28.2.2 Anwendung Crypt.MD5

In diesem Kapitel lernen Sie eine Anwendung kennen, die Methoden der Klasse *Crypt* verwendet. Es sind die Methoden *Crypt.MD5* und *Crypt.Check(Password As String, Crypt As String)*. Mit der Methode *Crypt.Check(..)* können Sie prüfen, ob ein eingegebenes Passwort – das intern nach dem Algorithmus MD5 verschlüsselt wird – mit einem im Programm oder in einer Datei hinterlegten verschlüsselten Passwort übereinstimmt oder nicht. Das vorgestellte Programm ist durch ein starkes (Start-)Passwort (*+Gambas340*) geschützt:

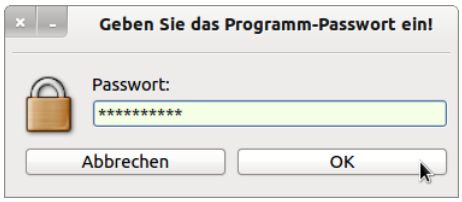

Abbildung 28.2.2.1: Passwort-Eingabe (Klartext)

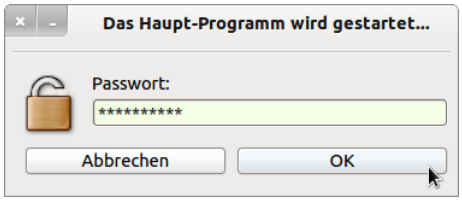

Abbildung 28.2.2.2: Das Passwort ist korrekt

Das verwendete Start-Passwort wurde mit dem MD5-Passwort-Generator mit zufälligem Präfix generiert. Dieses *verschlüsselte* Passwort wird im Programm fest hinterlegt. Außerdem wurden folgende Erweiterungen implementiert:

- Der Passwort-Check lässt nur 3 ungültige Eingaben zu. Danach wird die Passwort-Abfrage beendet und das geschützte Hauptprogramm nicht gestartet.
- Nach erfolgreicher Passworteingabe *kann* der Benutzer im Hauptprogramm zur Laufzeit das Programm-Passwort *ändern*.

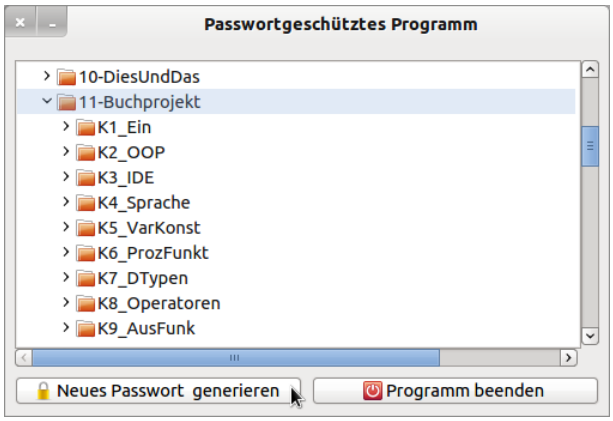

Abbildung 28.2.2.3: Das Hauptprogramm wurde erfolgreich gestartet ...

Im folgenden Abschnitt werden nur Auszüge aus dem Quelltext vorgestellt und kurz erläutert.

```
' Gambas class file
' Die Komponenten CRYPT und SETTINGS müssen eingebunden werden.

Public iCount As Integer = 1
Public sMD5Passwort As String
Public bPWAbbruch As Boolean = False
Public bPWEingabeFehler As Boolean = False
Public pw2Settings As New Settings(Application.Path &/ ".pw.conf")
' Hinweis: Die Datei .pw.conf ist eine versteckte Datei im Anwendungspfad

Public Sub Form_Open()
```

```
  FGetPassword.Center
  FGetPassword.Resizable = False

  PictureBox1.Picture = Picture["Symbols/schloss_zu.png"]
  txtPasswortEingabe.Password = True
  txtPasswortEingabe.Clear

  sMD5Passwort = pw2Settings["Passwort/MD5-Passwort", "$1$V/eCyFQp$hDAEMfcO7yuN3o0UFfkKL0"]

End ' Form_Open

Public Sub btnAbbrechen_Click()
  bPWAbbruch = True
  FGetPassword.Close
End ' btnAbbrechen_Click()

Public Sub txtPasswortEingabe_Activate()
  btnOK_Click()
End ' Eingabe_Activate

Public Sub btnOK_Click()
  Dim sPassword, sMessage As String

  sPassword = txtPasswortEingabe.Text
  If sPassword = "" Then
     Message.Info("Geben Sie ein Passwort ein!")
     Return
  Endif ' sPassword = "" ?

  If Crypt.Check(sPassword, sMD5Passwort) = True Then

     If iCount <= 2 Then
        Message.Warning("Achtung!\nDas Passwort ist NICHT korrekt!")
        txtPasswortEingabe.Clear
        txtPasswortEingabe.SetFocus
        Inc iCount
     Else
        sMessage = "Fehler!\n"
        sMessage &= "Das Passwort ist auch nach 3 Eingaben NICHT korrekt!\n"
        sMessage &= "Die Password-Abfrage wird beendet."
        Message.Error(sMessage)
        bPWEingabeFehler = True
        FGetPassword.Close
     Endif ' iCount <= 2 ?
  Else
     bPWEingabeFehler = False
     FGetPassword.Close
  Endif ' Crypt.Check(sPassword, sMD5Passwort) = True ?

End ' btnOK_Click

Public Sub Form_Close()

  If bPWAbbruch = True Or bPWEingabeFehler = True Then
     FMain.PasswordError = True
  Else
     PictureBox1.Picture = Picture["Symbols/schloss_auf.png"]
     FGetPassword.text = "Das Haupt-Programm wird gestartet..."
     Wait 2
  Endif ' Fehler ?

End ' Form_Close()
```

Kommentare:

- Beim 1. Programmstart und den folgenden wird das *im Programm hinterlegte* md5-verschlüsselte Passwort verwendet → "1V/eCyFQp$hDAEMfcO7yuN3o0UFfkKL0", solange kein neues Passwort vom Anwender generiert wurde.
- Das Abfrage-Programm wird beendet, wenn der Benutzer die Passwort-Abfrage beendet oder wenn nach 2 Fehleingaben auch die 3. Eingabe einen Passwortfehler ergab oder das korrekte Passwort eingegeben wurde.
- Die Funktion Crypt.Check(...) liefert True, wenn das Passwortpaar NICHT übereinstimmt (!) und False, wenn das verschlüsselt hinterlegte Passwort mit der md5-verschlüsselten Klartext-Eingabe übereinstimmt!

- Die Abfrage in der Prozedur *Form_Close* ist wichtig, weil unter keinen Umständen die Passwortabfrage umgangen werden darf.

Im Hauptprogramm kann vom Anwender ein neues Passwort generiert werden:

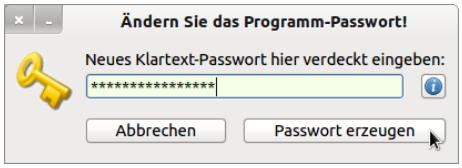

Abbildung 28.2.2.4: Ein neues Programm-Passwort wird generiert

Genutzt wird ein modifizierter MD5-Passwort-Generator, der bereits im Kapitel 28.2.1 beschrieben wurde. Die Komponente *gb.settings* wird für das komfortable Auslesen und Speichern des Referenz-Passwortes eingesetzt. Das neue, starke Passwort – nach MD5 verschlüsselt – wird in der *Konfigurationsdatei .pw.conf* im Anwendungsverzeichnis gespeichert.