

12.2.6 Form - Persistenz

Stellen Sie sich die folgende Situation vor: Sie öffnen ein (Haupt-)Fenster FMain und öffnen aus diesem heraus ein weiteres Fenster F2. Dann bearbeiten Sie ausgewählte Daten im Fenster F2. Danach schließen Sie das Fenster F2.

Welche Antwort hätten Sie auf die Frage: Wie realisieren Sie den (notwendigen) Datenaustausch zwischen dem Fenster F2 und dem (Haupt-)Fenster FMain?

Normalerweise gehen nach dem Schließen von F2 alle Daten in Bezug auf das Formular F2 wie zum Beispiel Text in einer TextBox oder die Werte von Variablen in der Klasse F2 verloren. Das können Sie verhindern, indem Sie für das Formular F2 die Eigenschaft *F2.Persistent* auf 'True' setzen. Das ist einfach und wirkungsvoll, denn Sie können jederzeit – auch wenn das Fenster F2 geschlossen wurde – auf die Daten der Objekte im Fenster F2 zugreifen.

Folgende Erfahrungen zur Arbeit mit der Eigenschaft *Form.Persistent* können vermittelt werden:

- Persistenz verwendet man vor Allem für Dialog-Fenster, die zum Beispiel dem Einlesen oder Erfassen von Daten dienen.
- Sie müssen die Public-Eigenschaft von Steuer-Elementen, auf deren Eigenschaften und Methoden u.U. wechselseitig von den beiden Formularen FMain und F2 zugegriffen wird, auf den Wert 'True' setzen. Achtung: Das funktioniert nur im Formular-Editor der IDE; zur Laufzeit gelingen diese Änderungen der virtuellen Public-Eigenschaft von Steuer-Elementen nicht!
- Wenn Timer für ein Dialog-Formular eingesetzt werden, dann laufen diese nach dem Schließen der Form nur weiter, wenn die Form persistent ist! Das gilt auch für andere Steuer-Elemente, die Ereignisse produzieren.
- Jede Aktion mit dem Dialog löst eine Form-Persistenz für alle Objekte einer Form aus, auch wenn Persistent auf 'False' steht.
- Praktisch sind alle Objekte einer Form grundsätzlich persistent – unabhängig von der Persistent-Eigenschaft. Die Eigenschaft Persistent verhindert nur, dass die Eigenschaften von Objekten bei *F2.Close()* gelöscht werden.
- *F2.Hide()* hat die gleiche löschende Wirkung wie *F2.Close()*.
- Es ist offensichtlich so, dass man die Funktionalität eines Fensters F2, dessen Persistent-Eigenschaft auf 'True' gesetzt wurde, sofort (persistent) nutzen kann, ohne das Fenster F2 auf dem Monitor gesehen zu haben! Ein Hintergrund-Programm im wahren Sinne der Bedeutung.
- Die Eigenschaft *Utility* darf für das (Haupt-)Fenster nicht auf 'True' gesetzt werden, weil dann das Fenster F2 nicht angezeigt wird!

12.2.6.1 Projekt

Im (Haupt-)Fenster FMain können Sie die Eigenschaft *F2.Persistent* über eine CheckBox ändern. Es werden auch die aktuellen Werte für den Bereich der Zufallszahlen aus den beiden Steuer-Elemente (TextBox First, TextBox Last) auf F2 angezeigt. Um das zu realisieren, ist für beide SpinBars auf F2 die Public-Eigenschaft auf den Wert 'True' gesetzt worden.

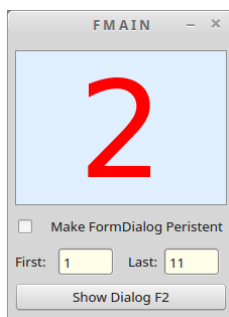


Abbildung 12.2.6.1.1: (Haupt-)Fenster

Im vorgestellten Projekt erzeugt die Klasse F2 nur Zufallszahlen, deren Bereich im Fenster F2 über zwei Steuer-Elemente (SpinBar) geändert werden kann. Das Fenster F2 wird beim Öffnen des (Haupt-) Fensters nicht angezeigt.

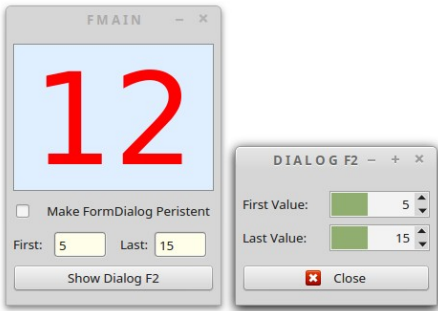


Abbildung 12.2.6.1.2: (Haupt-)Fenster mit geöffnetem Dialog-Fenster F2

Wenn Sie jetzt die Werte für den Bereich der Zufallszahlen ändern, dann werden die geänderten Werte beim nächsten Timer-Takt im (Haupt-)Fenster angezeigt. Nach dem Schließen von F2 wird Sie das Ergebnis nicht überraschen – die Änderungen wurden nicht übernommen:

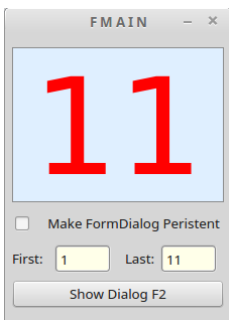


Abbildung 12.2.6.1.3: (Haupt-)Fenster wieder mit den Vorgabewerte-Werten!

Alle aktualisierten Daten in F2 sind gelöscht und auf die Vorgabewerte zurückgestellt, weil die Persistent-Eigenschaft von F2 beim Start des (Haupt-)Fensters auf den Wert 'False' eingestellt war. Das wird geändert, wie Sie es an der markierten CheckBox sehen können:

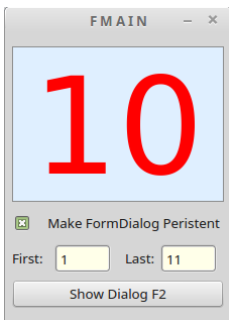


Abbildung 12.2.6.1.4: (Haupt-)Fenster mit Vorgabe-Werten

Im Dialog werden nun neue Werte für den Bereich der Zufallszahlen eingestellt:

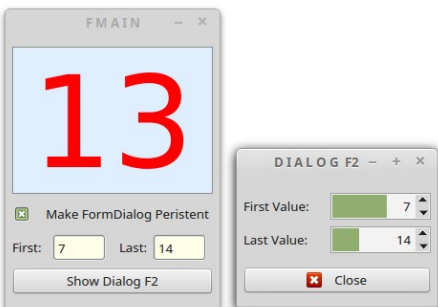


Abbildung 12.2.6.1.5: (Haupt-)Fenster und Dialog-Fenster F2



Abbildung 12.2.6.1.6: (Haupt-)Fenster und Dialog-Fenster F2

Mit den übernommenen, neuen Dialog-Werten werden jetzt von F2 nur noch Zufallszahlen von 7 bis 14 erzeugt und im (Haupt-)Fensters angezeigt. Achtung: Eine Änderung des Zustandes der CheckBox bewirkt nicht automatisch eine Änderung in der Funktionalität von F2! Das erreichen Sie nur durch einen erneuten Aufruf von F2, nachdem der Wert für die CheckBox auf 'nicht ausgewählt' gesetzt wurde und durch das Schließen von F2:



Abbildung 12.2.6.1.7: (Haupt-)Fenster

Der Quelltext wird für beide Klassen angegeben und kommentiert:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]   FMain.Resizable = 'False'
[5] ' FMain.Utility = 'True' ' FEHLER!
[6]   Timer1.Delay = 1000
[7]   Timer1.Enabled = 'True'
[8]   Timer1.Trigger()
[9]   F2.Persistent = 'False'
[10] End
[11]
[12] Public Sub btnShowFormP_Click()
[13]   If ckBoxPersistence.Value Then
[14]     F2.Persistent = 'True'
[15]   Else
[16]     F2.Persistent = 'False'
[17]   Endif
[18]   F2.Showmodal()
[19] End
[20]
[21] Public Sub Timer1_Timer()
[22]   txbFirst.Text = F2.sbarFirst.Value
[23]   txbLast.Text = F2.sbarLast.Value
[24]   txlRandomNumber.Text = F2.SetRandomNumber()
[25] End
```

Kommentar:

- Die Zeile 9 stellt sicher, dass die Persistent-Eigenschaft von F2 beim Öffnen des (Haupt-)Fensters auf 'False' eingestellt ist.
- Der Timer (Zeilen 6 bis 8) erzeugt den Takt, in dem die Zufallszahlen (Zeile 24) als Funktionswert der Funktion SetRandomNumber() aus der Dialog-Klasse abgerufen und angezeigt werden sowie in den Zeilen 22 und 23 die Vorgabewerte für den aktuellen Bereich der Zufallszahlen.

- In den Zeilen 12 bis 19 wird die Persistent-Eigenschaft von F2 festgelegt und dann anschließend das Dialog-Fenster F2 modal angezeigt wird.

Quelltext F2.class:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]     F2.Center()
[5] End
[6]
[7] Public Function SetRandomNumber() As Integer
[8]     Randomize
[9]     Return Rand(sbarFirst.Value, sbarLast.Value)
[10] End
[11]
[12] Public Sub btnClose_Click()
[13]     F2.Close()
[14] End
```

Kommentar:

- In der Zeile 9 werden durch die Funktion *SetRandomNumber()* Zufallszahlen im vorgegebenen Bereich erzeugt. Die *Rand(First, Last)*-Funktion generiert Zahlen vom Typ Integer!
- Der Zufallsgenerator wird durch *Randomize* mit dem aktuellen Zeitstempel initialisiert, wenn der optionale Parameter fehlt.