 0 zurücksetzen - in beiden Arrays:
29     for (Lottozahl=0; Lottozahl<49; Lottozahl++) {
30         LottoArray[Lottozahl] = 0;
31         MeineLottozahlen[Lottozahl] = 0;
32     }
33
34     // 6 Zahlen vom Benutzer erfragen:
35     // wir kopieren einfach die gleiche Prozedur wie beim Ziehen (s.u.) und
    modifizieren sie:
36     // Jetzt ?? wieviel ??? Eingaben:
37     int AnzahlGetippt = 0;
38     do {
39         cout << "Bitte geben Sie Ihren " << AnzahlGetippt+1 << ". Tipp ein: ";
40         cin >> Lottozahl;
41         // ist aber nur dann brauchbar, wenn die Zahl nicht schon einmal eingegeben
    wurde:
42         // UND die Zahl im Intervall [1,49] liegt!
43         if (Lottozahl < 1 || Lottozahl > 49) {
44             cout << "Die Zahl darf nicht kleiner als 1 und nicht groesser als 49
    sein!\n";
45         }
46         else {
47             if (MeineLottozahlen[Lottozahl-1] == 0) {
48                 // OK, ist noch auf 0, wurde also noch nicht gezogen
49                 // UND Achtung wegen der Index-Korrektur!!!!
50                 MeineLottozahlen[Lottozahl-1] = 1;
```

```
51         AnzahlGetippt++;
52     }
53     else {
54         // else brauchen wir hier schon wegen einer vernünftigen Benutzerführung:
55         cout << "Sie haben die Zahl " << Lottozahl << " bereits getippt!\n";
56     }
57 }
58 } while (AnzahlGetippt<6);
59
60 cout << "\nUND NUN DIE ZIEHUNG:\n";
61 // Jetzt ?? wieviel ??? Ziehungen:
62 int AnzahlGezogen = 0;
63 do {
64     // wieder die zufällige Lottozahl von 0 bis 48 erzeugen:
65     Lottozahl = rand() % MAX;
66     // ist aber nur dann brauchbar, wenn die Zahl nicht schon einmal gezogen wurde:
67     if (LottoArray[Lottozahl] == 0) {
68         // OK, ist noch auf 0, wurde also noch nicht gezogen
69         LottoArray[Lottozahl] = 1;
70         AnzahlGezogen++;
71     }
72     // else brauchen wir nicht, weil wir einfach so tun, als sei nichts geschehen...
73 } while (AnzahlGezogen<6);
74
75 // Jetzt Ergebnis anzeigen und auch prüfen:
76 int AnzahlRichtig = 0; // Anzahl Treffer
77
78 // mit 7 Zeilen zu je 7 Spalten
79 for (int Zeile=0; Zeile<7; Zeile++) {
80     for (int Spalte=0; Spalte<7; Spalte++) {
81         // das mathematisch anspruchsvollste ist die Berechnung des 1 Index aus 2 Zahlen ;- )
82         Lottozahl = 7*Zeile + Spalte; // puh, war DAS schwer...
83         // wenn gezogen, malen wir ein _X
84         if (LottoArray[Lottozahl] > 0) {
85             cout << " X";
86             // Hier testen wir auch gleich auf einen richtigen Tipp:
87             if (MeineLottozahlen[Lottozahl] > 0) {
88                 // HURRA - richtig getippt!!
89                 AnzahlGezogen++;
90             }
91         }
92         else {
93             if (Lottozahl<9) cout << " ";
94             cout << Lottozahl+1;
95         }
96         cout << " "; // für etwas Abstand
97     }
98     cout << endl; // Umbruch in nächste Zeile
99 }
100 cout << endl; // Kosmetik: Leerzeile
101
102 cout << "SIE HABEN " << AnzahlRichtig << " RICHTIGE GETIPPT\n\n";
```

```
103
104     system("pause");
105     return 0;
106 }
107
```