
Buchstaben sortieren

Schreiben Sie eine Funktion

```
func buchstabenSortieren(wort string) string {...}
```

welche die Buchstaben eines Wortes in alphabetischer Reihenfolge zu einem neuen Wort zusammenfügt und dieses zurückgibt.

Im Ausgangswort mehrfach vorkommende Buchstaben sollen im resultierendem Wort genauso oft vorkommen. Die zu verarbeitenden Wörter bestehen nur aus den 26 Buchstaben a bis z bzw. A bis Z. Großbuchstaben im Ausgangswort sollen als Kleinbuchstaben im resultierenden Wort erscheinen.

Beispiele:

Tieflieger -> eeefgiilrt

Golang -> agglno

Hinweise und Tipps

Stöbern Sie in der [strings-Bibliothek](#) nach ggf. hilfreichen Funktionen. Im Lerntext steht, wie Sie einen Teilstring eines Strings erhalten. Ebenfalls ist erklärt, wie Sie auf die einzelnen Bytes eines Strings zugreifen können. Da in den Wörtern nur die Buchstaben a-z und A-Z vorkommen, entspricht jedes Byte einem Buchstaben. In diesem Zusammenhang geben wir Ihnen noch zwei konkretere Tipps, die bei der Erstellung einer Lösung nützlich sein können:

- Sie erhalten einen Buchstaben, dessen Position pos in einem String s Sie kennen, mit `s[pos:pos+1]`.
- Um den Bytewert eines einzelnen Buchstabens zu erhalten, greifen Sie auf das einzige Byte dieses Buchstabens, also das 0. Byte, zu.

```
"tieflieger"[3:4]    // "f"  
"a"[0]              // 97
```

Ihre Funktion soll Teil eines Buchstaben-Sortier-Spiels sein, bei dem aus einem Wort obige sortierte Form gebildet werden soll. Eine einfache Version befindet sich in der Datei `main.go`. Sie präsentiert ein aus einer Liste zufällig ausgewähltes Wort und bewertet die eingegebene Lösung.

Wenn Sie möchten, können Sie dieses Spiel vielfältig erweitern. Beispielsweise

-
- könnten mehrere Wörter (anstatt nur eines) präsentiert werden.
 - könnten bei einem fehlerhaften Versuch weitere Chancen für das gleiche Wort angeboten werden.
 - könnte die Zeit ausgegeben werden, die ein Versuch benötigte.

Sie können das Spiel auch ganz einfach umdrehen: Es wird ein bereits sortiertes Wort präsentiert und es soll das “original” Wort geraten werden, aus welchem das sortierte Wort hervorgegangen ist.

Weiterhin könnte angeboten werden, erzielte Bestzeiten in einer Highscore-Liste einzutragen. Für letzteres müssten Sie sich aber über den Kurstext hinausgehendes Wissen aneignen (Dateioperationen oder Webzugriff, um die Scores dort abzulegen). Die benötigte Zeit hängt offenbar von der Länge des Wortes ab und wie alphabetisch ungeordnet es ist. Insofern wäre ein entsprechender Anpassungsfaktor für die erzielten Zeiten fair, der den Schwierigkeitsgrad widerspiegelt.

Präsentieren Sie Ihre Varianten gerne im Forum, wir sind gespannt!

Ressourcen

Im Ordner dieser Aufgabe finden Sie eine Datei `buchstabenSortieren.go`, die Sie entsprechend der Aufgabenstellung abändern sollen.

Weiterhin stellen wir Ihnen in der Datei `main.go` die oben erwähnte `main`-Funktion zur Verfügung, damit Sie Ihre Funktion kompilieren und ausführen können. Nachdem Sie in den Aufgabenordner gewechselt sind, geben Sie dazu folgenden Befehl ein:

```
go run .
```

Die Datei `buchstabenSortieren_test.go` stellt Tests bereit, die Sie mit folgendem Befehl durchführen können:

```
go test
```

Im Unterordner `ML` finden Sie einen Lösungsvorschlag.