

#### 19.6.4 Prüfung, ob bestimmte Zeichen in einer Zeichenkette vorhanden sind

Mit den folgenden drei Prozeduren kann geprüft werden, ob in einer vorgegebenen Zeichenkette jeweils mindestens ein Zeichen aus einem vorgegebenem Zeichenvorrat vorhanden ist. Die Prozeduren kommen zum Beispiel bei der Prüfung von Passwörtern zum Einsatz, wenn ein starkes Passwort gefordert wird.

Ein starkes Passwort kann m.E. so definiert werden kann:

- Das Passwort besteht aus mindestens 8 Zeichen.
- Das Passwort enthält mindestens 1 Großbuchstaben.
- Das Passwort enthält mindestens 1 Kleinbuchstaben.
- Das Passwort enthält mindestens 1 Ziffer.
- Das Passwort enthält mindestens ein Sonderzeichen aus einem definierten Zeichenvorrat.

Für die Arbeit mit regulären Ausdrücken muss die Komponente *gb.pcre* in das Gambas-Projekt eingebunden sein. Zuerst wird die Funktion *Match(Subject As String, Pattern As String)* codiert

```
Public Function Match(Subject As String, Pattern As String) As Boolean
    Dim rRegex As Regexp

    rRegex = New Regexp(Subject, Pattern)

    If rRegex.Offset = -1 Then
        Return False
    Else
        Return True
    Endif ' rRegex.Offset = -1 → kein Treffer ?
End ' Match(Subject As String, Pattern As String) As Boolean
```

und für die Prüfung eingesetzt:

```
Private Function CheckStrongPassword(sPasswort As String) As Boolean
    Dim sSubject, sPattern As String

    sSubject = sPasswort
    sPattern = "(?=^.{8,}$)(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?![\.\n])(?=.*[+#_@!?$%*]).*$"

    If Match(sSubject, sPattern) = True Then
        Return True
    Else
        Return False
    Endif ' Match(sSubject, sPattern) = True ?
End ' CheckStrongPassword(sPasswort As String) As Boolean
```

In ähnlicher Weise kann ermittelt werden, ob der optional eingesetzte Präfix bei der Verschlüsselung einer Zeichenkette nach dem DES-Algorithmus korrekt festgelegt worden ist. Der Präfix ist genau zwei Zeichen lang und enthält Ziffern oder kleine Buchstaben oder Großbuchstaben oder ein Sonderzeichen aus der Menge *{./}*:

```
Private Function CheckPrefix(sPrefix As String) As Boolean
    Dim sSubject, sPattern As String
    ' sPräfix-Zeichenvorrat = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz./"

    sSubject = sPrefix
    sPattern = "^([a-zA-Z0-9./]{2})$" ' ** Test-Muster

    If Match(sSubject, sPattern) = True Then
        Return True
    Else
        Return False
    Endif ' Match(sSubject, sPattern) = True ?
End ' CheckPrefix(sPrefix As String) As Boolean
```

Die Entwicklung regulärer Ausdrücke ist *kein Inhalt* des Kapitels. Es werden nur geprüfte Muster aus eigenen Gambas-Projekten übernommen. Der reguläre Ausdruck in *\*\** ist noch einfach – sagen jene, die sich intensiv mit regulären Ausdrücken beschäftigen haben.

Hier kommt der Nachweis:

- ^ Treffer nur dann, wenn hier der zu durchsuchende Präfix anfängt
- () Ausdruck in der Klammer als ein Element auffassen, auf das sich {2} bezieht
- [a-zA-Z0-9./] Klein- oder Großbuchstabe oder Ziffer 0..9 oder Zeichen . oder /
- {2} Genau 2 Zeichen aus der Menge [a-zA-Z0-9./]
- \$ Treffer nur dann, wenn hier der zu durchsuchende Präfix endet