

# Beispiel-Projekt „Lotto“

## *Ziehung von Lotto-Zahlen*

Erstelldatum:	2018-07-11
Diese Ausgabe ersetzt:	-/-
Verbindliche Änderungshistorie:	siehe I
Geltungsbereich:	SCHERDELGruppe
Herausgeber:	SCHERDELGruppe
Status:	in Erstellung/Freigegeben
Autor:	Harald Kirmaß

## INHALT

1	Aktuelle Situation des Projekts .....	3
1.1	Ausgangslage .....	3
1.2	Clonen des Repositories von Git-Hub .....	3
1.3	Projekt-Überblick verschaffen.....	3
1.4	Funktionalität des aktuellen Standes prüfen .....	3
2	Fehlerkorrekturen .....	4
2.1	Fehlender Willkommenstext .....	4
2.2	Eingabeprüfung – Prüfung auf Nummern.....	4
2.3	Eingabeprüfung – doppelte Nummerneingabe.....	4
2.4	Eingabeprüfung – Nummern-Bereich abprüfen.....	4
2.5	Überprüfung des Algorithmus für die Zahlenziehung .....	4
2.6	Überprüfung von Algorithmus für die Zahlenziehung .....	4
2.7	Korrektur Ausgabe.....	4
3	Anfragen für Optimierungen und Erweiterungen.....	5
3.1	Erweiterung Bestätigung – Eingabe „y“ .....	5
3.2	Erweiterung Bestätigung – Möglichkeit zum Verwerfen der Zahlen.....	5
3.3	Ablauf Menü – kein Programm-Ende nach Ziehung .....	5
3.4	Erweiterung Ziehung der Gewinnzahlen .....	5
3.5	Erweiterung Auswertung der Gewinnzahlen mit Jackpot-Information.....	6
3.6	Erweiterung Ziehung – Statistische Ziehung .....	6
3.7	Erweiterung Ziehung der Gewinnzahlen für mehrere Benutzer .....	6

## 1 Aktuelle Situation des Projekts

### 1.1 Ausgangslage

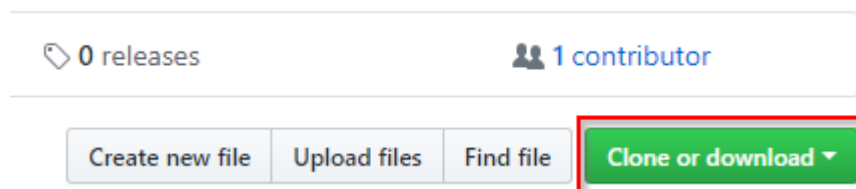
Bei einem internen Projekt wurde eine Konsolenapplikation implementiert, die die Lotto-Ziehung abbildet. Die Applikation ist bereits im Einsatz.

Es werden 6 Zahlen zwischen 1 und 49 von einem Benutzer abgefragt. Danach werden 6 Zufallszahlen aus demselben Zahlenbereich gezogen. Über einen Abgleich dieser Zahlen wird ermittelt, wie viele Zahlen der Benutzer richtig hat. 6 gleiche Zahlen bedeutet, dass der Jackpot geknackt wurde.

### 1.2 Clonen des Repositories von Git-Hub

Gehen Sie auf Github.com und clonen Sie sich das Repository lokal unter dem Verzeichnis „C:\Dev“.

<https://github.com/Scherdel/TrainingLottery>



Oder direkt per Link:

<https://github.com/Scherdel/TrainingLottery.git>

### 1.3 Projekt-Überblick verschaffen

Der Quellcode des Projektes liegt jetzt bei Ihnen unter C:\Dev\TrainingLottery\LotteryConsole.

Öffnen Sie die Applikation mit Ihrem Visual Studio Code.

Dazu öffnen Sie eine Kommando-Zeile (**cmd**) und navigieren in das Verzeichnis (cd C:\Dev\TrainingLottery\LotteryConsole).

Mit dem Befehl „**code** .“ öffnen Sie das Projekt in Visual Studio Code.

Der Einstieg in die Konsolen-Applikation befindet sich in der Klasse „Program.cs“.

Die Funktionalität des Lotto-Spiels befindet sich in der Klasse „Lottery.cs“ und die User-Interaktion auf der Konsole in der Klasse „LotteryMenue.cs“.

Verschaffen Sie sich einen Überblick über den Source-Code und versuchen Sie die Funktionalität nachzuvollziehen.

### 1.4 Funktionalität des aktuellen Standes prüfen

Führen Sie die Applikation mit „F5“ aus und spielen Sie einen Geradeausfall durch. Funktioniert alles wie oben beschrieben?

Seien Sie kreativ und versuchen Sie die Applikation mit fehlerhaften Eingaben zum Absturz zu bringen.

## 2 Fehlerkorrekturen

Leider wurden in der aktuellen Version einige Fehler gemacht. Versuchen Sie die Applikation zu verbessern.

### 2.1 Fehlender Willkommenstext

Der Entwickler der Applikation hatte vergessen einen Start-Text als „Willkommenstext“ anzuzeigen. Versuchen Sie beim Start der Applikation einen Willkommenstext zu integrieren.

### 2.2 Eingabeprüfung – Prüfung auf Nummern

Aktuell wird bei der Eingabe von Zahlen auf den Benutzer vertraut, dass Zahlen eingegeben werden. Bei Buchstaben oder andere Zeichen kommt ein Laufzeitfehler.

Verbessern Sie die Eingabe, dass das Programm bei einer fehlerhaften Eingabe nicht abstürzt, sondern einen Fehler ausgibt und die Eingabe wiederholt.

Sehen Sie sich dazu folgenden Source-Code an:

```
string input = "A";
int n = -1;
if(int.TryParse(input, out n)) {
    //n is a number
}
```

### 2.3 Eingabeprüfung – doppelte Nummerneingabe

Es ist in der aktuellen Version möglich eine Zahl doppelt einzugeben. Das macht natürlich keinen Sinn. Versuchen Sie hier abzu prüfen, ob eine eingegebene Zahl bereits erfasst wurde. Falls ja ignorieren Sie die Zahl, geben eine Fehlermeldung aus und fahren Sie mit einer neuen Eingabe fort.

### 2.4 Eingabeprüfung – Nummern-Bereich abprüfen

Da nur Zahlen zwischen 1 und 49 gewinnen können, sollten auch nur Zahlen in diesem Bereich für die Eingabe vorgesehen werden.

Prüfen Sie bei der Eingabe der Zahlen ab, ob die Zahl im o.g. Nummernkreis ist.

### 2.5 Überprüfung des Algorithmus für die Zahlenziehung

In der Klasse „Lottery.cs“ gibt es eine Methode „GetWinningNumbers()“. Überprüfen Sie die Logik der Methode. Über die Klasse Random werden Zufallszahlen ermittelt. Die Methode dazu ist „Next(von, bis)“. Gehen Sie mit dem Mauszeiger über die Funktion und sehen sich die IntelliSense an, ob die Übergabeparameter richtig konfiguriert sind.

### 2.6 Überprüfung von Algorithmus für die Zahlenziehung

Wie in 2.3 beschrieben, können aktuell auch Gewinnzahlen doppelt vorkommen. Überprüfen Sie analog zu 2.3 auch die Gewinnzahlen auf doppelte Ergebnisse.

### 2.7 Korrektur Ausgabe

Bei der Ausgabe der richtigen Nummern wird bei einer richtigen Zahl folgender Text angezeigt: „Sie haben 1 Zahlen richtig!“. Ändern Sie die Ausgabe entsprechend ab, dass bei einer Zahl auch nur der Singular angezeigt wird.

### 3 Anfragen für Optimierungen und Erweiterungen

#### 3.1 Erweiterung Bestätigung – Eingabe „y“

Zusätzlich zu den aktuellen Möglichkeiten „j“, „J“ für „Ja“ soll auch die englische Variante „y“ oder „Y“ berücksichtigt werden.

#### 3.2 Erweiterung Bestätigung – Möglichkeit zum Verwerfen der Zahlen

Aktuell ist es möglich bei der Bestätigung mit „ja“ und „nein“ zu antworten.

Erweitern Sie diese Funktionalität um die Variante „neue Zahlen erfassen“. Überlegen Sie sich dazu ein Kürzel und erweitern Sie das Menü. Bei der Auswahl dieser Variante sollen die bisher eingegebenen Zahlen verworfen werden und das Menü zur Eingabe der Zahlen zurückkehren.

#### 3.3 Ablauf Menü – kein Programm-Ende nach Ziehung

Erweitern Sie das Programm, dass es nach der Ziehung nicht endet. Implementieren Sie dafür ein Hauptmenü, in dem man das Programm schließen und eine neue Ziehung starten kann.

Optional können Sie das Menü derartig erweitern, dass die Eingabe der Zahlen und die Ziehung unterschiedliche Menüpunkte darstellen. Somit wären mehrere Ziehungen für ausgewählte Zahlen möglich.

#### 3.4 Erweiterung Ziehung der Gewinnzahlen

Aktuell werden die Zahlen sehr schnell gezogen und dem Benutzer sofort angezeigt. Das dient nicht der Spannung der Ziehung.

Erweitern Sie die Ziehung der Zahlen, dass diese nach und nach gezogen werden und Zahl für Zahl dem Benutzer präsentiert werden. Lassen Sie zwischen den Zahlen etwas Zeit vergehen. Sehen Sie sich dazu folgenden Quellcode an:

```
using System;
using System.Threading;

namespace lotto_console {
    1 reference
    class Lotto {
        0 references
        public void Wait() {
            Thread.Sleep(1000);
        }
    }
}
```

Optional können Sie die Zahlen – falls falsch oder richtig – auch mit einer Farbe hinterlegen (grün = richtig; rot = falsch).

Sehen Sie sich dazu folgenden Quellcode an:

```
static void Main(string[] args) {
    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine("Falsch!");
    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("Richtig!");
    Console.ForegroundColor = ConsoleColor.White;
```

Versions-Nr.:1.0

Die Ausgabe sieht folgendermaßen aus:

```
Das Laden von persönlichen und Systemprofilen dauerte 960 ms.  
C:\Temp\lotto-console> & 'C:\Users\u32791\.vscode\extensions\ms-v  
Falsch!  
Richtig!  
C:\Temp\lotto-console> |
```

### *3.5 Erweiterung Auswertung der Gewinnzahlen mit Jackpot-Information*

Erweitern Sie die Auswertung der Gewinnzahlen um eine Information, wenn der Jackpot geknackt wird (alle 6 Zahlen sind richtig!)

### *3.6 Erweiterung Ziehung – Statistische Ziehung*

Implementieren Sie im Hauptmenü eine Funktion, die solange eine Ziehung durchführt, bis die eingegebenen Zahlen gezogen wurden. Geben Sie die Anzahl an notwendigen Ziehungen auf der Konsole aus.

### *3.7 Erweiterung Ziehung der Gewinnzahlen für mehrere Benutzer*

Gerne würden zukünftig mehr als eine Person an einer Ziehung teilnehmen.

Legen Sie dazu eine Klasse „Contributors.cs“ an mit dem Namen des Teilnehmers und dessen Gewinnzahlen. Erfassen Sie über das Hauptmenü beliebig viele Teilnehmer und starten Sie die Ziehung über einen anderen Menüpunkt. Sichern Sie die Eingaben in einem Array und werten diese bei der Ziehung aus.