

Übungsaufgaben – mit Strings

Ü6_1: Schreiben Sie ein Programm, das einen Buchstaben und eine Zeichenkette einliest und dann abzählt, wie oft der Buchstabe in der Zeichenkette vorkommt.

Ü6_2: Schreiben Sie ein Programm, das einen Buchstaben und eine Zeichenkette einliest und dann die Positionen in der Zeichenkette ausgibt, an denen der eingegebene Buchstabe vorkommt.

Ü6_3: Schreiben Sie ein Programm, das eine kurze und eine lange Zeichenkette einliest und dann überprüft, ob die kurze in der langen Zeichenkette vorkommt. Die Position, an der in der langen Zeichenkette die kurze zum ersten Mal beginnt, soll ausgegeben werden.

Ü6_4: Schreiben Sie ein Programm, das eine Zeichenkette einliest und überprüft, ob alle 26 Kleinbuchstaben des Alphabets darin vorkommen.

Ü6_5: Schreiben Sie ein Programm, das eine Zeichenkette einliest und feststellt, wie oft jeder Vokal (a, e, i, o, u) vorkommt.

Ü6_6: Schreiben Sie ein Programm, das einen Hashcode für eine Zeichenkette berechnet. Verwenden Sie dafür die Divisionsrest-Methode, wobei als Divisor eine Primzahl verwendet werden soll.

Bei der Divisionsrest-Methode wird der jeweils bis dahin ermittelte Hash mit der Zeichensatzgröße multipliziert und das nächste Zeichen dazuaddiert. Von diesem Wert wird der Divisionsrest vom ausgewählten Divisor gebildet. Diese Berechnung wird für alle Zeichen der Zeichenkette wiederholt.

Bei einer Zeichensatz-Größe von 128 und der Primzahl p gilt:

$$h = (\dots (s[1] * 128 + s[2]) \text{ MOD } p) * 128 + \dots s[\text{len}-1]) \text{ MOD } p * 128 + s[\text{len}]) \text{ MOD } p$$

z.B. für die Zeichenfolge "Java"

$$h = (((('j' * 128 + 'a') \% 1021 * 128 + 'v') \% 1021 * 128 + 'a') \% 1021$$