

Übungsblatt 5

Aufgabe 1)

a)

Erstelle eine Funktion, welche einen Satz und ein Zeichen entgegennimmt. Die Funktion soll zählen, wie oft das Zeichen in dem Satz vorkommt.

Bsp.

```
String satz = "Wow! Was fuer ein toller Satz";  
  
Console.WriteLine("Anzahl an e: " + countChar(satz, 'e'));  
Console.WriteLine("Anzahl an Leerzeichen: " + countChar(satz, ' '));
```

Ausgabe:

```
Anzahl an e: 3  
Anzahl an Leerzeichen: 5
```

b)

Erstelle eine Funktion, welche einen Satz und ein Zeichen entgegennimmt. Die Funktion soll den Satz so umschreiben, dass das Zeichen nicht mehr darin vorkommt.

Bsp.

```
Console.WriteLine("Satz ohne e: " + deleteChar(satz, 'e'));
```

Ausgabe:

```
Satz ohne e: Wow! Was fur in tollr Satz
```

c)

Erstelle eine Funktion welche einen Satz entgegennimmt und jedes Wort in einer einzelnen Zeile ausgibt.

Bsp.

```
words(satz);
```

Ausgabe:

```
Wow!  
Was  
fuer  
ein  
toller  
Satz
```

Aufgabe 2)

- a) Erzeugen Sie ein zweidimensionales Feld `f` mit beliebiger, durch Sie festgelegter Größe, z.B. 3 Zeilen, 5 Spalten.

Befüllen Sie das Feld mit Zufallszahlen zwischen 0 und 9.

- b) Schreiben Sie eine Funktion `Ausgabe`, die ein zweidimensionales Integer-Feld auf der Console ausgibt. Den Code dafür können Sie i.W. aus dem Handout zu mehrdimensionalen Arrays oder der Vorlesung übernehmen.

Der Funktionskopf sollte z.B. so aussehen:

```
static void Ausgabe(int[,] f)
```

Lassen Sie damit das Array aus der ersten Teilaufgabe ausgeben.

- c) Schreiben Sie eine Funktion `"Transponiere"`, welche die Zeilen und Spalten eines beliebig großen zweidimensionalen Feldes vertauscht (transponiert), d.h. folgende Transformation soll vorgenommen werden:

$$\begin{bmatrix} 11 & 12 \\ 21 & 22 \\ 31 & 32 \end{bmatrix} \Rightarrow \begin{bmatrix} 11 & 21 & 31 \\ 12 & 22 & 32 \end{bmatrix}$$

Die Funktion soll ein zweidimensionales Array als Parameter bekommen und ein neues zweidimensionales Array erzeugen und als Ergebnis zurückgeben. Der Funktionskopf sollte also so aussehen:

```
static int[,] Transponiere(int[,] f)
```

Testen Sie die Funktion mit dem Feld aus der ersten Aufgabe.

```
int[,] f2 = Transponiere(f);  
Console.WriteLine("Transponiert:");  
Ausgabe(f2);
```

- d) Ausgabe mit Spaltensumme

Stellen Sie sich vor, Ihr Feld würde die Zahlen in einem Excel-Arbeitsblatt Verkäufe oder Lagerbestände beinhalten. In Excel muss häufig eine Zeilen- oder Spaltensumme errechnet werden. Das machen wir jetzt auch.

Erweitern Sie dafür Ihre Ausgabefunktion von oben: Geben Sie in einer Summenzeile unterhalb des 2D-Feldes die jeweiligen Spaltensummen aus.

```
3  5  6  
7  8  2  
4  5  3  
9  7  8  
-----  
23 25 19
```

- e) Damit die Spaltensumme nicht immer ausgegeben wird, ergänzen Sie in der Ausgabefunktion einen optionalen Bool-Parameter (default false).

Nur, wenn dieser explizit mit `true` angegeben wird, soll die Summe ausgegeben werden.