Projekt 7: Wenn ich Lottomillionär werde –Speichern in einem Array

Ziehung am Samstag:
Angaben ohne Gewähr
4, 10, 18, 25, 29, 31

6 Richtige:	1,2 Millionen €
5 Richtige:	5223 €
4 Richtige:	55,90 €
3 Richtige:	11,90 €

In diesem Projekt kann sich der Benutzer seine persönlichen Lottozahlen aussuchen. Dann werden 100.000 Ziehungen der Lottozahlen simuliert, das schaft Java. Zum Schluss wird untersucht, wie oft dabei 4, 5 oder 6 richtige gezogen wurden und wie viel Geld man gewonnen hätte.

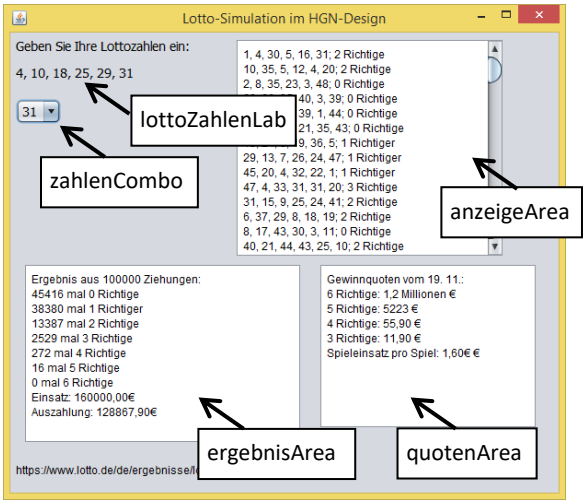
Bei einer Ziehung der Lottozahlen müssen 6 Zahlen gespeichert werden, dazu eignet sich ein sogenanntes **Array**: Ein Array stellt mehrere Speicherplätze des gleichen Datentypes zur Verfügung, die man mit einer Nummer, dem sogenannten **Index**, ansperechen kann. Die Nummer steht immer in eckigen Klammern. Um die obigen Lottozahlen abzuspeichern geht man so vor:
In der obersten Zeile wird ein Array mit 7 Speicherplätzen vom Typ Integer deklariert. Die Zahl 7 ist notwendig, da Java immer bei 0 anfängt zu zählen. Es wurden also Speicherplätze mit den Indizes 0, 1, 2, 3, .. , 6 erzeugt, den Speicherplatz mit der Nummer 0 benötigen wir nicht. Anschließend werden die Speicherplätze mit den Nummern 1 bis 6 mit den gewünschten Lottozahlen belegt. Ein Array kann man sich auch vorstellen wie eine Kommode mit vielen Schubladen, in jeder Schublade steckt eine Variable, unsere Kommode hat 7 Schubladen, die erste bleibt leer.

```
int [] meineLottozahlen = new int [7];
```

Datentyp Integer | Zeichen für ein Array | Name des Arrays | Speicherplätze von 0 bis 6

```
meineLottozahlen [1] = 4;
meineLottozahlen [2] = 10;
meineLottozahlen [3] = 18;
meineLottozahlen [4] = 25;
meineLottozahlen [5] = 29;
meineLottozahlen [6] = 31;
```

Aufgabe 1: Importiere das Programm Lottomillionen aus dem Schülerverzeichnis. Die grafischen Elemente der Klasse Gui sind bereits erstellt. Der Benutzer soll in der Combobox **zahlenCombo** jeweils eine Lottozahl aussuchen können. Schreibe eine Event-Prozedur so, dass die gerade gezogene Zahl im Array **meineLottozahlen** gespeichert wird. Im Label **meineLottozahlenLab** sollen ferner immer alle bisher gezogenen Lottozahlen angezeigt werden. Du benötigst dazu eine Variable **anzahlZahlen**, in der gespeichert wird, wie viele Zahlen bisher vom Benutzer ausgewählt wurden.



Aufgabe 2: Nachdem die letzte Lottozahl gewählt wurde, sollen mit dem Zufallsgenerator – zunächst einmal - 6 Zahlen gezogen und damit eine Ziehung simuliert werden. Die 6 Zahlen werden in einem neuen Array **gezogeneZahlen** gespeichert. Anschließend wird gezählt, wie viele „Richtige“ Lottozahlen unter den selbstgewählten sich befinden. Das Ergebnis soll in der TextArea **anzeigeArea** im rechten Gui-Bereich angezeigt werden (s. Grafik). Rechts siehst du zwei Beispiele zum Ziehen von Zufallszahlen, du musst die Zeile etwas abwandeln, damit sie zur Ziehung der Lottozahlen passt.

```
x1 = (int) ((Math.random() * 5);
  zieht eine Zufallszahl zwischen 0 und 4
  (einschließlich 0 und 4)
x1 = (int) ((Math.random() * 5) + 200);
  zieht eine Zufallszahl zwischen 200 und 204
```

Tipp: Mit dem Befehl anzeigeArea.append („Text“); fügt man in der TextArea weiteren Text ein. Einen Zeilensprung erhält man mit “\n“.

Wenn die Anzeige funktioniert, lasse innerhalb einer Schleife den Ziehungsvorgang 100.000 mal ablaufen.
Noch ein Tipp: Beim Ziehen der Zahlen kann es vorkommen, dass man zwei mal die gleiche Zahl zieht. Dies kann man foolgendermaßen verhindern. Nachdem man die k-te Zahl gezogen hat, vergleicht man diese Zhal mit den bisherigen Zahlen:

```
boolean gefunden = false;
while (gefunden == false){
    neueZahl = …  // hier wird die neue Zahl gezogen
    gleicheZahl = false;
    m = 1;
    while ((m < k) && (gleicheZahl = false)){
       if (neueZahl == gezogeneZahl [m]){
          gleicheZahl = true;
       }
    }
    if (gleicheZahl == false){
        gefunden = true;
        gezogeneZahl [k] = neueZahl;  // jetzt wird
                          // die neue Zahl gespeichert
    }
}
```

```
boolean gefunden = false;
while (gefunden == false){
  neueZahl = (int)((Math.random()) * 49 + 1);
  gleicheZahl = false;
  m = 1;
  while ((m < k) && (gleicheZahl = false)){
   if (neueZahl == gezogeneZahl [m]){
     gleicheZahl = true;
   }
  }
  if (gleicheZahl == false){
     gefunden = true;
     gezogeneZahl [k] = neueZahl;
  }
}
```

Kann kopiert werden!

Aufgabe 3: Nachdem 100.000 Ziehungsvorgänge simuliert wurden, soll ausgezählt werden, wie oft 0 Richtige, 1 Richtiger, 2 Richtige, … bis 6 Richtige vorkamen. Es reicht, wenn man nur die gerade gezogenen Lottozahlen speichert, wenn man ein neues Array **richtige** anlegt, indem jeweils am Ende einer Ziehung weitergezählt wird, wie oft z. B. 1 Richtiger, 2 Richtige usw. bisher gefallen sind. Zu Beginn müssen die Zähler auf Null gesetzt werden. Das Ergebnis soll schließlich in der Textarea **ergebnisArea** ausgegeben werden.

Zusatzaufgabe 1: Am Ende der 100.000 Ziehungen soll in der **ergebnisArea** in zwei weiteren Zeilen ausgegeben werden, wie viel Geld man für die Tippscheine hätte zahlen müssen und wie groß der ausgezahlte Gewinn gewesen wäre. Dazu kannst du die Gewinnquoten, die in der **quotenArea** dargestellt sind, verwenden (tatsächliche Gewinnquoten vom 19. 11. 2016), oder aus dem Inernet die aktuellen Gewinnquoten vom letzten Samstag entnehmen. Eine mögliche Adresse dazu ist am unteren Bildschirmberich eingeblendet. In der quotenArea sollten ferner Die Gewinnquoten dargestellt werden (s. gelber Kasten).

```
quotenArea.append ("Gewinnquoten vom 19. 11.:\n");
quotenArea.append ("6 Richtige: 1,2 Millionen €\n");
quotenArea.append ("5 Richtige: 5223 €\n");
quotenArea.append ("4 Richtige: 55,90 €\n");
quotenArea.append ("3 Richtige: 11,90 €\n");
quotenArea.append ("Spieleinsatz pro Spiel: 1,60€ €\n");
```

Kann kopiert werden!

Zusatzaufgabe 2: Zur Zeit könnte ein Benutzer noch mehrmals eine gleiche Lottozahl eingeben. Ergänze dein Programm so, dass eine solche Fehleingabe durch eine Fehlermeldung abgefangen und der Benutzer die letzte Lottozahl erneut eingeben kann.