

Aufgabe 1: "Text-Analyser"**[24 Punkte]**

Gesucht ist ein Programm, welches es ermöglicht einen Text zu analysieren.

Bei der Analyse soll festgestellt werden, wie oft bestimmte Zeichen, die im Parameter **analyseChars** gegeben sind, im Parameter **text** vorkommen. Das Ergebnis soll die Anzahl der jeweiligen Zeichen aus **analyseChars** an der jeweiligen Position zurückgeben (siehe (B)).

Mit dem Parameter **case** soll eingestellt werden können ob die Analyse case sensitive ist oder nicht (**true** = case sensitive, **false** = case insensitive).

Realisiere Sie dazu insbesondere die folgenden Methoden und vervollständigen Sie das am Schluss gegebene Programm:

A) printText()-Methode**[4 / 24 Punkte]**

Realisieren Sie dazu eine Methode:

```
void printText(char[] text, char[], analyseChars, boolean caseSensitive)
```

Welche den Text sowie die Analyse-Zeichen, wie im Beispiel gezeigt ist, ausgibt.

Beispiel:

```
text = {'D','a','s',' ','i','s','t',' ','d','e','r',' ','T','e','x','t'}
analyseChars = {'T', 'e', 'x', 't', '!' }
caseSensitive = false
```

Gibt aus:

Das ist der Text

wird bzgl.

Text!

analysiert (case insensitive)!

Die Methode soll den Text "--- keine gueltigen Analyse-Daten ---" ausgeben, für den Fall, dass die Parametrisierung keine sinnvollen Werte beinhaltet.

B) analyseText()-Methode**[10 / 24 Punkte]**

Realisieren Sie dazu eine Methode:

```
int[] analyseText(char[] text, char[], analyseChars, boolean caseSensitive)
```

Diese Methode soll den Text bzgl. des Auftreten von Zeichen (wie oben beschrieben) analysieren.

Beispiele:

```
text = {'D','a','s',' ','i','s','t',' ','d','e','r',' ','T','e','x','t'}
analyseChars = {'T', 'e', 'x', 't', '!' }
caseSensitive = false
```

Gibt zurück:

```
{3, 2, 1, 3, 0}
```

```
text = {'D','a','s',' ','i','s','t',' ','d','e','r',' ','T','e','x','t'}
analyseChars = {'T', 'e', 'x', 't', '!' }
caseSensitive = true
```

Gibt zurück:

```
{1, 2, 1, 2, 0}
```

Die Methode soll **null** zurückgeben, im Fall, dass die Parametrisierung keine sinnvollen Werte beinhaltet.

C) printAnalysis()-Methode**[10 /24 Punkte]**

Weiteres realisieren Sie eine Methode:

```
void printAnalysis(char[] analyseChars, int[] analysisResults, Boolean caseSensitive)
```

Welche das Analyse-Ergebnis formatiert ausgibt.

Beispiel:

```
analyseChars = {'T', 'e', 'x', 't', '!'}  
analysisResults = {3, 2, 1, 3, 0}  
case = false
```

Gibt aus:

```
Analyse (case insensitive)  
-----  
T: 3  
e: 2  
x: 1  
t: 3  
!: 0
```

Die Methode soll den Text **"--- keine gueltigen Analyse-Daten ---"** ausgeben, für den Fall das die Parametrisierung keine sinnvollen Werte beinhaltet.

Vervollständigen Sie schlussendlich folgendes Programm:

```
class TextAnalyser {
    public static void main(String [] args) {

        Out.print("\nText-Analyser");
        Out.print("\n=====");

        // Eingabe des Texts
        Out.print("\nBitte zu analysierenden Text eingeben: ");
        char [] text = readTextAsCharArray();

        // Eingabe des AnalyseTexts
        Out.print("\nBitte Analyse Text eingeben: ");
        char [] analyseChars = readTextAsCharArray();

        // Eingabe ob case Sensitive
        Out.print("\nGross- / Kleinschreibung beachten: ");
        boolean caseSensitive = In.readBoolean();

        // Ausgabe des Texts (A)
        printText(text, analyseChars, caseSensitive);

        // Analyse des Texts (B)
        int[] analysisResults;
        analysisResults = analyseText(text, analyseChars, caseSensitive);

        // Ausgabe des Analyse-Ergebnisses (C)
        printAnalysis(analyseChars, analysisResults, caseSensitive);

    } // main()

    static char [] readTextAsCharArray() {
        String text = In.readString();
        if (text == null) return null;
        if (text.length() <= 0) return null;
        return text.toCharArray();
    } // end inputTextAsCharArray()

    // TODO -----
} // end TextAnalyser
```

Abgabe:

→ Mit Ihrem Code **vervollständigtes JAVA-Programm "Text-Analyser"**.