

Praktikumsaufgabe Hangman (mit Award - S. 2)

Entwickeln Sie Ihre Variante des bekannten Spiels Hangman (Galgenmännchen). Der Benutzer soll zufällig eines von mindestens 10 im Programm vordefinierten Wörtern raten. Dafür soll das Suchwort zunächst durch Minuszeichen "-" dargestellt werden. Ist zum Beispiel das Wort "Programmieren" ausgewählt, bekommt der Benutzer die Anzeige



Das geheime Wort ist: -----

Drückt er nun einen Buchstaben, der im gesuchten Wort vorkommt, so wird er an den entsprechenden Stellen eingeblendet und eine Erfolgsmeldung ausgegeben. Wird ein Buchstabe gedrückt, der nicht vorkommt, so wird ein Fehlerzähler hochgezählt und angezeigt.

Wurden alle Buchstaben und damit das Wort vor Erreichen der maximalen Fehlerzahl gefunden, so hat der Spieler gewonnen, anderenfalls hat er verloren. Die maximale Fehlerzahl dürfen Sie selbst im Programm festlegen. Sinnvoll sind Werte zwischen 5 und 10.

Hier eine beispielhafte Ausgabe für das geheime Suchwort "Programmieren".

```
Hangman
=====
Gesucht ist ein Wort mit 13 Buchstaben.
Das geheime Wort ist: -----

Buchstaben eingeben: a
Das geheime Wort ist: -----a----- --> a ist richtig!

Buchstaben eingeben: e
Das geheime Wort ist: -----a---e-e- --> e ist richtig!

Buchstaben eingeben: x
Das geheime Wort ist: -----a---e-e- --> x ist falsch!
1 von 5 Fehlern.

Buchstaben eingeben: t
Das geheime Wort ist: -----a---e-e- --> t ist falsch!
2 von 5 Fehlern.
...

Buchstaben eingeben: p
Das geheime Wort ist: programmieren --> p ist richtig!
Du hast gewonnen!
```

Die Ausgabe muss, abgesehen von der Darstellung des Suchwortes, nicht exakt so aussehen. Sie müssen aber bei jedem Fehler ein Feedback über die bisher gemachte Fehlerzahl geben.

Das Programm soll nicht nach Groß- und Kleinbuchstaben unterscheiden.

Aus der C#-Funktionsbibliothek dürfen Sie neben den Console-Funktionen nur die Funktionen zur Zufallszahlen-Generierung, ToCharArray() sowie ToLower()/ToUpper() verwenden.

Lösungshinweise

- Legen Sie die möglichen Suchworte in einem String-Array ab und wählen beim Programmstart zufällig eins aus.
- Sie können für das Lösungswort ein Character-Array der gleichen Länge wie das Suchwort anlegen, das Sie zunächst mit lauter Minus-Zeichen initialisieren.
- Ein einzelnes Zeichen können Sie mit `Console.ReadKey().KeyChar` einlesen.
- Die Methode `Char.ToLower(c)` wandelt ein Zeichen `c` in einen Kleinbuchstaben um. Die String-Methode `s.ToLower()` tut dasselbe für einen String `s`.

The Best Hangman Award

Die Programmierung eines Hangman ist optional und geht nicht in die offizielle Praktikumswertung ein, hier geht es um Ruhm und Ehre (und Spaß)!

Wer Lust hat, kann den Coolness-Faktor des Spiels durch eine entsprechende Ausgabe nach jedem Fehler deutlich erhöhen. Ein einfaches Beispiel, das es mindestens zu erreichen gilt, könnte so aussehen:



```
5 von 5 Fehlern.
```

```
+---+-  
|  o  
|  /|\  
|  / \  
|  \/  
|  \
```

```
Du bist tot!
```

Es darf dabei nur mit der Konsolen-Ausgabe gearbeitet werden (mehr dazu unten), d.h. die Verwendung echter Grafik-Funktionen ist nicht erlaubt!

Natürlich sollte nicht nur am Schluss ein witziges Piktogramm ausgegeben werden. Vielmehr soll sich das Bild mit jedem Fehler schrittweise aufbauen.

Auszeichnung der Besten Lösungen! 100€ Gewinnsumme zu verteilen!

Jede/r kann seine eigene Lösung oder eine Lösung, die er korrigiert hat, in Moodle für den "Best Hangman Award" nominieren. Dafür ist eine entsprechende Moodle-Aufgabe angelegt. Geben Sie hier einfach Ihren eigenen Namen an (wenn Sie sich selbst nominieren möchten) oder laden Sie bei der Bewertung eine Solution hoch, die nominierungswürdig erscheint. Nominierungen sind bis zum Ende des Bewertungszeitraum möglich.

In einer Vorauswahl werden die besten Lösungen ausgewählt (solange die Nominierten zustimmen) und dann allen Kursteilnehmer zur Endauswahl in Moodle vorgestellt. Bei der Endauswahl werden das ausführbare Programm und der Quelltext zur Begutachtung veröffentlicht. D.h. Ihr Programm sollte von innen und außen gut aussehen!

Die drei Bestplatzierten erhalten eine wundervolle Urkunde 😊. Außerdem wird eine Gewinnsumme von 100€ ausgeschrieben, die aber nicht an die Gewinner ausgeschüttet, sondern die für einen guten Zweck gespendet wird. Die Gewinner dürfen festlegen, welche Organisation Sie unterstützen wollen (Platz 1: 50€, Platz 2: 30€, Platz 3: 20€). Dabei kommen alle Organisationen in Frage, die das DZI-Spendensiegel haben



(<https://www.dzi.de/spenderberatung/das-spenden-siegel/liste-aller-spenden-siegel-organisationen-a-z/>).

Tipp für die Optik: Mit Unicode-Grafiken die Ausgabe verschönern

Wir machen zwar keine Grafik-Programmierung, aber man kann auch ansprechende Konsolen-Programme schreiben. So lassen sich mit Unicode-Zeichen Rahmen, Kästen, Symbole etc. darstellen. Eine Übersicht über die entsprechenden Codes findet sich hier:

<https://www.key-shortcut.com/zeichentabellen/unicode-2000-2fff/#c1608>

Ein kleines Beispiel illustriert die Verwendung. Ausgegeben wird der eingerahmte Smiley.
Die erste Zeile ist evtl. notwendig, damit die Konsolendarstellung richtig funktioniert.

```
Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.BackgroundColor = ConsoleColor.White;
Console.ForegroundColor = ConsoleColor.Blue;
Console.WriteLine("\u2554\u2550\u2550\u2550\u2557");
Console.Write("\u2551");
Console.ForegroundColor = ConsoleColor.Red;
Console.Write(" \u263b "); // Smiley
Console.ForegroundColor = ConsoleColor.Blue;
Console.WriteLine("\u2551");
Console.WriteLine("\u255a\u2550\u2550\u2550\u255d");

Console.ResetColor();
```



Die Console hat noch etliche weitere Möglichkeiten, z.B.

```
Console.Clear()
Console.SetCursorPosition(x, y)
```

Probieren Sie es mal aus und basteln Sie eine kreative Lösung!