

9.5 Zufallszahl-Funktionen

In vielen theoretischen Untersuchungen werden Zustände, Zustandsänderungen oder Werte definierter Prozessgrößen wie zum Beispiel bei Untersuchungen zu Warteschlangen simuliert. Benötigt werden für diese Simulationen zufällige Ereignisse, die mit Zufallszahlen in ausgewählten Intervallen realisiert werden, welche mit Zufallsgeneratoren erzeugt werden. In den mitgelieferten Beispielen zu Gambas befindet sich das Projekt Datenbank/Database, in dem mit mehreren Tausend Zufallsdaten gearbeitet wird.

Ein Zufallsgenerator liefert keine reellen Zahlen, die chaotisch verteilt sind, sondern Zahlen, die nach einem bestimmten Algorithmus berechnet werden. Das bedeutet aber auch, dass solche Generatoren einerseits Startwerte benötigen und andererseits nur reelle Pseudo-Zufallszahlen generieren.

9.5.1 Initialisierung

Syntax: `RANDOMIZE [Seed AS Integer]` mit Seed als optionalem Parameter

Der Zufallsgenerators kann mit einem frei wählbaren ganzzahligen Startwert initialisiert werden, wenn beim Aufruf von *Randomize* der optionale Parameter auf einen festen Integer-Wert gesetzt wird. Folgende Ansätze können genutzt werden:

```
Randomize -1 ; auch Randomize (-1) ist korrekt
Randomize 3
```

Nutzen Sie den optionalen Parameter nicht, dann wird für Seed der Zeitstempel gesetzt, sonst wird der ganzzahlige Wert von Seed genutzt.

9.5.2 Zufallszahlen

Auch in Gambas können Sie einen Zufallsgenerator nutzen, der Ihnen in der Standardkonfiguration Zufallszahlen im Bereich $0 \leq \text{Zufallszahl} < 1$ liefert, wobei die Initialisierung durch den aktuellen Zeitstempel erfolgt.

9.5.2.1 Zufallszahlen in Intervall $0 \leq \text{Zufallszahl} < 1$

```
DIM fZufallszahl AS Float
Randomize
fZufallszahl = Rnd()
```

9.5.2.2 Zufallszahlen in Intervall $0 \leq \text{Zufallszahl} < R_{\min}$

```
DIM fZufallszahl AS Float
Randomize
fZufallszahl = Rnd(3)
```

9.5.2.3 Zufallszahlen in Intervall $R_{\min} \leq \text{Zufallszahl} < R_{\max}$

```
DIM fZufallszahl AS Float
Randomize
fZufallszahl = Rnd(-2,+2)
```

9.5.2.4 Ganzzahlige Zufallszahlen in Intervall $R_{\text{unten}} \leq \text{Zufallszahl} \leq R_{\text{oben}}$

```
DIM fZufallszahl As Float
DIM iObergrenze, iUntergrenze As Integer
Randomize
fZufallszahl = Int(Rnd(iUntergrenze, iObergrenze + 1))
```

9.5.3 Beispiele

Die folgenden Beispiele demonstrieren den Einsatz der 2 Zufallszahl-Funktionen in unterschiedlichen Einsatzfällen.

Beispiel 1

Im ersten Beispiel werden Zufallszahlen mit den Zufallszahl-Funktionen *Rnd* und *Randomize* erzeugt.

Es wird die Wirkung der Festlegung eines Startwertes demonstriert. Der Quelltext ist kurz und wird deshalb vollständig angegeben:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]     FMain.Center
[5]     FMain.Resizable = False
[6] End ' Form_Open()
[7]
[8] Public Sub btnRnd1_Click()
[9]     Dim fZufallszahl As Float
[10]    Dim iCount As Integer
[11]
[12]    TextArea1.Clear
[13]    Randomize ' optional, weil Standard
[14]    For iCount = 1 To 10
[15]        If iCount < 10 Then
[16]            TextArea1.Insert(iCount & gb.Tab & Rnd() & gb.NewLine)
[17]        Else
[18]            TextArea1.Insert(iCount & gb.Tab & Rnd())
[19]        Endif
[20]    Next ' iCount
[21] End ' Rnd1
[22]
[23] Public Sub btnRnd2_Click()
[24]     Dim fZufallszahl As Float
[25]     Dim iCount As Integer
[26]
[27]     TextArea1.Clear
[28]     For iCount = 1 To 10
[29]         Randomize
[30]         If iCount < 10 Then
[31]             TextArea1.Insert(iCount & gb.Tab & Rnd() & gb.NewLine)
[32]         Else
[33]             TextArea1.Insert(iCount & gb.Tab & Rnd())
[34]         Endif
[35]     Next ' iCount
[36] End ' Rnd2
[37]
[38] Public Sub btnRnd3_Click()
[39]     Dim fZufallszahl As Float
[40]     Dim iCount As Integer
[41]
[42]     TextArea1.Clear
[43]     Randomize -2
[44]     For iCount = 1 To 10
[45]         If iCount < 10 Then
[46]             TextArea1.Insert(iCount & gb.Tab & Rnd() & gb.NewLine)
[47]         Else
[48]             TextArea1.Insert(iCount & gb.Tab & Rnd())
[49]         Endif
[50]     Next ' iCount
[51] End ' Rnd3
[52]
[53] Public Sub btnClose_Click()
[54]     FMain.Close
[55] End ' Close
```

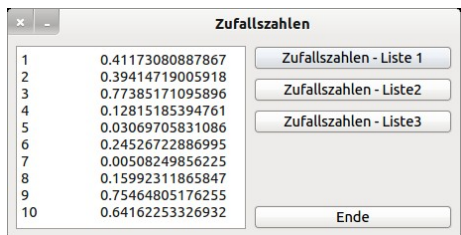


Abbildung 9.5.3.1: Liste mit 10 unterschiedlichen Zufallszahlen

Ergebnisse:

Die Prozedur in den Zeilen 8-22 liefert stets eine Liste mit 10 unterschiedlichen Zufallszahlen. Die zweite Prozedur in 23-36 liefert nur Zufallszahlen, die Serien gleicher Zahlen enthalten kann – im Extremfall sind alle Zufallszahlen gleich. Der Grund liegt darin, dass der Zufallsgenerator für eine Sekunde den gleichen Startwert hat. Die Zufallszahlen in der 3. Prozedur in 41-58 sind zwar unterschiedlich, jedoch ist die Liste der erzeugten Zufallszahlen stets gleich, weil immer der gleiche Startwert für den Zufallsgenerator gesetzt wird.

Beispiel 2

Für Testzwecke benötigt man oft Zufallsdaten, die nicht notwendigerweise Zahlen sein müssen. Im zweiten Beispiel werden für den Test zur Sortierung von Daten in einer GridView Zufallsdaten vom Datentyp Boolean, Float, Integer, String und Date benötigt.

In den folgenden Zeilen wird nur die Prozedur zur Generierung der Zufallsdaten vorgestellt:

```
Private hSort As New CSort
Private iCount As Integer

Public Sub btnFillGrid_Click()
    Dim i As Integer
    Dim vMatrix As Variant[]
    Dim aStrs As String[] = {"Merkur", "Venus", "Erde", "Mars", "Jupiter", "Saturn", "Uranus", "Neptun", "Pluto"}

    Randomize

    For i = 0 To iCount Step 1
        vMatrix = New Variant[]
        vMatrix.Add(Round(Rnd(0, 10), -2)) ' Reelle Zahl
        vMatrix.Add(CBool(Round(Rnd(0, 1)))) ' Wahrheitswert
        vMatrix.Add(aStrs[CInt(Rnd(0, aStrs.Count))]) ' Zeichenkette
        vMatrix.Add(CDate(Rnd(CFloat(Now()), CFloat(Now() + 1000)))) ' Datum 1
        vMatrix.Add(CInt(Rnd(-10, 10))) ' Ganze Zahl
        vMatrix.Add(CDate(Rnd(CFloat(Now()), CFloat(Now() + 2000)))) ' Datum 2
        hSort.Add(vMatrix)
    Next i
    ArrayToGrid()
End ' FillGrid
```

Number	Boolean	String	Date1	Number2	Date2
0,02	True	Uranus	09.12.2013 00:16:30	-2	19.11.2013 21:38:58
0,02	False	Erde	16.04.2013 15:20:28	1	25.10.2016 09:43:14
0,04	True	Neptun	30.12.2012 16:29:24	7	17.07.2013 01:33:50
0,06	True	Merkur	14.04.2013 22:55:36	-9	25.07.2017 19:30:18
0,06	False	Saturn	21.03.2015 07:17:48	-6	24.11.2012 03:58:03
0,08	True	Mars	26.05.2014 00:25:58	2	27.03.2014 10:33:39
0,08	True	Mars	08.05.2013 14:22:29	5	31.05.2016 22:02:45
0,1	False	Saturn	19.02.2015 11:04:50	3	14.02.2017 04:31:57
0,13	False	Neptun	12.06.2014 19:52:32	-2	26.09.2014 15:07:36
0,13	True	Uranus	29.09.2013 12:44:45	-1	08.03.2015 06:42:17

Abbildung 9.5.3.2: GridView mit zufällig erzeugten Datensätzen, sortiert nach der 1. Spalte

Die vollständigen Projekte für die ersten beiden Beispiele werden im Download-Bereich zur Verfügung gestellt.

Beispiel 3

Unter http://de.wikibooks.org/wiki/Gambas:_Zufall#Zufallsquadrante finden Sie ein kleines Projekt, bei dem sowohl die Koordinaten der zu zeichnenden Quadrate als auch deren Farbe zufällig berechnet werden.