

Aufgabe (Lektion 8): Passwort Generator

Implementieren Sie einen einfachen Passwort-Generator. Nachdem sich Passwörter gut mit Hilfe eines passenden Satzes merken lassen, soll der Passwort-Generator einen String entgegennehmen und ein Passwort (String) zurückgeben. Der Passwort-Generator erstellt auf Basis der Wörter eines Satzes (Leerzeichen als Trennzeichen) ein Passwort. Ein sicheres Passwort soll mind. eines der Zeichen

- Ausrufezeichen
- Fragezeichen
- Doppelpunkt
- Prozentzeichen

und eine Zahl enthalten. Außerdem soll das resultierende Passwort mindestens die Länge 8 und maximal die Länge 15 haben.

- a) Ermitteln Sie das Passwort folgendermaßen:

- (1) Anfangszeichen aller Wörter
- (2) Erste Ziffer einer Zahl (falls Zahl von Leerzeichen eingerahmt wird)
- (3) Die Sonderzeichen können an beliebiger Stelle innerhalb eines Wortes stehen und werden entweder zusätzlich zum Anfangszeichen ergänzt (falls es nicht das erste Zeichen des Worts ist), oder ist bereits durch Regel (1) abgedeckt.

Beispiele:

Ich gehe morgen mit meinen 11 Freund\*innen zum Essen!

Igmmm1FzE!

Fünf Fakultät oder auch 5! Ergibt 120.

FFoa5le1

So wird das nichts!

*nicht gültig (Swdn)*

Hurr! Ich habe 90% der Aufgaben richtig gelöst.

H!ih9%dArg

- b) Erstellen Sie eine weitere Methode, welche das Passwort wie bei a) ermittelt, nur dass nun der letzte Buchstabe eines Wortes verwendet wird.
- (1) Letztes Zeichen eines Worts
  - (2) Letzte Ziffer einer Zahl (falls Zahl von Leerzeichen eingerahmt wird)
  - (3) Die Sonderzeichen können an beliebiger Stelle innerhalb eines Wortes stehen und werden zusätzlich ergänzt (falls es nicht das letzte Zeichen des Worts ist) oder ist bereits durch (1) abgedeckt.

- (4) Außerdem soll alternierend und unabhängig vom eingegebenen Satz der letzte **Buchstabe** eines Worts entweder als Klein- oder Großbuchstabe verwendet werden (beginnend mit einem Kleinbuchstaben).

**Beispiele:**

Ich gehe morgen mit meinen 11 Freund\*innen zum Essen!

hEnTn1Nm!

Fünf Fakultät oder auch 5! Ergibt 120.

*nicht gültig (fTrH!t.)*

Wiederhole 3 mal: Das ! in Java ist der Negierungsoperator der booleschen Algebra.

e3:S!nAtRrRn.

Hurr!a ich habe 90 % der Aufgaben richtig gelöst.

!aHe0%RnG.

**Achtung:** Es sind nur folgende Methoden der String-Klasse erlaubt:

*length, charAt, substring, replace*

Außerdem gelten die im Klausurdeckblatt genannten Einschränkungen (zu Finden im Moodle-Kurs unter ‚Altklausuren und Übungsaufgaben‘).

Vorgehensweise:

- Erstellen Sie sich eine Skizze, welche die notwendigen Teilaufgaben des Programms abbildet.
- Überlegen Sie Methoden, welche zur Lösung der Aufgabe erforderlich sind.  
Benennen Sie diese Methoden konkret mit Rückgabewert, einem Methodennamen (der klar erkennen lässt, WAS die Aufgabe der Methode ist) und der Parameterliste.  
Beispiel: `boolean isSpace (char c)`
- Implementieren Sie zuerst bei den Teilaufgaben a) und b) die Anforderungen (1) und (2). Anschließend erweitern Sie (3) (und (4))
- Testen Sie Ihre Implementierung zusätzlich mit folgenden Beispielsätzen:  
`" Ich gehe morgen mit meinen 11 Freund*innen zum Essen! "`  
`"Hurr!a ich habe 90 % der Aufgaben richtig gelöst."`

## Weitere Hinweise zur Vorgehensweise

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
I c h | g e h e | m o r g e n | m i t | m e i n e n | 1 1 | F r e u n d * i n n e n | z u m | E s s e n .
```

Sehen Sie sich die Übungsaufgabe *StringMethoden* (ebenfalls bei ‚Altklausuren und Übungsaufgaben‘ im Moodle Kurs bei Lektion 8 zu finden) an.

Überlegen Sie, wie das Problem a) mit Hilfe der darauf genannten Methoden lösen können.