LibreOffice
The Document Foundation

Calc 4.1 Handbuch

Kapitel 12
Arbeiten mit Makros

Automatisierung wiederkehrender Aufgaben

Copyright

Mitwirkende/Autoren

Robert Großkopf | Monika Leibold | Jochen Schiffers

Wilhelm Schulz

Englisches Originaldokument

John A Smith | Jean Hollis Weber | Martin J Fox

Andrew Pitonyak | Simon Brydon | Gabriel Godoy

Barbara Duprey | Peter Schofield | Kieran Peckett

Mark Morin | Christian Chenal | Laurent Balland-Poirier

Philippe Clément | Pierre-Yves Samyn | Shelagh Manton

Martin Saffron | Klaus-Jürgen Weghorn | Preston Manning Bernstein

Datum der Veröffentlichung und Softwareversion

Veröffentlicht am 04.08.2015. Basierend auf der LibreOffice Version 4.1.

Was Sie sehen, kann unterschiedlich sein

Illustrationen

LibreOffice läuft auf Windows, Linux, und Mac OS X. Jedes dieser Betriebssysteme hat mehrere Versionen und können durch den Anwender individuell eingerichtet werden (Schriftarten, Farben, Themen, Fenster Manager). Die Illustrationen in dieser Anleitung wurden von einer Vielfalt von Computern und Betriebssysteme übernommen. Deshalb werden einige Illustrationen nicht genau aussehen, wie was Sie auf Ihrem Computer-Bildschirm sehen.

Auch einige Dialoge können infolge der ausgewählten Einstellungen in LibreOffice unterschiedlich sein. Sie können entweder Dialoge von Ihrem Computersystem verwenden (standardmäßig) oder von LibreOffice bereitgestellte Dialoge. Um die Verwendung der LibreOffice Dialoge zu ändern:

1) Auf Linux und Windows Betriebssysteme, gehen Sie zu **Extras > Optionen > LibreOffice > Allgemein** auf der Hauptmenüleiste, um den Dialog für allgemeine Optionen zu öffnen.
2) Auf einem MAC-Betriebssystem, gehen Sie zu **LibreOffice > Voreinstellungen > Allgemein** auf der Hauptmenüleiste, um den Dialog für allgemeine Optionen zu öffnen.

   Wählen Sie LibreOffice Dialoge verwenden in Öffnen/Speichern Dialoge und, nur in Linux und Mac OS X Betriebssysteme, gehen Sie in Druck-Dialoge, um die LibreOffice Dialoge auf Ihrem Computer-Bildschirm anzuzeigen.
3) Klicken Sie auf OK, um Ihre Einstellungen zu speichern und den Dialog zu schließen.

Symbole

Die Symbole, die angewendet werden, um einige der vielen verfügbaren Werkzeuge in LibreOffice zu veranschaulichen, können von denen, die in dieser Anleitung dargestellt werden, abweichen. Die Symbole in dieser Anleitung wurden aus einer LibreOffice Installation übernommen, die für die Anzeige der Galaxy Symbol Reihe eingestellt war.

Wenn Sie es wünschen, können Sie Ihr LibreOffice Softwarepaket für die Anzeige der Galaxy Symbole wie folgt ändern:

1) Auf Linux und Windows Betriebssysteme, gehen zu **Extras > Optionen > LibreOffice > Ansicht** auf der Hauptmenüleiste, um den Dialog für die Ansicht Optionen zu öffnen.

   Auf einem MAC-Betriebssystem, gehen Sie zu **LibreOffice > Einstellungen > Ansicht** auf der Hauptmenüleiste, um den Dialog für die Ansicht Optionen zu öffnen.
2) In *Ansicht > Symbolgröße und Symbolstil* wählen Sie von den verfügbaren Optionen ***Galaxie*** in der Auswahlliste.
3) Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und den Dialog zu schließen.

Hinweis	Einige Linux-Betriebssysteme, zum Beispiel Ubuntu, gehören als Teil der LibreOffice Installation und beinhalten nicht die Symbole der Galaxy Reihe. Sie können den Galaxy-Symbol-Satz von den Software-Repositorien Ihres Linux-Betriebssystems herunterladen.

Die LibreOffice Verwendung auf einem MAC

Einige Tastatureingaben und Menüpunkte auf einem MAC unterscheiden sich von solchen in Windows und Linux. Die Tabelle unten gibt einige allgemeine Ersetzungen für die Instruktionen in diesem Abschnitt. Eine ausführlichere Liste finden Sie unter der Anwendungen Hilfe.

Windows oder Linux	***Mac gleichwertig***	***Effekt***
Extras > Optionen Menüauswahl	LibreOffice > Voreinstellungen	Der Zugriff auf Einrichtungsoptionen
Rechtsklick	Kontrolle+Klick oder Rechtsklick je nach Computer Einrichtung	Öffnet das Kontext-Menü
Ctrl (Strg)	⌘ *(Befehl)*	Benutzt mit anderen Tasten
F5	Umschalttaste+⌘+F5	Öffnet den Navigator
F11	⌘+*T*	Öffnet das Vorlagen- & Formatierungs-Fenster

Inhaltsverzeichnis

Kapitel 12 Arbeiten mit Makros....1
Copyright....2
Was Sie sehen, kann unterschiedlich sein....3
Illustrationen....3
Symbole....3
Die LibreOffice Verwendung auf einem MAC....4
Einführung....7
Die Anwendung des Makroaufzeichners....7
Schreiben Sie Ihre eigenen Funktionen....11
Die Verwendung eines Makros als eine Funktion....14
Das Übergeben von Argumenten an ein Makro....18
Argumente werden als Werte weitergegeben....19
Das Schreiben von Makros, die wie integrierte Funktionen agieren....19
Direkter Zugriff auf Zellen....19
Das Sortieren....21
Schlussfolgerung....22

Einführung

Ein Makro ist eine gespeicherte Befehlsfolge oder Tastatureingabe, die für den späteren Gebrauch gespeichert sind. Ein Beispiel von einem einfachen Makro ist eines, das “Eingeben“ Ihrer Adresse. Die LibreOffice-Makro Sprache ist sehr flexibel, und erlaubt die Automatisierung von beidem, einfache und komplexe Aufgaben. Makros sind besonders nützlich, um eine Aufgabe auf dieselbe Art und Weise immer wieder zu wiederholen.

Dieser Abschnitt diskutiert kurz die allgemeinen Probleme im Zusammenhang mit der Makroprogrammierung mit Calc.

Die Anwendung des Makroaufzeichners

Kapitel 13, in der “Erste Schritte Anleitung“, erste Schritte mit Makros, stellt eine Grundlage zum Verständnis der allgemeinen Makrofähigkeiten in LibreOffice mit dem Makroaufzeichner bereit.

Ein Beispiel ist hier ohne die Erläuterungen aus der *Ersten Schritte Anleitung* gezeigt.

Die folgenden Schritte, um ein Makro zu erstellen, das ***Inhalte einfügen mit Multiplizieren*** ausführt.

Tipp	Wenden Sie **Extras > Optionen > LibreOffice > Erweitert** an und wählen die Option **Ermöglicht eine Makroaufzeichnung (eingeschränkt)**, um den Makroaufzeichner zu aktivieren.

1) Öffnen Sie eine neue Tabellenkalkulation.
2) Geben Sie Zahlen in eine Tabelle ein.

	A	B	C
1	1	8	9
2	2	7	10
3	3	6	11

Abbildung 285: Geben Sie Zahlen ein

3) Wählen Sie Zelle A3, die die Zahl 3 enthält, und kopieren den Wert in die Zwischenablage.
4) Wählen Sie den Bereich A1:C3.
5) Wenden Sie **Extras > Makros > Makro aufzeichnen** an, um den Makroaufzeichner zu starten. Der ***Makro aufzeichnen*** Dialog wird mit einer **Aufzeichnung beenden** Schaltfläche angezeigt.

Abbildung 286: Aufzeichnung beenden Schaltfläche

6) Wenden Sie **Bearbeiten > Inhalte einfügen** an, um den “Inhalte einfügen“ Dialog zu öffnen (Abbildung 287).

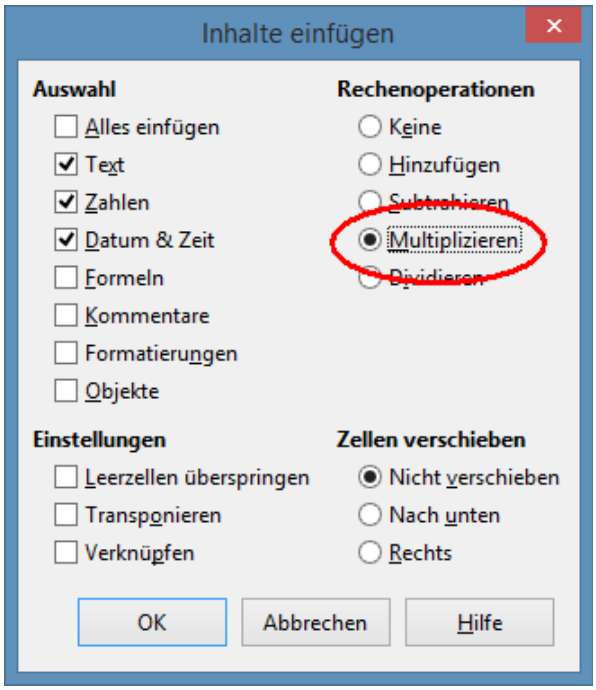


Abbildung 287: Inhalte einfügen Dialog

7) Setzen Sie die Rechenoperation auf **Multiplizieren** und klicken auf **OK**. Die Zellen sind jetzt mit 3 multipliziert (Abbildung 288Abbildung 288).

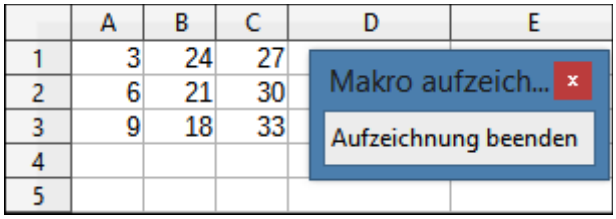
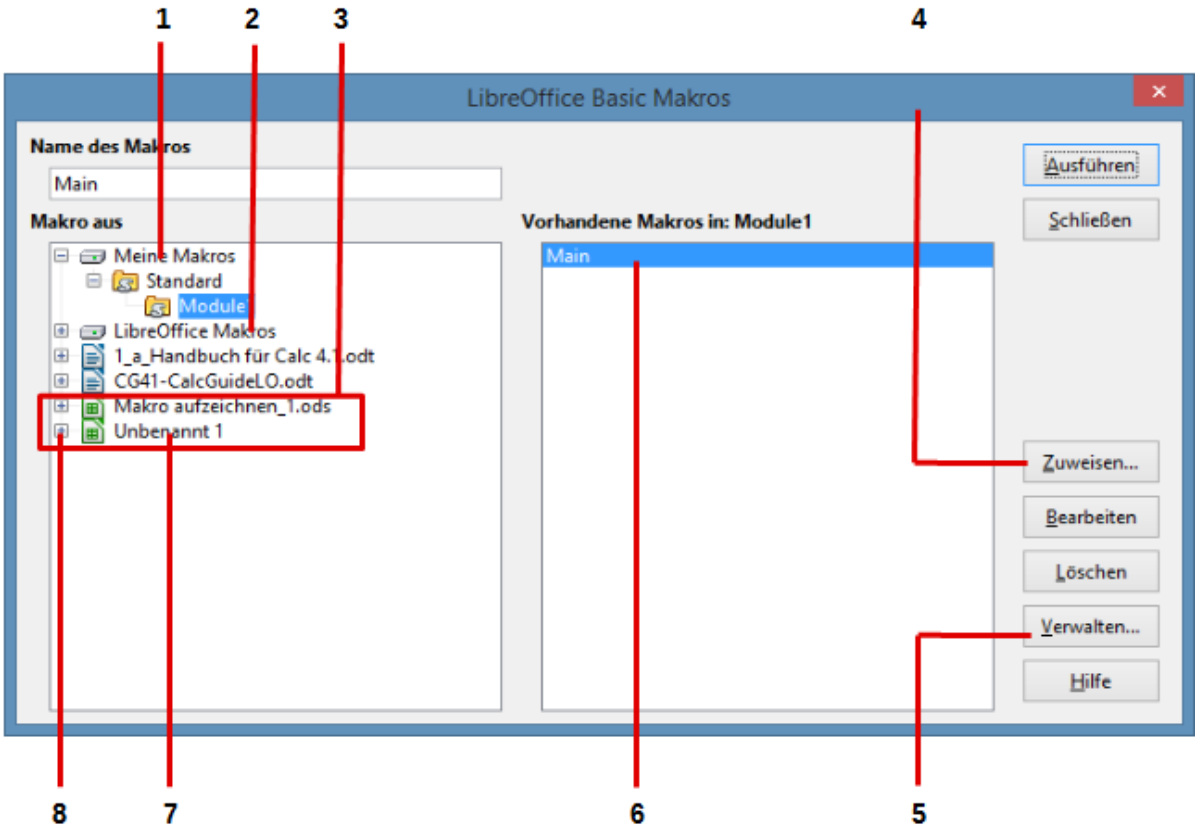


Abbildung 288: Zellen multipliziert mit 3

8) Klicken Sie auf *Aufzeichnung beenden*, um den Makroaufzeichner anzuhalten. Der LibreOffice Basic Makros Dialog (Abbildung 289) öffnet sich.

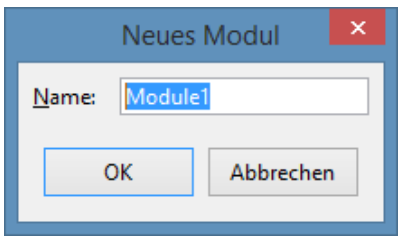


1	Meine Makros	5	Erstellen eines neuen Moduls in Bibliothek
2	LibreOffice Makros	6	Makros in ausgewählter Bibliothek
3	Dokumente öffnen	7	Aktuelles Dokument
4	Erstellen einer neuen Bibliothek	8	Erweitern/ausblenden Liste

Abbildung 289: Anteile des LibreOffice Basic Makros Dialogs

9) Wählen Sie das aktuelle Dokument. In diesem Beispiel, ist es *Unbetitelt 1*. Existierende Dokumente zeigen eine Bibliothek, genannt Standard. Diese Bibliothek wird nicht erstellt, bis, dass das Dokument gespeichert ist, oder die Bibliothek benötigt wird, so dass an dieser Stelle Ihr neues Dokument keine Bibliothek enthält. Sie können eine neue Bibliothek erstellen, um das Makro darin aufzunehmen, aber dies ist nicht erforderlich.

10) Klicken Sie auf **Neu**. Wenn keine Bibliotheken existieren, dann wird die Standardbibliothek automatisch erstellt und genutzt. In dem Dialog ***Neues Modul***, geben Sie einen Namen für das neue Modul ein oder lassen Sie den Namen als Module1.



Hinweis	Die Bibliotheken, Module und Makronamen müssen einigen strengen Regeln folgen. Im Anschluss müssen die Namen nach den Hauptregeln: • Mit einem Buchstaben beginnen • Keine Leerzeichen enthalten • Keine Sonderzeichen, einschließlich Akzente enthalten, abgesehen vom _ (Unterstrich)

11) Klicken Sie auf **OK**, um ein Neues Modul, genannt Module1, zu erstellen. Markieren Sie das neu erstellte Module1, schreiben Sie **`Multiplizieren_einfügen (PasteMultiply)`** in das Makro Namensfeld oben links, und klicken Sie auf **Speichern**. (siehe Abbildung 290.)

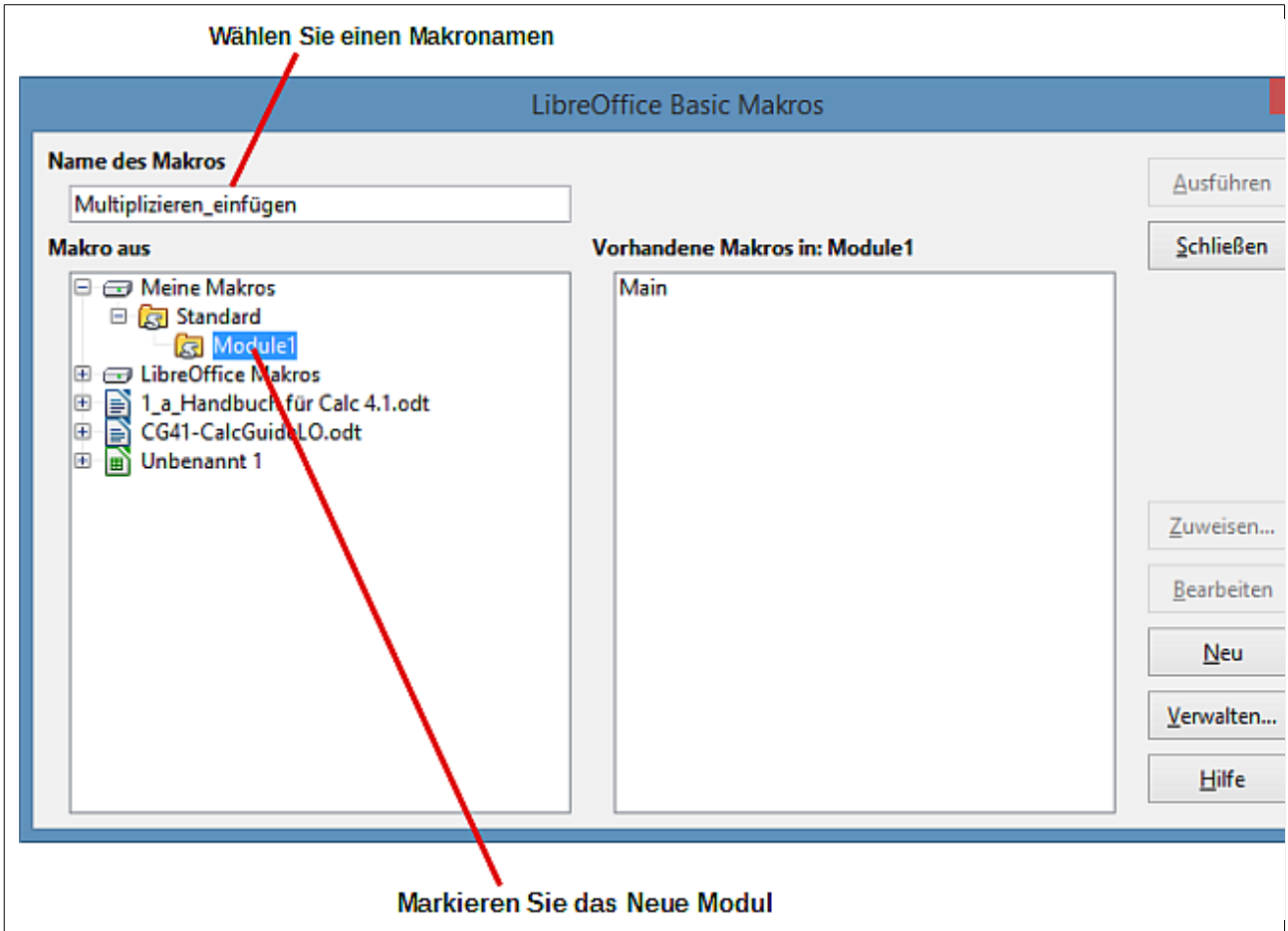


Abbildung 290: Markieren Sie das Modul und benennen Sie das Makro

Das erstellte Makro ist in Module1 der Standardbibliothek in dem *Unbenannt 1* Dokument gespeichert. Die Auflistung 1 zeigt die Inhalte des Makros.

Die Auflistung 1. Inhalte einfügen mit multiplizieren.

```
sub PasteMultiply
   rem ----------------------------------------------------------------
   rem define variables
   dim document   as object
   dim dispatcher as object
   rem ----------------------------------------------------------------
   rem get access to the document
   document   = ThisComponent.CurrentController.Frame
   dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")

   rem ----------------------------------------------------------------
   dim args1(5) as new com.sun.star.beans.PropertyValue
   args1(0).Name = "Flags"
   args1(0).Value = "A"
   args1(1).Name = "FormulaCommand"
   args1(1).Value = 3
   args1(2).Name = "SkipEmptyCells"
   args1(2).Value = false
   args1(3).Name = "Transpose"
   args1(3).Value = false
   args1(4).Name = "AsLink"
   args1(4).Value = false
   args1(5).Name = "MoveMode"
   args1(5).Value = 4

   dispatcher.executeDispatch(document, ".uno:InsertContents", "", 0, args1())
end sub
```

Weitere Details über die Aufzeichnung von Makros sind in *Kapitel 13, Einführung in Makros, in der Ersten Schritte Anleitung* bereitgestellt; wir empfehlen Ihnen dieses zu lesen, wenn Sie es nicht bereits getan haben. Weitere Details sind auch in den folgenden Abschnitten vorgesehen, aber nicht auf die Aufzeichnung der Makros bezogen.

Schreiben Sie Ihre eigenen Funktionen

Calc kann Makros als ***Calc Funktionen*** anrufen. Wenden Sie die folgenden Schritte an, um ein einfaches Makro zu erstellen:

1) Erstellen Sie ein neues Calc Dokument, genannt `CalcTestMacros.ods`.

2) Wenden Sie **Extras > Makros > Makros verwalten > LibreOffice Basic** an, um den LibreOffice Basic Makros Dialog zu öffnen. Das Fach ***Makro aus*** listet die verfügbaren Makrobibliothek Behälter auf, einschließlich alle aktuell geöffneten LibreOffice Dokumente. ***Meine Makros*** enthalten Makros die Sie schreiben oder zu LibreOffice hinzufügen. *LibreOffice Makros* enthalten Makros, die in LibreOffice enthalten sind und sollten nicht geändert werden.

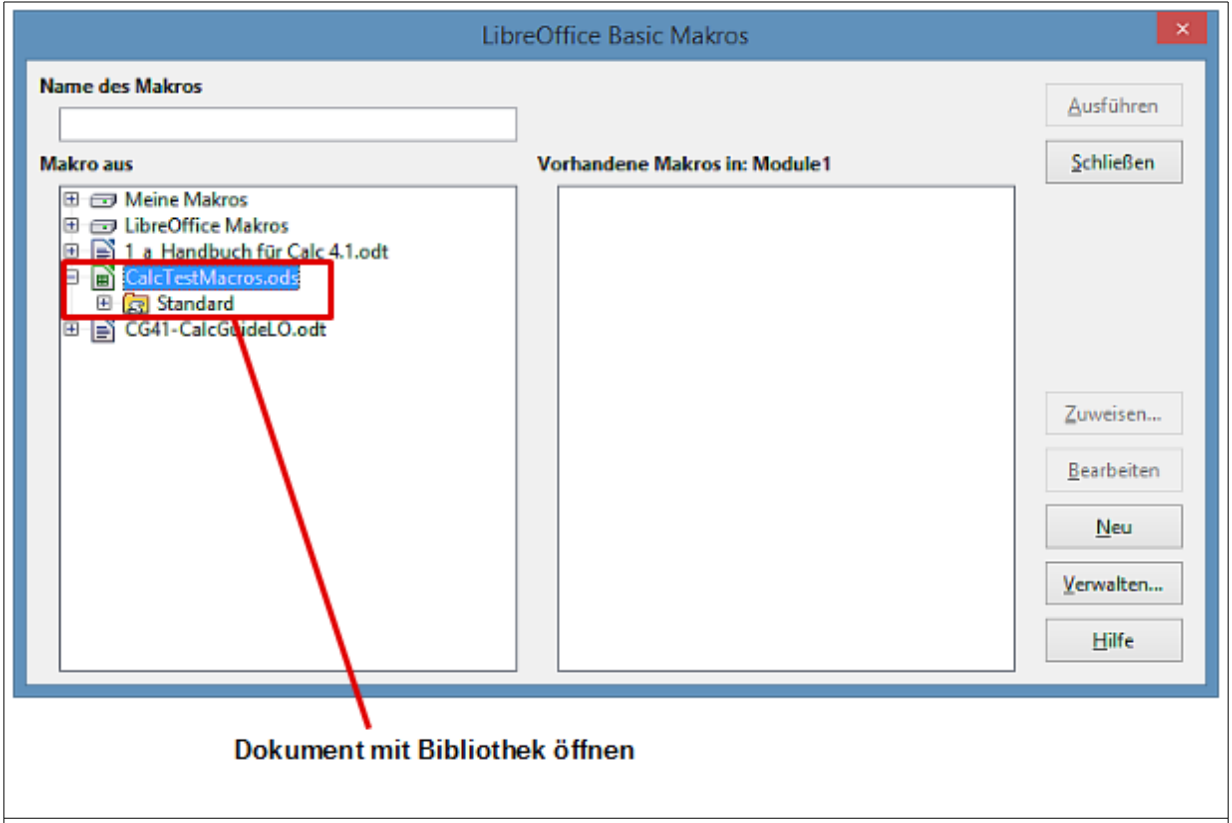


Abbildung 291: LibreOffice Basic Makros Dialog

3) Klicken Sie auf **Verwalten**, um den LibreOffice Basic-Makro Verwalten Dialog zu öffnen (Abbildung 292). Auf der Bibliotheken Registerkarte, wählen Sie das Dokument mit dem enthaltenen Makro.

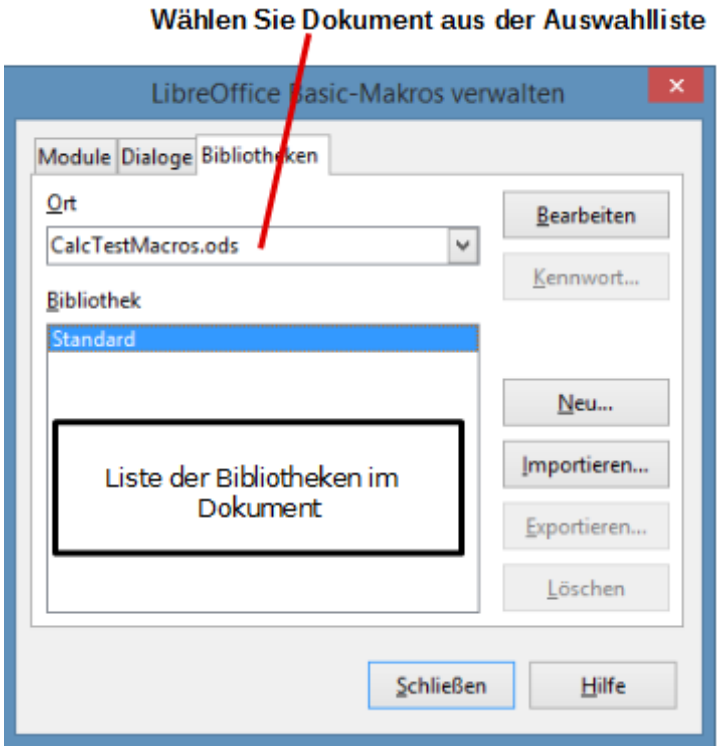


Abbildung 292: LibreOffice Basic-Makro verwalten

4) Klicken Sie auf **Neu**, um den neuen Bibliotheksdialog zu öffnen.

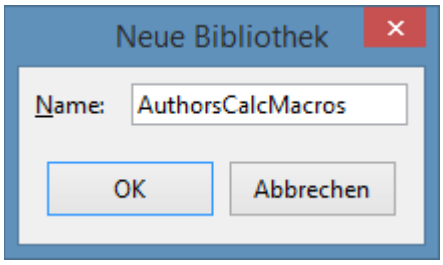


Abbildung 293: Neue Bibliothek Dialog

5) Geben Sie einen beschreibenden Bibliotheksnamen ein (wie zum Beispiel AuthorsCalcMacros) und klicken Sie auf **OK**, um die Bibliothek zu erstellen. Der neue Bibliotheksname wird in der Bibliotheksliste angezeigt, aber der Dialog kann nur einen Teil des Namens zeigen.

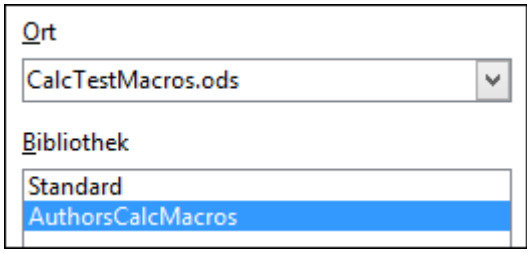


Abbildung 294: Die Bibliothek wird in dem Verwalten angezeigt

6) Wählen Sie ***AuthorsCalcMacros*** und klicken Sie auf **Bearbeiten**, um die Bibliothek zu bearbeiten. Calc erstellt automatisch ein Modul, genannt Module1 und ein Makro, genannt Main.

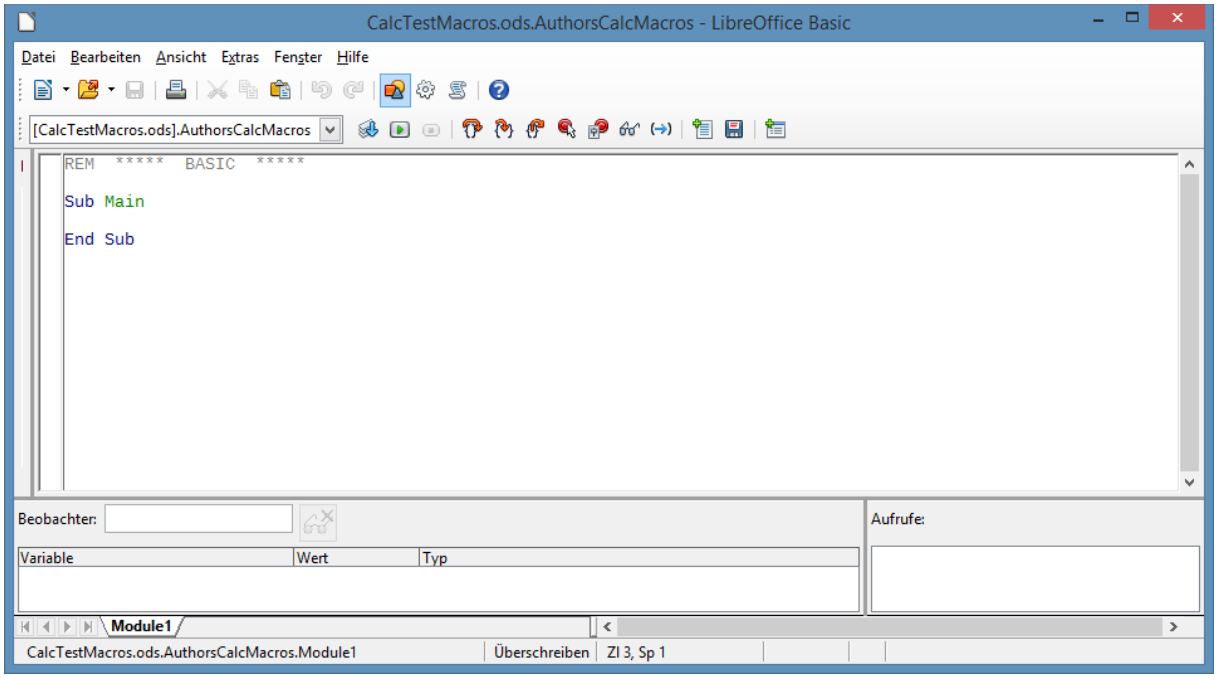


Abbildung 295: Basic integrierte Entwicklungsumgebung (IDE)

7) Modifizieren Sie den Code, sodass er genauso ist, wie der in Auflistung 2 angezeigte.

   Die wichtigste Hinzufügung ist die Erstellung von der `NumberFive` Funktion, welche die Nummer fünf zurückgibt. Die `Option Explicit` Anweisung zwingt alle Variablen deklariert zu sein, bevor sie verwendet werden. Wenn die `Option Explicit` fehlt, werden die Variablen zuerst automatisch definiert und als Typenvariante verwendet.

8) Speichern Sie das modifizierte Module1.

Die Auflistung 2. Funktion, die fünf zurückgibt.

```
REM  *****  BASIC  *****
Option Explicit

Sub Main

End Sub

Function NumberFive()
  NumberFive = 5
End Function
```

Die Verwendung eines Makros als eine Funktion

Die Verwendung des neu erstellten Calc Dokument `CalcTestMacros.ods`, geben Sie die Formel **`=NumberFive()`** ein (siehe Abbildung 296). Calc findet das Makro und ruft es auf.

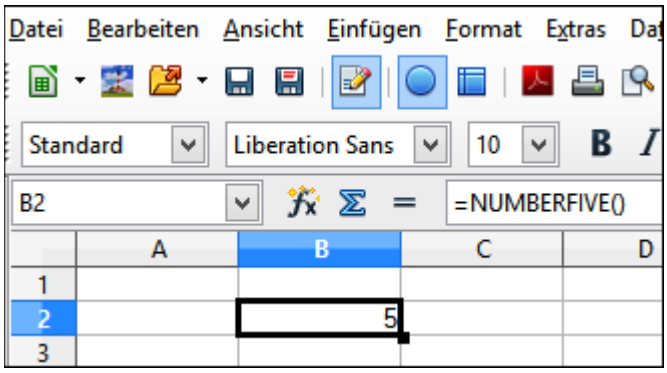


Abbildung 296: Anwendung des NumberFive() Makros als eine Calc Funktion

Tipp	Funktionsnamen werden nicht die Groß-/Kleinschreibung beachten. In Abbildung 296 können Sie =NumberFive() eingeben und Calc zeigt deutlich =NUMBERFIVE().

Speichern Sie das Calc Dokument, schließen Sie es, und öffnen es wieder. In Abhängigkeit von Ihren Einstellungen in **Extras > Optionen > LibreOffice > Sicherheit > Makro Sicherheit**, wird Calc die Warnung anzeigen, gezeigt in Abbildung 297 oder die Eine in Abbildung 298 gezeigt.

Sie müssen **Makros aktivieren** anklicken, oder Calc wird keine Makros bewilligen, um es innerhalb des Dokuments zu starten. Wenn Sie nicht erwarten, das ein Dokument ein Makro enthält, ist es sicherer **Makros deaktivieren** anzuklicken, falls das Makro ein Virus sein sollte.

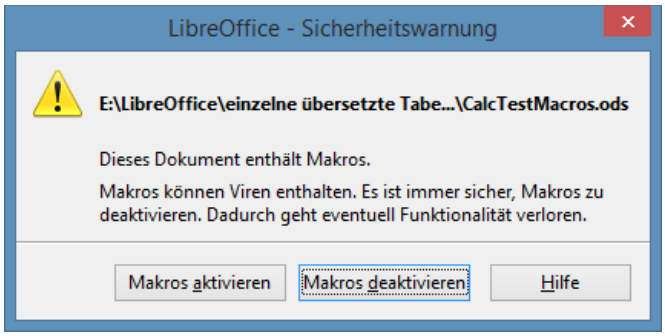


Abbildung 297: LibreOffice warnt Sie, wenn das Dokument Makros enthält

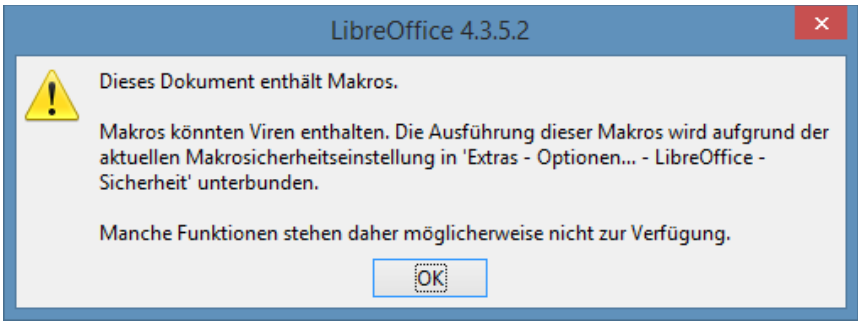


Abbildung 298: Warnung, wenn Makros deaktiviert sind

Wenn Sie ***Makros deaktivieren*** wählen, dann, wenn das Dokument lädt, kann Calc die Funktion nicht mehr finden.

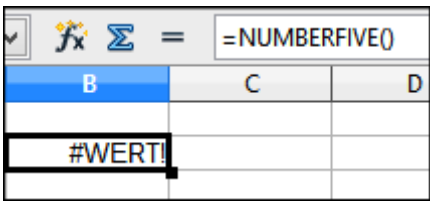


Abbildung 299: Die Funktion ist verschwunden

Wenn ein Dokument erstellt und gespeichert wird, enthält es automatisch eine Bibliothek, genannt Standard. Die Standardbibliothek wird automatisch geladen, wenn das Dokument geöffnet wird. Keine andere Bibliothek wird automatisch geladen.

Calc enthält keine Funktion, benannt als `NumberFive()`, deshalb prüft es alle geöffneten und sichtbaren Makrobibliotheken für die Funktion. Bibliotheken in *LibreOffice Makros, Meine Makros*, und das Calc Dokument sind für eine entsprechend benannte Funktion aktiviert (siehe Abbildung 291). Die NumberFive() Funktion ist in der AuthorsCalcMacros Bibliothek gespeichert, was nicht automatisch geladen ist, wenn das Dokument geöffnet wird.

Verwenden Sie **Extras > Makros > Makros verwalten > LibreOffice Basic**, um den LibreOffice Basic Makrodialog zu öffnen (siehe Abbildung 300). Erweitern Sie **[+] CalcTestMacros** und Sie finden *AuthorsCalcMacros*. Das Symbol für eine geladene Bibliothek ist in einer anderen Farbe, als das Symbol für eine Bibliothek, die nicht geladen wird.

Klicken Sie auf das Erweiterungs-Symbol (gewöhnlich ein plus bzw. ein Dreieck) neben AuthorsCalcMacros, um die Bibliothek zu laden. Das Symbol ändert die Farbe, um anzuzeigen, dass die Bibliothek jetzt geladen ist. Klicken Sie auf **Schließen**, um den Dialog zu schließen.

Bedauerlicherweise sind die Zellen die **`=NumberFive()`** enthalten, fehlerhaft. Calc berechnet keine fehlerhaften Zellen neu, es sei denn, Sie bearbeiten sie oder ändern diese irgendwie. Die normale Lösung ist, verwendete Makros als Funktionen in der Standardbibliothek zu speichern. Wenn das Makro groß ist oder, wenn es viele Makros gibt, ist eine abgekürzte Form, mit dem gewünschten Namen, in der Standardbibliothek gespeichert. Die abgekürzte Makro Form lädt die Bibliothek mit der Implementierung und ruft dann die Durchführung auf.

1) Verwenden Sie **Extras > Makros > Makros verwalten > LibreOffice Basic**, um den LibreOffice Basic Makrodialog zu öffnen. Markieren Sie das NumberFive-Makro und klicken Sie auf Bearbeiten, um das Makro zur Bearbeitung zu öffnen.

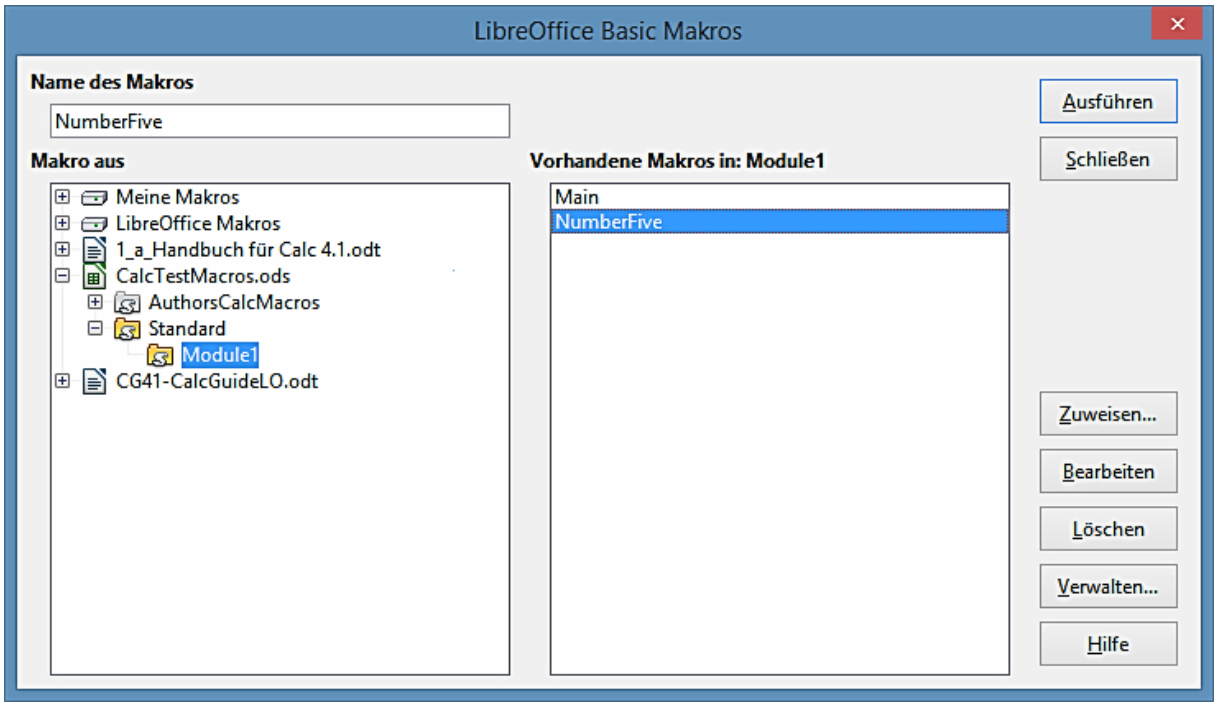


Abbildung 300: Markieren Sie ein Makro und klicken Sie auf Bearbeiten

2) Ändern Sie den Namen von *NumberFive* in *NumberFive_Implementation* um (Auflistung 3).

   Die Auflistung 3. Ändern Sie den Namen von NumberFive in NumberFive_Implementation

```
Function NumberFive_Implementation()
  NumberFive_Implementation() = 5
End Function
```

3) In dem Basic IDE (siehe Abbildung 295), bewegen Sie den Maus-Cursor über die Symbolleisten-Schaltflächen, um die Quickinfos anzuzeigen. Klicken Sie auf die **Makro auswähle**n Schaltfläche, um den LibreOffice Basic Makrodialog zu öffnen (siehe Abbildung 300).

4) Wählen Sie die Standardbibliothek in dem CalcTestMacros Dokument und klicken Sie auf **Neu**, um ein Neues Modul zu erstellen. Geben Sie einen sinnvollen Namen ein, wie zum Beispiel CalcFunctions und klicken Sie auf **OK**. LibreOffice erstellt automatisch ein Makro, genannt Main und öffnet das Modul zur Bearbeitung.

5) Erstellen Sie in der Standardbibliothek ein Makro, dass die Implementierungs-Funktion aufruft (siehe Auflistung 4). Das neue Makro lädt die AuthorsCalcMacros Bibliothek, wenn sie nicht bereits geladen ist, und dann ruft es die Implementierungs-Funktion auf.

6) Speichern, Schließen, und dann das Calc Dokument wieder öffnen. Dieses Mal arbeitet die NumberFive() Funktion.

Die Auflistung 4. Ändern Sie den Namen von NumberFive in NumberFive_Implementation um.

```
Function NumberFive()
  If NOT BasicLibraries.isLibraryLoaded("AuthorsCalcMacros") Then
    BasicLibraries.LoadLibrary("AuthorsCalcMacros")
  End If
  NumberFive = NumberFive_Implementation()
```

```
End Function
```

Das Übergeben von Argumenten an ein Makro

Um eine Funktion zu veranschaulichen, die Argumente akzeptiert, werden wir ein Makro schreiben, das die Summe seiner Argumente berechnet, die positiv sind – es wird Argumente ignorieren, die kleiner als Null sind (siehe Auflistung 5).

Die Auflistung 5. PositiveSum berechnet die Summe der positiven Argumente.

```
Function PositiveSum(Optional x)
  Dim TheSum As Double
  Dim iRow As Integer
  Dim iCol As Integer

  TheSum = 0.0
  If NOT IsMissing(x) Then
    If NOT IsArray(x) Then
      If x > 0 Then TheSum = x
    Else
      For iRow = LBound(x, 1) To UBound(x, 1)
        For iCol = LBound(x, 2) To UBound(x, 2)
          If x(iRow, iCol) > 0 Then TheSum = TheSum + x(iRow, iCol)
        Next
      Next
    End If
  End If
  PositiveSum = TheSum
End Function
```

Das Makro in Auflistung 5 demonstriert einige wichtige Techniken:

1) Das Argument `x` ist optional. Wenn ein Argument nicht optional ist und die Funktion wird ohne es aufgerufen, druckt LibreOffice jedes Mal eine Warnmeldung, wenn das Makro aufgerufen wird. Wenn Calc die Funktion oft aufruft, dann wird der Fehler auch oft angezeigt.

2) `IsMissing` prüft, dass ein Argument weitergegeben wurde, bevor das Argument gebraucht wird.

3) `IsArray` prüft, um zu sehen, ob das Argument ein einzelner Wert oder ein Bereich ist. Zum Beispiel, `=PositiveSum(7)` oder `=PositiveSum(A4)`. Im ersten Fall, wird die Zahl 7 als ein Argument weitergegeben, und im zweiten Fall, wird der Wert von Zelle A4 zu der Funktion weitergeleitet.

4) Wenn ein Bereich zu der Funktion weitergegeben wird, ist er als ein zweidimensionaler Bereich aus Werten weitergegeben; z. B. **`=PositiveSum(A2:B5)`**. **LBound** und **UBound** werden verwendet, um die Indexgrenzen zu bestimmen, die benutzt werden. Obwohl die ***untere Grenze*** **1** ist, kommt es sicherer in Betracht **LBound** zu benutzen, falls es zukünftig geändert wird.

Tipp	Das Makro in Auflistung 5 ist sorgfältig und prüft, um zu sehen, ob das Argument ein Feld oder ein einzelnes Argument ist. Das Makro überprüft nicht, dass jeder

Wert numerisch ist. Sie können so sorgfältig sein wie Sie mögen. Je mehr Dinge Sie prüfen, umso robuster ist das Makro, und desto langsamer läuft es.

Die Weitergabe von einem Argument ist so leicht wie die Weitergabe von zwei: fügen Sie ein weiteres Argument zu der Funktionsdefinition hinzu (siehe Auflistung 6). Wenn Sie eine Funktion mit zwei Argumenten aufrufen, werden die Argumente mit einem Semikolon getrennt; z. B. **`=TestMax(3; -4)`**.

Die Auflistung 6. TestMax akzeptiert zwei Argumente und gibt das größere von den zwei zurück.

```
Function TestMax(x, y)
  If x >= y Then
    TestMax = x
  Else
    TestMax = y
  End If
End Function
```

Argumente werden als Werte weitergegeben

Weitergegebene Argumente in einem Calc Makro, sind immer Werte. Es ist nicht möglich zu erfahren welche Zellen, sofern vorhanden, verwendet werden. Zum Beispiel **`=PositiveSum(A3)`** gibt den Wert von Zellen A3 weiter, und **`PositiveSum`** hat keine Möglichkeit zu wissen, dass die Zelle A3 benutzt wurde. Wenn Sie wissen müssen, welche Zellen und nicht die Werte in den Zellen referenziert werden, übergeben Sie den Bereich als Zeichenkette, analysieren Sie die Zeichenfolge, und bekommen die Werte in den bezugnehmenden Zellen.

Das Schreiben von Makros, die wie integrierte Funktionen agieren

Obwohl Calc Makros findet und wie normale Funktionen aufruft, verhalten sie sich nicht wirklich als integrierte Funktionen. Zum Beispiel, Makros erscheinen nicht in der Funktionslisten. Es ist möglich Funktionen zu schreiben, die sich als reguläre Funktionen, durch Schreiben eines Add-In, verhalten. Jedoch ist dies ein fortgeschrittenes Thema, das hier nicht behandelt wird.

Direkter Zugriff auf Zellen

Sie können auf die LibreOffice internen Objekte direkt zugreifen, um ein Calc Dokument zu manipulieren. Zum Beispiel, das Makro in Auflistung 7 fügt die Werte der Zelle A2 in jede Tabelle in dem aktuellen Dokument hinzu. `ThisComponent` ist von Star Basic eingestellt, wenn das Makro beginnt, um auf das aktuelle Dokument bezugnehmen. Ein Calc Dokument enthält Tabellen: `ThisComponent.getSheets()`. Verwenden Sie `getCellByPosition(col, row)` (Spalte, Zeile), um eine Zelle an eine bestimmte Zeile und Spalte zurückzugeben.

Die Auflistung 7. Fügt die Zelle A2 in jede Tabelle hinzu.

```
Function SumCellsAllSheets()
  Dim TheSum As Double
  Dim i As integer
  Dim oSheets
  Dim oSheet
  Dim oCell

  oSheets = ThisComponent.getSheets()
  For i = 0 To oSheets.getCount() - 1
    oSheet = oSheets.getByIndex(i)
    oCell = oSheet.getCellByPosition(0, 1) ' GetCell A2
    TheSum = TheSum + oCell.getValue()
  Next
  SumCellsAllSheets = TheSum
End Function
```

Tipp

Ein Zellobjekt unterstützt die Methoden **`getValue()`**, **`getString()`**, und **`getFormula()`**, um den Zahlenwert, die Zeichenfolge, oder die Formel zu erhalten, die in einer Zelle angewendet werden. Verwenden Sie den entsprechenden Funktions-Satz, um die entsprechenden Werte festzulegen.

Verwenden Sie `oSheet.getCellRangeByName("A2")`, um einen Zellenbereich namentlich zurückzugeben. Wenn eine einzelne Zelle bezugnehmend ist, dann wird ein Zellobjekt zurückgegeben. Wenn ein Zellenbereich gegeben ist, dann wird ein ganzer Zellenbereich zurückgegeben (siehe Auflistung 8). Beachten Sie, dass ein Zellenbereich, Daten als einen Bereich von Bereichen zurück gibt, was umständlicher ist, als es als einen Bereich mit zwei Dimensionen zu behandeln, wie es in der Auflistung 5 geschieht.

Die Auflistung 8. Fügen Sie Zellen A2:C5 in jede Tabelle hinzu

```
Function SumCellsAllSheets()
  Dim TheSum As Double
  Dim iRow As Integer, iCol As Integer, i As Integer
  Dim oSheets, oSheet, oCells
  Dim oRow(), oRows()

  oSheets = ThisComponent.getSheets()
  For i = 0 To oSheets.getCount() - 1
    oSheet = oSheets.getByIndex(i)
    oCells = oSheet.getCellRangeByName("A2:C5")
    REM getDataArray() returns the data as variant so strings
    REM are also returned.
    REM getData() returns data data as type Double, so only
    REM numbers are returned.
    oRows() = oCells.getData()
    For iRow = LBound(oRows()) To UBound(oRows())
      oRow() = oRows(iRow)
      For iCol = LBound(oRow()) To UBound(oRow())
        TheSum = TheSum + oRow(iCol)
      Next
    Next
  Next
  SumCellsAllSheets = TheSum
End Function
```

Tipp	Wenn ein Makro als eine Calc Funktion aufgerufen wird, kann das Makro keinen Wert in der Tabelle ändern, aus welcher das Makro aufgerufen wurde, außer den Wert der Zelle, die die Funktion enthält.

Das Sortieren

Betrachten Sie die Sortierung der Daten in Abbildung 301. Zuerst, sortieren Sie auf Spalte B absteigend und dann Spalte A aufsteigend.

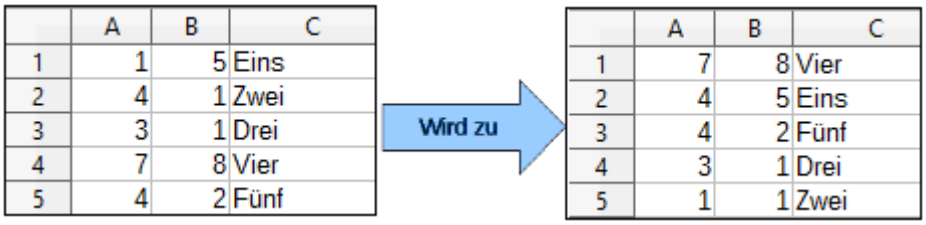


Abbildung 301: Sortieren der Spalte B absteigend und Spalte A aufsteigend

Das Beispiel in Auflistung 9, veranschaulicht jedoch, wie man auf zwei Spalten sortiert.

Die Auflistung 9. Sortieren der Zellen A1:C5 auf Tabelle 1.

```
Sub SortRange
  Dim oSheet        ' Calc sheet containing data to sort.
  Dim oCellRange    ' Data range to sort.

  REM An array of sort fields determines the columns that are
  REM sorted. This is an array with two elements, 0 and 1.
  REM To sort on only one column, use:
  REM Dim oSortFields(0) As New com.sun.star.util.SortField
  Dim oSortFields(1) As New com.sun.star.util.SortField

  REM The sort descriptor is an array of properties.
  REM The primary property contains the sort fields.
  Dim oSortDesc(0) As New com.sun.star.beans.PropertyValue

  REM Get the sheet named "Sheet1"
  oSheet = ThisComponent.Sheets.getByName("Sheet1")

  REM Get the cell range to sort
  oCellRange = oSheet.getCellRangeByName("A1:C5")

  REM Select the range to sort.
  REM The only purpose would be to emphasize the sorted data.
  'ThisComponent.getCurrentController.select(oCellRange)

  REM The columns are numbered starting with 0, so
  REM column A is 0, column B is 1, etc.
  REM Sort column B (column 1) descending.
  oSortFields(0).Field = 1
  oSortFields(0).SortAscending = FALSE
```

```
    REM If column B has two cells with the same value,
    REM then use column A ascending to decide the order.
    oSortFields(1).Field = 0
    oSortFields(1).SortAscending = True

    REM Setup the sort descriptor.
    oSortDesc(0).Name = "SortFields"
    oSortDesc(0).Value = oSortFields()

    REM Sort the range.
    oCellRange.Sort(oSortDesc())
End Sub
```

Schlussfolgerung

Dieser Abschnitt stellt eine Kurzübersicht darüber bereit, wie man Bibliotheken und Module erstellt, den Makroaufzeichner anwendet, Makros als Calc Funktionen übernimmt, und um Ihre eigenen Makros, ohne den Makroaufzeichner, zu schreiben. Jedes Thema verdient mindestens ein Kapitel, und das Schreiben Ihrer eigenen Makros für Calc, könnte leicht ein vollständiges Buch ausfüllen. Mit anderen Worten, dies ist nur der Beginn davon, was Sie lernen können!