

# Aufgabe 1: Wörter aufräumen

---

Team-ID: 00111

Team-Name: Team Name

Marek Freunscht

6. September 2020

## Inhaltsverzeichnis

Lösungsidee.....	1
Unterüberschriften.....	1
Umsetzung .....	1
Beispiele .....	1
Quellcode .....	2

**Anleitung:** Trage oben die Aufgabennummer, die Team-ID, den Team-Namen und alle Bearbeiter/-innen dieser Aufgabe mit Vor- und Namen ein. Vergiss nicht, auch den Aufgabennamen anzupassen (statt „Word-Dokument“)!

Drücke dann F9 zur Aktualisierung, lösche diese Anleitung und lege los ;)

Wenn du neue Überschriften einfügst, solltest du das Inhaltsverzeichnis mittels Rechtsklick aktualisieren.

Die gelb hinterlegten Texte, die hier bereits stehen, geben ein paar Hinweise zur Einsendung. Du solltest sie aber in deiner Einsendung wieder entfernen!

## Lösungsidee

Um alle Wörter gut zuzuweisen, muss man sich zuerst eine geeignete Einteilung der Wörter überlegen. Eine solche Einteilung wäre zum Beispiel nach Wortlänge. In der Einteilung nach Wortlänge kann man über diese iterieren und überprüfen welche Wörter in ein unvollständige Wort passen würden. Wenn dies nur ein Wort ist kann man dieses direkt Abspeichern. Dies würde zum Beispiel in einem Dictionary geschehen, welches unvollständiges Wort zu vollständigem Wort zuordnet. Diese müsste man dann auch aus der aktuellen Liste von vollständigen und unvollständigen Wörtern löschen. Wenn nun aber mehrere Wörter in ein unvollständiges Wort passen kann man dies fürs Erste ignorieren, da durch das Löschen von Wörtern aus der aktuellen Wörterliste sich diese Möglichkeiten letztendlich verringern. Nun muss man so lange über diese aktuelle Längenliste iterieren, bis es keine Wörter mehr zu vergeben gibt. Diese Schritte muss man nun für jede mögliche Wortlänge wiederholen.

## Unterüberschriften

Wenn nötig, füge sinnvoll formatierte Unterpunkte ein, indem du die Formatvorlage „Überschrift 2“ (oder „Heading 2“) anwendest. Vergiss nicht, das Inhaltsverzeichnis anschließend wieder mittels Rechtsklick zu aktualisieren.

## Umsetzung

Um die Gruppierung nach Wortlänge überhaupt umzusetzen, muss man zuerst die maximale Wortlänge herausfinden. Dies war schnell durch ein Sortieren der Wortliste nach Wortlänge und dem Auswählen des letzten Wortes in dieser sortierten Liste getan:

```
int maxLength = availableWords.OrderBy(word => word.Length).Last().Length;
```

Nun kann man über eine Schleife bis zu dieser maximalen Länge an jede Wortlänge kommen. In dieser Schleife werden dann 2 Arrays und 2 Listen initialisiert. Die 2 Arrays (hier `currentAvailableWords` und `currentWordsToFind`) stellen die beiden gesamten Wortlisten für diese Wortlänge dar, während die beiden Listen (hier `notUsedWords` und `notUsedUnfinishedWords`) Kopien dieser Arrays sind und den Zweck haben dass aus ihnen entfernt werden kann. Ob für die gesamten Wörterlisten ein Array oder eine Liste benutzt werden sollten ist in diesem Fall recht irrelevant, nur da sich die Länge von ihnen nicht ändert bevorzuge ich Arrays. Der nächste Teil ist nun der Hauptteil vom Algorithmus. Als äußersten Teil findet man eine While-Schleife:

```
bool finished = false;
while (!finished)
{
    ...
}
```

In dieser While-Schleife wird nun immer wieder über alle Wörter in `currentWordsToFind` iteriert. Die Zuweisungslogik (im nächsten Codeblock '...') erfolgt aber nur für diejenigen Wörter, die sich auch noch in der `notUsedUnfinishedWords` Liste befinden, da nurnoch diese zur Verfügung stehen.

```
for (int i = 0; i < currentWordsToFind.Length; i++)
{
    if (notUsedUnfinishedWords.Contains(currentWordsToFind[i]))
    {
        ...
    }

    if (notUsedUnfinishedWords.Count == 0)
    {
        finished = true;
        break;
    }
}
```

Wenn die Listen mit den noch zur Verfügung stehenden Wörtern leer sind, wird die Schleife abgebrochen.

## Beispiele

Bei den Beispieldateien und zwei eigenen Dateien werden folgende Outputs produziert:

### Raetsel0.txt

\_h \_\_, \_a \_\_r \_\_e \_\_b \_\_!

arbeit eine für je oh was

Output:

oh je, was für eine arbeit!

#### Raetsel1.txt

\_m \_a \_ \_e \_s \_ \_e \_ \_ \_ \_ \_ . D \_ \_a \_ \_i \_ \_u \_ \_ \_ \_e \_ \_n \_ \_u \_ \_ \_ \_  
\_l \_ \_h \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_e \_ \_ .

Am in als das Das die und sehr Leute viele wurde wurde Anfang machte wütend falsche Schritt Richtung  
angesehen Universum erschaffen allenthalben

Output:

Am Anfang wurde das Universum erschaffen. Das machte viele Leute sehr wütend und wurde allenthalben als  
Schritt in die falsche Richtung angesehen.

#### Raetsel2.txt

\_s \_e \_ \_a \_ \_e \_ \_ \_ \_ \_n \_ \_u \_ \_ \_ \_ \_m \_ \_ \_ \_ \_ , \_a \_ \_r \_s \_ \_n \_ \_ \_m \_ \_t \_ \_u \_ \_ \_m  
\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_e \_ \_ \_ \_ \_ \_e \_ \_ .

er in zu Als aus Bett fand sich einem eines Samsa Gregor seinem Morgens Träumen erwachte unruhigen  
Ungeziefer verwandelt ungeheueren

Output:

Als Gregor Samsa eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem  
ungeheueren Ungeziefer verwandelt.

#### Raetsel3.txt

\_ \_ \_ \_ \_ \_ \_t d \_ \_ \_e \_ \_ \_ \_ \_n \_ \_e \_g, \_e \_ \_ \_ \_ \_ , \_a \_ \_ \_ \_ \_ \_n \_  
\_ \_r \_ \_ \_ \_ \_n \_ \_o \_ \_ \_ \_ \_ , \_ \_ \_ \_ \_ \_e \_ \_ \_ \_t \_ \_ \_a \_ \_ \_ \_ \_ \_i \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ .

der der die ist mit und von von besonders Informatik Darstellung Speicherung Übertragung Verarbeitung  
Verarbeitung Wissenschaft Informationen automatischen systematischen Digitalrechnern

Output:

Informatik ist die Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und  
Übertragung von Informationen, besonders der automatischen Verarbeitung mit Digitalrechnern.

#### Raetsel4.txt

\_a \_ \_ \_n \_ \_ \_ \_t \_n \_i \_ \_e \_ \_ \_ \_ \_s d \_ \_ \_e \_ \_d \_i \_ \_e \_ \_t \_ \_ . \_s \_s \_ \_n \_ \_ \_e \_ \_n  
\_ \_ \_n \_ \_e \_ \_ , \_i \_i \_i \_ \_h \_ \_ \_ \_ \_ \_ \_ \_ \_ \_e \_ \_b \_ \_ \_ \_d \_ \_s \_ \_ , \_o \_s \_s \_ \_i \_ \_t \_  
\_e \_ \_ \_ \_ \_r \_ \_ \_ . \_ \_ \_ \_ \_n \_ \_a \_ \_ \_ \_s \_ \_e \_ \_ \_ \_ \_b \_ \_ \_d \_ \_n \_ \_ \_e \_ \_ .

Es in in so aus der die die ein ist Opa sie und und von dass eine eine sind einer Liste schon sowie einige findet  
Jürgen Rätsel sollen werden ergeben gegeben lustige Wörtern Apotheke blättert gebracht richtige Buchstaben  
Geschichte vorgegeben Leerzeichen Reihenfolge Satzzeichen Zeitschrift

Output:

Opa Jürgen blättert in einer Zeitschrift aus der Apotheke und findet ein Rätsel. Es ist eine Liste von Wörtern gegeben, die in die richtige Reihenfolge gebracht werden sollen, so dass sie eine lustige Geschichte ergeben. Leerzeichen und Satzzeichen sowie einige Buchstaben sind schon vorgegeben.

#### Raetsel5.txt

O\_o \_u\_ \_u\_ \_m\_i\_ , d\_r \_c\_ \_f \_n\_ i\_we\_ h\_ \_d\_r\_ü\_ \_r.

seinem kam war er sich hatte ging zur Otto da an und und Hinweg verlaufen Müde auf er nicht Schule

Output:

Otto ging zur Schule und kam nicht an, da er sich auf seinem Hinweg verlaufen hatte und er Müde war.

Hier hat man den Sonderfall das bei einem unvollständigen Wort zwei statt nur einem Buchstaben gegeben sind, was den Algorithmus jedoch nicht behindert.

#### Quellcode

Unwichtige Teile des Programms sollen hier nicht abgedruckt werden. Dieser Teil sollte nicht mehr als 2–3 Seiten umfassen, maximal 10.