



# **DIE rS1.METHODE**

Kalkulationsmodelle planen, entwickeln,  
strukturieren und dynamisieren mit

## **MICROSOFT EXCEL 2007 – 2010**

(Die Beschreibungen gelten sinngemäß auch für die Versionen ab Excel 97)

### **Teil 1: Anwendung der Methode auf Arbeitsmappen, Arbeitsblätter und Tabellen**

Bitte beachten Sie, dass für die übergebenen Materialien uneingeschränkte urheberrechtliche Vorschriften gelten. Inhalte, Texte, Bilder und Gestaltung sind geschützt und dürfen ohne ausdrückliche schriftliche Zustimmung des Autors weder ganz noch teilweise in irgendeiner Form vervielfältigt, gespeichert oder anderweitig verwendet werden.



# Die rS1.Methode

## Teil 1

### Anwendung der Methode auf Arbeitsblätter und -mappen

Dieser Text ist Beigabe und Ergänzungsinformation zu einem Buch. Die meisten der Beispieldateien, die auf der zum Buch gehörenden CD gespeichert sind, habe ich nach den Regeln der nachstehend beschriebenen rS1.Methode angefertigt. Die Inhalte des Buchs beschäftigen sich schwergewichtig mit der Entwicklung von dynamischen Lösungen, die keine Programmierung benötigen. Aus diesem Grund sind jene Teile der rS1.Methode, die Vorgaben zur Programmierung von Excel beschreiben, hier nicht enthalten.

Ein Teil der Abbildungen und die zugehörigen Texte beziehen sich auf die Anwendung von Excel 2010. Die Unterschiede zu Excel 2007 sind geringfügig und betreffen lediglich einige Handhabungen oder Zugangswege. Alle Anleitungen und Regeln gelten sinngemäß auch für die älteren Programmfassungen ab der Version Excel 97.

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeine Informationen</b>	<b>2</b>
1.1	Zur Entstehung der rS1.Methode	2
1.2	Generelle Vorteile	3
<b>2</b>	<b>Grundregeln der Methode</b>	<b>4</b>
<b>3</b>	<b>Regeln für Mappen und Blätter</b>	<b>6</b>
3.1	Arbeitsmappen und deren Gestaltung	6
3.2	Bereichsnamen und deren Verwendung	11
3.3	Objektnamen und deren Verwendung	15
3.4	Besonderheiten, Empfehlungen und Hinweise	17
3.5	Mustervorlagen	27

# 1 ALLGEMEINE INFORMATIONEN

Die rS1.Methode hat den Charakter einer Richtlinie und dient der Strukturierung und Vereinheitlichung von Entwicklungen unter Microsoft Excel. Hört sich ziemlich langweilig an. Ist es aber nicht, wie Sie sehen werden und hoffentlich auch am zukünftigen Erfolg Ihrer Arbeit erfahren. Denn was die Methode will und kann, egal ob sie auf einfache Tabellenmodelle oder auf umfangreiche, komplexe und dynamische Kalkulationslösungen Anwendung findet, ist schwer zu ignorieren: Sie macht Ihre Arbeit leichter, Ihre Vorgehensweise übersichtlicher, Ihre Ergebnisse attraktiver.

## 1.1 Zur Entstehung der rS1.Methode

Ich arbeite seit 1989 intensiv mit Excel und habe mich von Anfang an immer wieder über dieselben Probleme geärgert. Nicht so sehr über Excel-Probleme – davon gab es nicht so viele –, sondern über selbst gemachte: Die eigenen Lösungen, schnell geschaffen und immer wieder mal funktional erweitert, entstanden, stets nur am Ergebnis orientiert, Stück für Stück, nach und nach. Negativ ausgedrückt: „im Wildwuchs“. Sie funktionierten meistens gut und manche immer besser, hatten aber mangels Struktur diverse Mängel und Nachteile:

- Die Ergebnisse waren unübersichtlich und für andere kaum nachvollziehbar.
- Benutzte Methoden und Wege waren nach einiger Zeit auch für den Entwickler selbst nur noch schwer interpretierbar.
- Sich ähnelnde Probleme wurden unsystematisch und ohne Auswahl einer „besten Methode“ verschiedenartig gelöst.
- Gefundene Lösungen waren nur schwer oder gar nicht auf andere, gleichartige Aufgabenstellungen übertragbar.

Es ergab sich als glückliche Fügung, dass zu jener Zeit US-amerikanische Makroprogrammierer eine Methode definierten und veröffentlichten, mit der sie eine sehr schöne Ordnung in die so genannte Excel-Makrovorlage brachten. (Ganz kurz dazu: Die Makrovorlage, es gibt sie auch heute unter Excel 2007 noch, ist ein spezielles tabellarisches Arbeitsblatt, in dem sich auch damals schon unter Excel sehr ansehnliche und flotte Programme schreiben ließen.) Das Verfahren ist seit 1993 durch die Einführung von VBA in den Hintergrund geraten und wurde schließlich weitestgehend vergessen. Aber die so erstellten alten Programme funktionieren immer noch! Ein Lob an die Excel-Entwickler: Sie haben sich trotz zahlreicher – und teilweise eher überflüssiger – Versionswechsel beharrlich um Abwärtskompatibilität bemüht. Grundidee der amerikanischen Kollegen war eine konsequente und funktional orientierte Namensvergabe für Makros, eine deutlich gegliederte Schreibweise der Programmsequenzen sowie deren modulare Aufteilung in verschiedene Bereiche der Vorlage. Unter Aufnahme dieser Gedanken habe ich ein ähnliches Strukturungsverfahren für VBA-Projekte entwickelt und, sehr viel wichtiger noch, mein System dann auch auf solche Excel-Modelle und -Lösungen übertragen, die zwar völlig ohne Programmcode auskommen, aber dennoch ähnlich hohe, bisweilen gar höhere Ansprüche an ein leicht nachvollziehbares Regelwerk stellen. Das Ergebnis war wirklich verblüffend günstig und ist es bis heute geblieben:

- Sie suchen etwas und finden es auch.
- Besser noch: Sie wissen schon vorher, wo Sie und was Sie *nicht* suchen müssen.
- Sie verstehen auch noch Monate oder Jahre später selbst hoch komplizierte und seit langem nicht mehr benutzte Formeln ohne große Mühe.
- Kenner und Anwender der Methode verstehen ohne umfangreiche Erläuterungen Inhalte und Lösungswege auch fremder Modelle.
- Ihre Lösung ist leicht und elegant auf ähnliche Aufgabenstellungen anzuwenden.

Einige Teile des Systems wurden erstmals bereits 1994 als „LS1-Methode“ publiziert. Gemessen an den sehr kurzen Innovationszyklen der Branche ist die rS1.Methode also, was keinesfalls gegen sie spricht, schon *uralt*. In all den Jahren kam sie, immer wieder mit kleinen Varianten und Ergänzungen versehen, in vielen eigenen Projekten, Auftragsarbeiten und komplexen Branchenlösungen zum Einsatz. Mein Interesse an ihrer Verbreitung war aber, vorwiegend aus Gründen des Schutzes von Entwicklungs-Know-how, nur sehr gering. Die vollständige öffentliche Darlegung – nun als „rS1.Methode“ – erfolgte dann erstmals im Jahr 2003.

## 1.2 Generelle Vorteile

Als Anwender der rS1.Methode können Sie vielen Anforderungen genügen, wie sie ansonsten in stringenten und konsistenten Profi-Programmierrichtlinien gelten. Mehr noch: Sie dehnen solche hohen Qualitätsansprüche auf Modelle aus, die mit Programmierung gar nichts zu tun haben. Damit schaffen Sie für jede Excel-Lösung, gleich welcher Art und Aufgabe, Ordnung und Nachvollziehbarkeit. Solche Vorteile begünstigen Sie natürlich nicht nur in der konstruktiven Phase, sondern bereits vorher und ganz besonders nachher. Die bloße Existenz definierter Rahmenbedingungen ermöglicht oder erzwingt Überlegungen, die ansonsten eher nicht, noch nicht, vielleicht auch nie erfolgen würden. Ein Beispiel: In der rS1-Arbeitsmappe gibt es ein Arbeitsblatt mit dem Namen *Parameter 1*. Spätestens wenn ich meine rS1-Excel-Vorlage (früher *Mustervorlage*) geöffnet habe, um mit meiner Arbeit zu beginnen, reflektiere ich folglich, was Parameter sind, ob mein Modell schon Parameter hat oder benötigt, welche beim Auftraggeber vorhandenen Parameter für die Lösung nutzbar wären und Ähnliches. Nicht selten – Sie werden so etwas kennen – tauchen dann schon vor Beginn der Arbeit Ideen auf, die es ohne das Vorhandensein des Parameterblatts gar nicht erst gegeben hätte. Und wenn das erst verinnerlicht ist, gehört ein Planen von Parametern bald schon wie selbstverständlich zur Vorbereitung.

Der überzeugendste Vorteil der rS1.Methode für die Zeit *nach* der Entwicklungsphase eines Modells ist die enorme Erleichterung von Pflege und Support. Wenn ich als „Architekt“ meiner eigenen Lösung jederzeit ihren Gesamtaufbau, ihre Konstrukte und ihre Details nachvollziehen und verstehe, dann kann ich auch sämtliche Fragen dazu beantworten. Vor allem auch solche, die ich mir selber stelle. Es mag sein, dass mir ein vor einiger Zeit gefundenes Ergebnis nicht mehr so gut oder gar nicht mehr gefällt. Oder auch, dass ich mich insgeheim wundere, warum es dem Anwender *noch* gefällt. Wie auch immer, wenn ich nach Monaten oder gar Jahren leicht und schnell nachvollziehe, was ich da und warum ich es gemacht habe, sei es der „Putz“ oder seien es die „Innereien“, kann ich es auch leicht und schnell ändern oder verbessern. Mehr noch: Weil mich die Anwendung der rS1.Methode zu bestimmten Vorgehensweisen zwar nicht zwingt, wohl aber drängt, entstehen ganz selbstverständlich Lösungen, deren konstruktive Elemente leicht austauschbar sind. Das heißt, und darüber ist die Freude oft beträchtlich, ich kann aus einem Kalkulationsmodell große Teile kopieren, sie in einen neuen Projekt einbauen, dort anpassen (dazu genügt bisweilen *Suchen und Ersetzen*) – und fertig.

So weit zum „Davor“ und zum „Danach“. Auf eine ausführliche Beschreibung der Vorteile in der Konstruktionsphase selbst kann an dieser Stelle verzichtet werden. Denn wenn Sie nach den hier beschriebenen Vorgaben arbeiten, werden Sie kennen- und hoffentlich schätzen lernen, welcher unmittelbare und mittelbare Nutzen sich für Ihre alltägliche Arbeit mit Excel aus der Anwendung der rS1.Methode ziehen lässt. Was zunächst etwas sperrig und gewöhnungsbedürftig wirkt, erschließt Ihnen neue und vor allem sichere Wege zum erfolgreichen Arbeiten mit Excel.

## 2 GRUNDREGELN DER METHODE

### Grundregel 1 Programmierung nur, wenn nötig oder nützlich

Eine dynamische Excel-Lösung wird unter Ausschöpfung aller Möglichkeiten mit Tabellenfunktionalitäten erstellt. Eine Programmierung erfolgt nur dann, wenn es zwingend nötig oder besonders nützlich ist.

### Grundregel 2 Variablen sind Konstanten vorzuziehen

Eine Verwendung von Konstanten werden immer dann vermieden, wenn derselbe Zweck mit Variablen erfüllbar ist. Dies gilt ganz besonders auch für Tabellenformeln.

### Grundregel 3 Redundanz immer dann, wenn von Vorteil

Redundanz wird immer dann praktiziert, wenn sich daraus ein Vorteil für spätere Überprüfungen und Bearbeitungen ableiten lässt.

#### Zur Grundregel 1 (Programmierung nur, wenn nötig oder nützlich)

Es gibt zahlreiche effiziente Lösungen, die ich als so genannte „programmfreie Anwendungen“ bezeichne. Dabei handelt es sich um Modelle, die dynamische Komponenten aufweisen oder sogar hochdynamisch sind (z. B. vollständig mit Mausklicks steuerbar). Sie erfüllen komplette und komplexe Aufgaben oder leisten spezialisierte Dienste für Spezialisten, sind gut geeignet für den Arbeitsplatz des Excel-Laien, den Schreibtisch des Geschäftsführers oder das Notebook des Managers, dürfen sich also durchaus mit dem Titel „Anwendung“ schmücken. Auch dann, wenn Sie keine einzige Zeile Programmcode aufweisen. Die rS1.Methode verzichtet auf Programmierung, wenn die zu lösenden Aufgaben mit einem Tabellenmodell vollständig und effizient zu erledigen sind. Sie können mit einem methodischen und geschickten Einsatz von Steuerelementen und Tabellenfunktionalitäten die Grenzen des gerade formulierten Anspruchs sehr weit hinausschieben. Simpler gesagt: Alles, was Sie ohne Programmierung machen können – und das ist ziemlich viel –, *sollten* Sie auch ohne Programmierung machen.

#### Zur Grundregel 2 (Variablen sind Konstanten vorzuziehen)

Hier wird verlangt, dass der Gebrauch von Konstanten immer dann vermieden werden soll, wenn derselbe Zweck mit Variablen erfüllbar ist. Es mag überraschen, dass dieser Anspruch auch für Arbeitsblätter gelten soll. Wichtig aber ist, besonders in großen Kalkulationsmodellen, die aus Tausenden von Formeln bestehen können, dass die Formeln in Mehrzahl die Werte ihrer Argumente von außerhalb beziehen. Dies gilt übrigens selbst dann, wenn es zunächst unwahrscheinlich ist (besser gesagt: unwahrscheinlich erscheint), dass eine Formel jemals andere Werte benötigt, als bei ihrem Ersteinsatz festgelegt wurde. Das geforderte Verfahren ist im Zusammenhang mit Abb. 1 nachvollziehbar;

D13		fx		=BEREICH.VERSCHIEBEN(rN1.Knoten;C\$13;D\$12)					
	A	B	C	D	E	F	G	H	I
1									
2									
3					01	02	03	04	05
4			01	Deutsch	Aachen	Berlin	Köln	München	Hamburg
5			02	English	Aachen	Berlin	Cologne	Munich	Hamburg
6			03	Français	Aix-la-Chapelle	Berlin	Cologne	Munich	Hambourg
7			04	Nederlands	Aken	Berlijn	Keulen	Munchen	Hamburg
8			05	Italiano	Aquisgrana	Berlino	Colonia	Monaco	Amburgo
9									
10									
11				Ort					
12				3					
13		Sprache	4	Keulen					
14									
15									

Abbildung 1: Die Formel in Zelle D13 bezieht die Werte ihrer Argumente von außerhalb

Die Formel in Zelle *D13* ist eine typische rS1-Formel. Sie benutzt, ausgehend von Zelle *D3*, die den Bereichsnamen *rN1.Knoten* hat, die Variablen aus *D12* und *C13* als Argumente für die Ermittlung der referenzierten Schnittstelle aus Zeile und Spalte – Sprache = 4, Ort = 3 – im Bereich *E4:I8*. (Weiter unten im Abschnitt 3.2 mehr zur Funktion *BEREICH.VERSCHIEBEN*).

### *Zur Grundregel 3 (Redundanz immer dann, wenn von Vorteil)*

Das Argument, man könne, solle oder müsse beim Entwickeln einer Anwendung „Schreibarbeit sparen“ halte ich für ebenso überflüssig wie unsinnig. Ich kann einfach nicht nachvollziehen, wem das nun wirklich nützen soll. Was bringen mir die hübsch gekürzten, elegant verknüpften und rasch geschriebenen Formeln oder Programmzeilen, wenn ich sie vielleicht nach einiger Zeit nicht mehr auf Anhieb verstehe (von den anderen, mir nachfolgenden Verständnissuchenden gar nicht erst zu reden)? Und wird ein Software-Programm durch redundante Schreibweise wirklich langsamer? Oder wird eine Tabellenformel, werden 10.000 Tabellenformeln wirklich schneller abgearbeitet, wenn ich auf alle überflüssigen Argumente in ihnen verzichte? (Lassen Sie uns da nicht über Millisekunden diskutieren, sondern über für den Anwender fühlbare Zeit.) Kurzum, in der rS1.Methode gilt: Redundanz wird praktiziert, wenn sich ein zukünftiger Vorteil daraus ableiten lässt. Das ist immer dann der Fall, wenn redundante Schreibweisen helfen, etwas zu suchen, zu finden, zu ändern oder besser zu verstehen. Dazu mit *Abb. 2* ein kleines Formelbeispiel:

H10      fx      =LINKS(F10;1)&". "&LINKS(G10;1)&". "					
	E	F	G	H	I
9					
10		Anja	Gruhn	A. G.	
11					

**Abbildung 2: Formel mit überflüssigen Argumenten**

Die Formel `=LINKS($F10; 1)&LINKS($G10; 1)` in Zelle *H10* ermittelt aus den Zellen *F10* und *G10* die Initialen eines Namens. Wenn der Funktion *LINKS* ihr zweites Argument – Anzahl\_Zeichen – fehlt, geht Excel davon aus, dass der Wert dieses Arguments *1* ist, also nur ein Zeichen ausgegeben werden soll. Zur rS1.Methode gehört, dass solche optionalen Argumente auch dann in der Formel erscheinen, wenn sie, wie hier, von Excel nicht gebraucht werden. Gebraucht werden sie aber möglicherweise von *Ihnen* – z. B. dann, wenn Sie in einer viel komplizierteren Formel mit vielleicht 250 oder mehr Zeichen und zahlreichen Klammern einen Fehler suchen und Argument für Argument nachvollziehen müssen, was genau dort berechnet wird.

#### **Die Funktion LINKS**

Die Funktion `=LINKS(Text; Anzahl_Zeichen)` zeigt als Resultat so viele Zeichen von Text, wie es dem Argument Anzahl\_Zeichen entspricht. Wenn in Zelle *K11* der Text „Frankfurt am Main“ steht, hat die Formel `=LINKS(K11; 3)` also das Ergebnis „Fra“.

#### **Der Textoperator &**

Mit dem kaufmännischen UND, also dem Zeichen `&`, verbinden Sie Texte. Wenn in *K11* der Vorname „Michael“ steht und in *J11* der Nachname „Friel“ hat die Formel `=K11&" "&J11` das Resultat „Michael Friel“. Sie verkettet den Inhalt von *K11* mit einem Leerzeichen und mit dem Inhalt von *J11*.

#### Grundregel 4

Die vierte und noch nicht erwähnte allgemeine Grundregel: Halten Sie Grundregeln nicht für wichtiger, als sie sind. Wenn Sie mit den Vorgaben der Methode zurechtkommen und erfolgreich arbeiten – sehr gut. Wenn Sie die Regeln nur teilweise umsetzen und ansonsten als Anregung für eigene Varianten benutzen, also eine eigene Mischlösung praktizieren – auch gut. Ihre Modelle und Lösungen würden letztlich auch ohne rS1.Methode funktionieren. Zwar lange nicht so übersichtlich, kaum so pflegeleicht, sicher nicht so wertbeständig. Aber funktionieren, das ja. Wie auch immer Sie letztlich damit umgehen, die wichtigste Empfehlung ist diese: Seien Sie und bleiben Sie bei der Anwendung von Regeln und beim Gebrauch von Methoden auf jeden Fall konsequent. Es lohnt sich.

## 3 REGELN FÜR MAPPEN UND BLÄTTER

Dieser Abschnitt beschäftigt sich mit jenen Regeln und Empfehlungen der Methode, die für Arbeitsmappen und Arbeitsblätter gelten. Sie finden Informationen zu den Themen:

- Arbeitsmappen sowie Arbeitsblätter und deren Gestaltung
- Bereichsnamen und deren Verwendung
- Objektnamen und deren Verwendung
- Besonderheiten, Empfehlungen und Hinweise
- Excel-Vorlagen

### 3.1 Arbeitsmappen und deren Gestaltung

Eine rS1-Arbeitsmappe besteht – jedenfalls in der Entwicklungsphase eines Modells – aus mindestens sechs Arbeitsblättern. Über deren Benennung und Anordnung erfahren Sie mehr weiter unten. Zunächst ein paar Ratschläge zur Organisation der Arbeit und zur Benennung von Dateien.

#### 3.1.1 Dateinamen und Ordner

Bei der Entwicklung eines Excel-Modells können Sie leider sehr viele Fehler machen. Ein Standardproblem ist dabei der Verzicht auf das Speichern von Zwischenständen. Nicht selten aber sind Sie gezwungen, zwei oder drei oder vier Schritte zurückzugehen um eine fehlerhafte Entwicklung zu korrigieren. Gut ist es dann, wenn diese Schritte überhaupt machbar, also Spuren vorhanden sind, in die Sie treten können. Aus diesem Grund habe ich mir eine Ordnung und Vorgehensweise auferlegt, die schon bei vielen Projekten geholfen hat, eine kleine Havarie nicht zum Desaster werden zu lassen.

Zu einigen Einzelheiten der Abb. 3, einer Exploreransicht von Windows Vista:

- Ganz oben im Bild der komplette Pfad der aktuell markierten Datei,
- auf der linken Seite die Datenspeicherstruktur mit den tatsächlichen Namen der Ordner,
- rechts unten eine integrierte, schematische Abbildung mit abstrakten Bezeichnungen dieses Ordnersystems und einer verdeutlichenden Darstellung seiner Ebenen,
- rechts oben die Namen jener Dateien, die im aktuell aktivierten Ordner gespeichert sind.

Sie sehen, dass alle Ordnernamen mit numerischen Präfixen beginnen. Dies hat den großen Vorteil, dass Sie so eine Zugriffsordnung herstellen können, ohne auf eine alphabetisch bedachte Namensgebung angewiesen zu sein.

Ebene 1 ist der Sammelordner Excel-Projekte. Die zweite Ebene des Systems besteht aus den drei Ordnern

- 01 Projekte Aktuell
- 02 Allgemeines
- 03 Projektarchiv

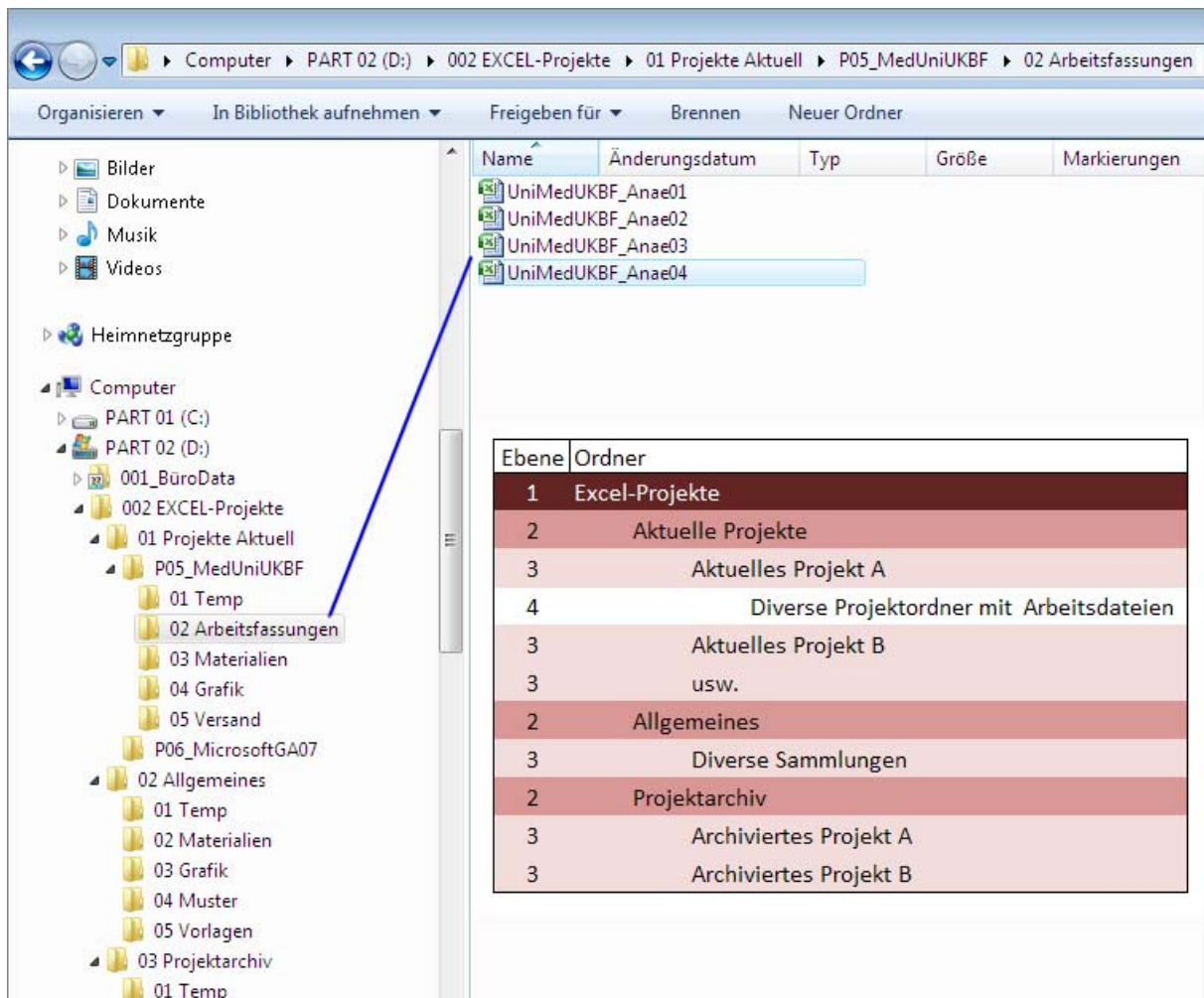


Abbildung 3: Die Struktur des Entwicklungsordners im Windows-Explorer

**01 Projekte Aktuell:** Dieser Ordner nimmt alle Projekte auf, die sich gegenwärtig in Bearbeitung befinden. Als Bearbeitung wird dabei der gesamte Zeitraum von der ersten Ideenskizze bis zur verwendungsfähigen Übergabe des Produkts angesehen. Erst wenn das Produkt völlig fertig gestellt, ggf. vervielfältigt und versandt ist, werden die zu ihm gehörenden Dateien archiviert. Danach allerdings haben sie in 01 Projekte Aktuell nichts mehr zu suchen.

In der nachgeordneten Ebene 3 sehen Sie zwei in Arbeit befindliche Projekte, die dort als *P05\_MedUniUKBF* und als *P06\_MicrosoftGA07* bezeichnet sind. Diese Benennungen gehorchen einem bestimmten System: Ein Präfix mit folgendem Trennzeichen, wie z. B. P05\_ bezeichnet die laufende Nummer des Projekts (natürlich kann es häufig vorkommen, dass Sie an mehreren Projekten gleichzeitig arbeiten müssen). Dem Präfix und Trennzeichen folgt ein Name, der das Projekt möglichst eindeutig einem Auftraggeber oder einem Thema zuordnet.

Zu jedem Projekt wiederum gibt es in der Ebene 4 mehrere standardisierte Ordner in einer festgelegten Reihenfolge:

- **01 Temp:** Dies ist eine Art Zwischenablage. Hier deponiere ich also alles, was ich vorläufig nicht anders zuordnen kann oder will bzw. was ich nur vorübergehend brauche.
- **02 Arbeitsfassungen:** Hier sind die Projektdateien in der Chronologie ihres Entwicklungsfortschritts gespeichert. Es handelt sich also um Versionsspeicherungen des Projekts. Bei komplizierten Entwicklungsarbeiten ist es sehr nützlich, nach jedem gelungenen Arbeitsschritt nicht nur zu speichern (das sowieso), sondern auch jeden als brauchbar bewerteten Zwischenstand als Version zu hinterlegen. Die Dateinamen enden, wie in *Abb. 3* oben rechts gezeigt, mit einer laufenden Versionsnummer.
- **03 Materialien:** Dieser Ordner enthält alle Gebrauchsmaterialien (soweit sie nicht Grafik sind) die zum Projekt und zu seinen Ausstattungen gehören: Planungsdaten, Rohdaten, Vorstudien, Übungsentwürfe, Tabellen, Texte usw.
- **04 Grafik:** Hier werden alle Grafiken hinterlegt, die zum Projekt gehören. Ich versammle in diesem Ordner Objekte, die ich aus größeren, allgemeinen Beständen (z. B. Bildsammlungen) zusammenführe, ggf. projektspezifisch anpasse und unter neuen Namen speichere.
- **05 Versand:** Die Schluss- bzw. Auslieferungsfassung des Produkts wird hier hinterlegt. (Als »Auslieferung« in diesem Sinne gilt auch eine innerbetriebliche Weitergabe.) Der Sinn des Ganzen: Das ist es, was ich als Schlussfassung herausgegeben habe und nichts anderes. Im möglichen Konfliktfall bietet das angenehme Sicherheit.

**02 Allgemeines:** Dieser Ordner der zweiten Ebene enthält in fünf weiteren Subkategorien alles, was allgemeinen Charakter hat, also keinem aktuellen oder archivierten Projekt zugeordnet ist, dennoch aber im ständigen Zugriff des Entwicklers einer Excel-Lösung liegen soll.

**03 Projektarchiv:** Damit Sie auf gute Erfahrungen und brauchbare Ergebnisse zurückgreifen können, ist es sehr wichtig, Gelungenes transferierbar zu halten und zu archivieren. Die Ordnerstruktur sollte also durch das Hinzufügen eines Archivordners ergänzt werden.

Der Vorteil dieser Vorgehensweise liegt nicht nur in der Erleichterung von Rettungsversuchen jeder Art. Sie hilft auch, ein neues und ähnliches Projekt mit erheblichem Minderaufwand zu realisieren. Die gespeicherte Vor- oder Zwischenstufe des alten Projekts kann zahlreiche Vorarbeiten ersetzen oder doch erheblich verkürzen.

### 3.1.2 Arbeitsblätter und deren Inhalte

Vor dem Erscheinen von Excel 2007 war die bis dahin innovativste Einführung einer neuen Excel-Version der Wechsel zu Excel 5. Das Vorhandensein einer Arbeitsmappe (zuvor gab es nur ein Tabellenblatt pro Datei) hat zu einer wahren Explosion der Möglichkeiten und zur erheblichen Verbesserung der Arbeitssicherheit geführt. Seit dieser Zeit gilt in der rS1.Methode der Grundsatz: „Jedem Inhalt sein eigenes Blatt.“ Inhalt in diesem Sinne ist eine bestimmte Kategorie von Daten, wie es nachstehend erläutert ist. Eine rS1-Arbeitsmappe besteht in der Entwicklungsphase eines Modells aus mindestens sechs Arbeitsblättern mit ihren in *Abb. 4* vorgestellten Blattnamen.



Abbildung 4: Die Standardnamen der Arbeitsblätter

Blatt 1: *Focus 1*

Blatt 2: *Basis 1*

Blatt 3: *Daten 1*

Blatt 4: *Listen 1*

Blatt 5: *Parameter 1*

Blatt 6: *Namensliste*

Die Namen der Blätter 1 bis 5 können noch durch ein sinntragendes und mit Leerzeichen abgetrenntes Suffix ergänzt werden. Ein solches Suffix kann auch aus mehreren Zeichenfolgen bestehen. Ebenfalls gültige Blattnamenkombinationen wären also z. B. die in Abbildung 5 gezeigten:

Namen von Arbeitsblättern			rS1.Methode / Regel 1.02
Namen der Arbeitsblätter können beliebige Suffixe erhalten. Zum Beispiel:			
Focus 1 Bericht	Daten 1 Plandaten	Parameter 1 Stammdaten	

**Abbildung 5: Ergänzte Standardnamen von Arbeitsblättern**

Die angegebenen Namen gelten auf jeden Fall für die Entwurfsphase Ihres Modells. Sollten sich nach vollständiger Fertigstellung und Funktionsprüfung andere Namen als sinnvoller erweisen, könnten die Blätter abschließend umbenannt werden. Die Bedeutung dieser Namensgebung und Unterteilung erschließt sich vollständig erst weiter unten, wo die Konventionen der Bereichsnamen erläutert sind. Wenn eine Entwicklung abgeschlossen ist, werden überflüssige Blätter gelöscht. Andere, die vom Anwender nicht eingesehen werden sollen, werden ausgeblendet.

Sollten mehrere gleichartige Blätter eines Typs notwendig sein, werden sie fortlaufend nummeriert, also z. B. *Focus 2*, *Focus 3* usw. Dieser Grundsatz gilt gleichsinnig natürlich auch für alle anderen Blattnamen der Methode.

Namen von Arbeitsblättern			rS1.Methode / Regel 1.03
Arbeitsblätter gleicher Art werden fortlaufend nummeriert. Zum Beispiel:			
Focus 1 Bericht GF	Focus 2 Bericht Budget	Focus 3 KstVergleich	

**Abbildung.6: Mehrere Blätter desselben Typs werden durchnummeriert.**

### *Focus 1* (Darstellung ausgewählter Daten)

Das Blatt *Focus 1* (oder ihm folgende weitere Focusblätter) ist das „Gesicht“ der Anwendung und nimmt alle Daten auf, die der unmittelbaren Betrachtung durch den Anwender bzw. die dem Ausdruck dienen. Es enthält also Diagramme und/oder tabellarisch aufgebaute Ergebnisse wie z. B. einen periodischen Bericht und bietet in vielen Fällen eine dynamische, benutzerdefinierte Datenauswahl an.

### *Basis 1* (Kalkulatorische Grundlage für den Focus)

Das Blatt *Basis 1* enthält die dynamisierbare Datengrundlage für den Focus. Hier werden mit selektierenden und extrahierenden Formeln jene Daten zusammengestellt, die im Focus zu präsentieren sind. Die Formeln dieses Arbeitsblatts benutzen häufig variable Argumente, die vom Anwender mittels Steuerelementen erzeugt werden (Mehr zu Steuerelementen im Abschnitt 3.4.4.)

### *Daten 1* (Rohdatencontainer oder Primärtafel)

Das Blatt *Daten 1* enthält die Quelldaten der Lösung, also alle Daten (entweder als Rohdaten oder besser als bereits strukturierte Primärtafel) einer bestimmten Kategorie bzw. Gruppe, die dort mittels Import oder anderer Verfahren aufgenommen wurden. Soll ein Modell mit seinem Focus auf mehrere Quelldatenkategorien referenzieren, werden gemäß dem Grundsatz „jedem Inhalt sein eigenes Blatt“ mehrere gleichartige Datenblätter angelegt, die dann entsprechende Namen mit fortlaufender Nummerierung erhalten.

### *Listen 1* (Materialien für Steuerelemente)

Das Blatt *Listen 1* ist ein sehr wesentliches Element der rS1.Methode und enthält in der Regel alle Listen, die dem Modell als Grundlage seiner dynamischen Elemente dienen. Dabei handelt es sich überwiegend um Definitionsbereiche für Steuerelemente. Was es damit auf sich hat, erfahren Sie weiter unten.

### *Parameter 1* (Stammdaten und Standards)

Im Blatt *Parameter 1* werden solche Daten hinterlegt, die als konstante oder variable Parameter in verschiedenen Teilen des Modells eine Rolle spielen. Dies sind beispielsweise Stammdaten wie die Bezeichnung und Adresse eines Unternehmens, Namen von Verantwortlichen und Ähnliches. Hinzu kommen Standards wie beispielsweise Prozentsätze für Abgaben und Rabatte oder geprüfte Kennzahlen, die in eine Berechnung von Werten einfließen sollen. Schließlich können Sie dort auch Objekte hinterlegen oder Farben definieren, die an anderer Stelle des Projekts zum Einsatz kommen sollen.

### *Namensliste* (Register der Bereichsnamen)

Namen spielen in der rS1.Methode eine höchst wichtige Rolle – sie sind das zentrale Strukturelement des Verfahrens. Deswegen ist es besonders wichtig, immer darüber orientiert zu sein, welche Namen im Projekt benutzt werden. Das Blatt *Namensliste* soll nichts anderes enthalten als eine vollständige Liste jener Bereichsnamen (einschließlich deren Bezüge), die aktuell in der Arbeitsmappe existieren.

### *Weitere Blattnamen – Möglichkeiten und Grenzen*

Sie können durchaus auch weitere Blätter einrichten und entsprechend den oben gezeigten Grundregeln benennen. Wichtig ist dabei, dass pro Arbeitsmappe jeder Buchstabe des Alphabets nur einmal als erster Buchstabe eines Leitbegriffs (wie *Focus*, *Grafik*, *Daten*, *Basis*) vorkommt. Ein Blatt mit dem Namen *Rechnung 2* ist demzufolge zulässig, wenn es kein anderes *R*-Blatt gibt. Ein Blatt mit dem Namen *Diagramm 1* hingegen wäre unzulässig, wenn es in Konkurrenz mit *Daten 1* stünde. Die Begründung für diese Forderung finden Sie in Abschnitt 3.2.

Namen von Arbeitsblättern		rS1.Methode / Regel 1.04	
Es kann auch andere Blattnamen geben. ABER: Als Führungsbuchstabe darf jeder Buchstabe des Alphabets pro Mappe nur einmal vorkommen.			
Szenario 1	Erlöse 5	Kosten 4	Rechnung 2
Auswertung 2 Gutachten			

Abbildung 7: Auch andere Blattnamen sind (unter eingeschränkten Bedingungen) zulässig

## 3.2 Bereichsnamen und deren Verwendung

Die Verwendung von standardisierten Bereichsnamen spielt in der `rS1.Methode` eine zentrale und äußerst bedeutsame Rolle. Der Einsatz solcher Namen wird die Anfertigung Ihrer Excel-Lösungen wesentlich erleichtern, die Lesbarkeit und Interpretierbarkeit Ihrer Anwendung erheblich verbessern und somit insgesamt eine sehr große Hilfe sein. Bereichsnamen sind auf alles anwendbar, was in Excel als so genannte *Range* angesprochen werden kann, also eine Gruppe von zusammenhängenden Zellen, eine einzige Zelle oder eine Mehrfachauswahl von Zellen (ein Bereich, der aus mehreren nicht zusammenhängenden Zellen oder Zellgruppen besteht).

Ein `rS1`-Bereichsname besteht aus Präfix, Trennzeichen und Suffix:

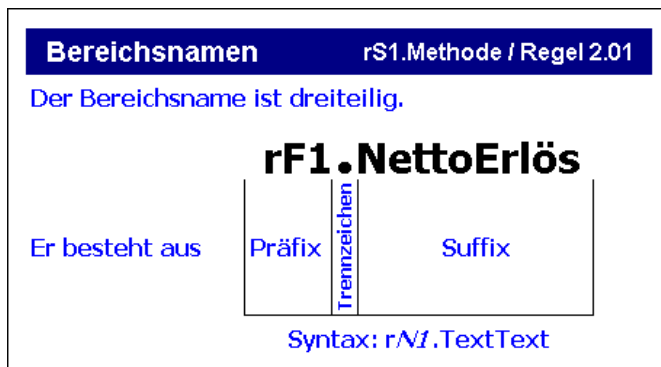


Abbildung 8: Syntax der Bereichsnamen

### Präfix

Das Präfix orientiert sich an dem Namen des Arbeitsblatts, in dem sich der benannte Bereich befindet. Das Präfix `rF1` sagt eindeutig, dass sich der benannte Bereich im Blatt *Focus 1* befindet, das Präfix `rD1` benennt die Lokalisierung des Namens im Arbeitsblatt *Daten 1*. Heißt das Arbeitsblatt z. B. *Parameter 1 Stammdaten*, dann beginnen alle Bereichsnamen in diesem Blatt mit `rP1`. Wurde ein Arbeitsblatt *Statistik 2 Lernerfolg* genannt, beginnen folglich alle seine Bereichsnamen mit `rS2`. Das führende kleine `r` im Präfix weist darauf hin, dass es sich nicht um ein grafisches Objekt handelt, sondern eben um einen Bereich (englisch: „range“). Jedes Arbeitsblatt hat also eigene und blattspezifische Präfixe seiner Bereichsnamen. Das ist der Grund für die oben beschriebene Forderung, dass pro Arbeitsmappe jeder Buchstabe des Alphabets nur einmal als erster Buchstabe eines Leitbegriffs im Arbeitsblattnamen vorkommen soll. Wenn der Bereichsname spezifisch einem bestimmten Blatt zugeordnet ist, kann ich in allen Namensauflistungen oder im Programmcode bereits am Präfix erkennen, wo ich bei Bedarf den so benannten Bereich finden kann. Dies ist für viele Entwicklungsschritte ein unschätzbar großer Vorteil.

### Trennzeichen

Das Trennzeichen der Bereichsnamen ist der Punkt.

### Suffix

Das Suffix, der eigentliche Namenstext, soll möglichst klar und deutlich den Inhalt des benannten Bereichs bezeichnen. Wenn dieser Sinn durch zwei Wörter oder mehr Wörter (beziehungsweise Kürzel) auszudrücken ist (wie bei `rD2.KostenPersonal` oder bei `rP1.ProduktNr` oder bei `rB3.AnzahlPaare`) beginnt jeder neue Sinnbegriff mit einem Großbuchstaben, jedoch werden die Begriffe niemals mit Leerzeichen oder anderen Zeichen getrennt. Grundsätzlich dürfen die Bereichsnamen keine Leerzeichen oder Sonderzeichen enthalten.

## Suffixergänzungen

Das bisher beschriebene Verfahren garantiert, dass bestimmte Namen in einer Arbeitsmappe mehrfach – aber nicht mehrdeutig – vorkommen können, weil sie durch ihre Präfixe eindeutig unterscheidbar sind. Was aber, wenn ein Bereichsname nicht in einer Mappe, sondern – bei der rS1.Methode durchaus üblich – in einem einzigen Blatt mehrfach anzuwenden ist? In diesem Fall wird das Suffix durch eine laufende, in der Regel zweistellige Nummer ergänzt. Diese Nummer ist dann nicht nur Unterscheidungsmerkmal, sondern auch eine sehr nützliche Sortierhilfe.

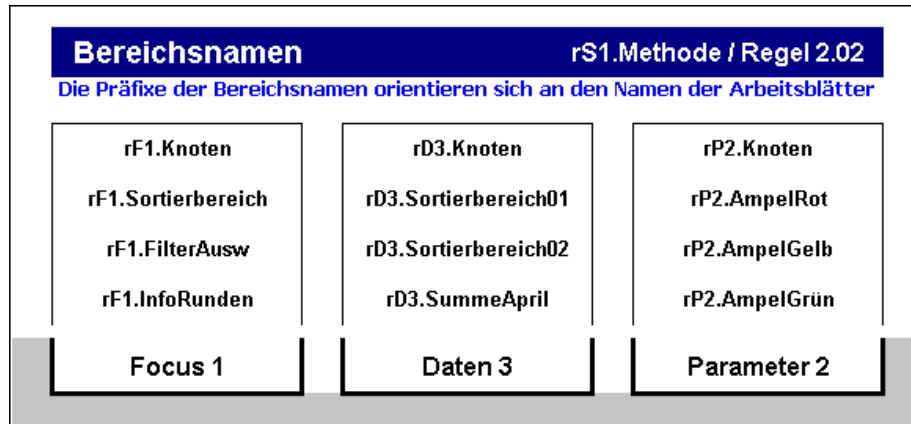


Abbildung 9: Typische rS1-Bereichsnamen

### Besondere Namen 1: Knoten

Zu den gerade angesprochenen mehrfach auftauchenden Namen zählen in rS1-Modellen die so genannten *Knoten*. Funktional ist der Knoten ein standardisierter Ausgangs- oder Ankerpunkt, von dem aus Sie bestimmte Zellen oder Zellbereiche ansteuern können. Er bezeichnet stets eine einzige Zelle, von der aus die bei der rS1.Methode sehr häufig benutzte Funktion `BEREICH.VERSCHIEBEN` in jeder Richtung jede andere Zelle oder auch jeden beliebig großen Zellbereich des Blattes ansprechen bzw. auslesen kann. Das Argument `Bezug` für die spinnenartigen Zugriffsmöglichkeiten einer Formel des Typs `BEREICH.VERSCHIEBEN` ist der *Knoten*.

### Die Funktion `BEREICH.VERSCHIEBEN`

`BEREICH.VERSCHIEBEN(Bezug; Zeilen; Spalten; Höhe; Breite)` hat in der deutschen Fassung leider einen etwas unglücklichen Namen erhalten. In der Ursprungsversion heißt sie schlicht `OFFSET` und das drückt besser aus, was sie kann und macht: Verschoben wird gar nichts, sondern versetzt wird etwas. Die Formel liefert einen Bezug (bzw. den dort vorhandenen Wert), der gegenüber dem angegebenen Argument `Bezug` um eine bestimmte Anzahl von Zeilen und Spalten versetzt ist. Das Argument `Bezug` definiert also den Ausgangspunkt für einen Zugriff auf eine andere Zelle oder einen Zellbereich. Sie können mit dieser Funktion von einer einzigen Zelle aus nicht nur in jeder Richtung andere Zellen ansprechen, sondern auch Bereiche jeder beliebigen Höhe (= Zeilenzahl) und Breite (= Spaltenzahl) abgreifen. Dadurch erhalten Sie die höchst erfreuliche Möglichkeit zur Lösung sehr diffiziler Aufgaben auf Basis sehr simpler Strukturen.

In vielen Blättern müssen Sie keinen anderen Namen vergeben als den für den Knoten. Existiert der Knoten, ist es für das Kalkulationsmodell völlig gleichgültig, ob später einmal der referenzierbare Datenbereich eines Blattes größer oder kleiner wird.

Bereichsnamen		rS1.Methode / Regel 2.03			
Knoten ↓					
→		S1	S2	S3	S4
	Z1	18	33	49	67
	Z2	13	91	61	82
	Z3	55	54	48	83
	Z4	81	98	14	75
	Z5	16	90	78	53

Eine Knotenzelle heißt auch so. Zum Beispiel

**rD3.Knoten**

oder

**rF1.Knoten01**

oder

**rP3.Knoten02**

Abbildung 10: Position des Knotens und Namen von Knoten

Ein Knoten soll sich immer in der linken oberen Ecke einer tabellarischen Struktur befinden, also am Schnittpunkt der vorhandenen bzw. der imaginären Zeilen- und Spaltenbeschriftungen.

#### Besondere Namen 2: Auswahlzellen

Als Auswahlzelle wird eine Zelle bezeichnet, in die der Anwender eine Eingabe macht oder in die das Ergebnis einer Eingabeaktion gelenkt wird. Der Anwender trifft also eine Auswahl, die bei konsequenter Anwendung der rS1.Methode häufig eine Zahl ist. Beispielsweise gibt er – was eher selten vorkommt – die Ziffer 5 direkt in eine Zelle ein. Sehr viel häufiger aber: Er klickt z. B. in einer Auswahlliste auf den fünften Texteintrag, worauf die Ziffer 5 in einer mit dieser Auswahlliste verknüpften Zelle erscheint.

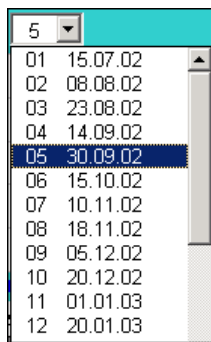


Abbildung 11: Der Klick auf den fünften Listeneintrag erzeugt in einer verbundenen Zelle die Ziffer 5

In allen solchen Fällen hat die aufnehmende Zelle immer die Suffixergänzung *Ausw.* Der Vorteil: Sie können in entsprechenden Formeln und im Programmcode jederzeit erkennen, ob sich ein weiter zu verarbeitender Wert auf eine Auswahlentscheidung des Anwenders bezieht.


Bereichsnamen		rS1.Methode / Regel 2.04	
Eingabe		Eingaben macht der Anwender direkt (manuell + Enter) oder indirekt (mit einem Steuerelement)	
Das Suffix einer Eingabezelle endet immer mit der Zeichenfolge "Ausw" für "Auswahl (des Anwenders)"			
rF1.PlanzahlAusw		oder	rL1.RegionAusw

Abbildung 12: Das Kürzel „Ausw“ steht immer für eine Auswahl des Anwenders

Unter den zahlreichen Vorteilen dieser Namenskonventionen ist das Herstellen einer vielfältig nutzbaren Ordnung und Übersicht von herausragender Bedeutung. In jeder Excel-Auflistung, die Namen verwaltet, steht alles zusammen, was zusammen gehört. Sie können obendrein meistens auch leicht erkennen, was sich hinter dem Namen verbirgt.

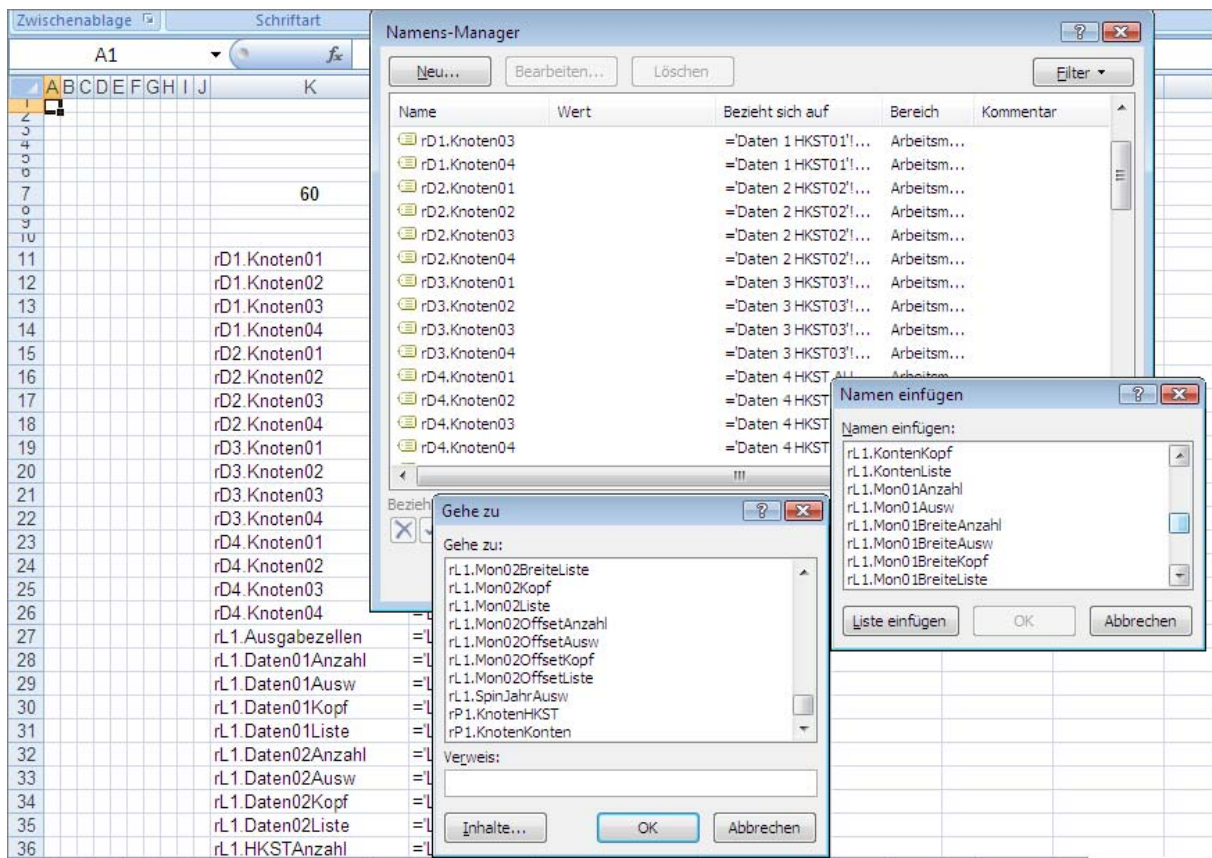


Abbildung 13: Klarheit und Übersicht in Namensauflistungen

Auch bei einer Programmierung wird auf den ersten Blick deutlich (vgl. Abb. 14), an welchen Stellen Bereichsnamen eine Rolle spielen. Besser noch und besonders hilfreich: Sie können bereits an einem Namen erkennen, wo sich der so benannte Bereich befindet und welche Inhalte er höchstwahrscheinlich hat.

Bereichsnamen im VBA-Code		rS1.Methode / Beispiel
<b>Bereich</b>	=	Range
<b>Beispiele:</b> <code>Range("rF1.Sortierbereich").Select</code>		
Den Bereich auswählen, der den Namen "rF1.Sortierbereich" trägt		
<code>Range("rP1.Knoten").Offset(2, 5).Copy</code>		
Eine Zelle kopieren, die von der Zelle mit dem Namen "rP1.Knoten" um zwei Zeilen nach unten und 5 Spalten nach rechts versetzt ist.		

Abbildung 14: Bereichsnamen im Programmcode

### 3.3 Objektnamen und deren Verwendung

Auch Steuerelemente und grafische Elemente erhalten bisweilen spezifische Namen. Diese sind allerdings nur dann von Bedeutung, wenn bei späteren Ergänzungen und Erweiterungen einer Lösung mit Programmcode auf solche Objekte zugegriffen werden soll.

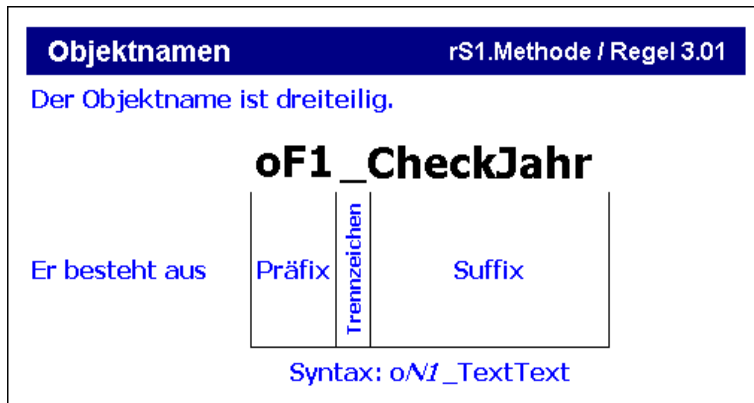


Abbildung 15: Syntax der Objektnamen (Beispiel: Objekt im Blatt Focus 1 ...)

Wie die Bereichsnamen werden auch Objektnamen nach dem Muster Präfix – Trennzeichen – Namenstext gebildet. Typische Objektnamen sind beispielsweise

- *oF1\_OptWarnung* (wobei das Kürzel „Opt“ für einen OptionButton steht)
- *oF2\_BoxKst* (wobei das Kürzel „Kst“ für Kostenstellen steht)
- *oG3\_SpinRegionen* (wobei das Kürzel „Spin“ für einen SpinButton steht)

Vgl. dazu auch Abb. 16 und den zu Abb. 16 gehörenden Text.

#### Präfix

Das Präfix ist wiederum an dem Namen des Arbeitsblatts orientiert, in dem sich das Objekt befindet. Das führende kleine *o* im Präfix weist darauf hin, dass es sich nicht um einen Bereich handelt, sondern eben um ein (grafisches) Objekt.

#### Trennzeichen

Das Trennzeichen der Objektnamen ist der Unterstrich.

#### Bei Objektnamen kein Punkt als Trennzeichen:

Sie dürfen hinsichtlich einer eventuellen Excel-Programmierung in einem Objektnamen nicht den Punkt als Trennzeichen benutzen, weil dies im Programm zu bestimmten Fehlern führen wird!

#### Namenstext

Der eigentliche Namenstext soll Art, Sinn oder Inhalt des Objekts ausdrücken.

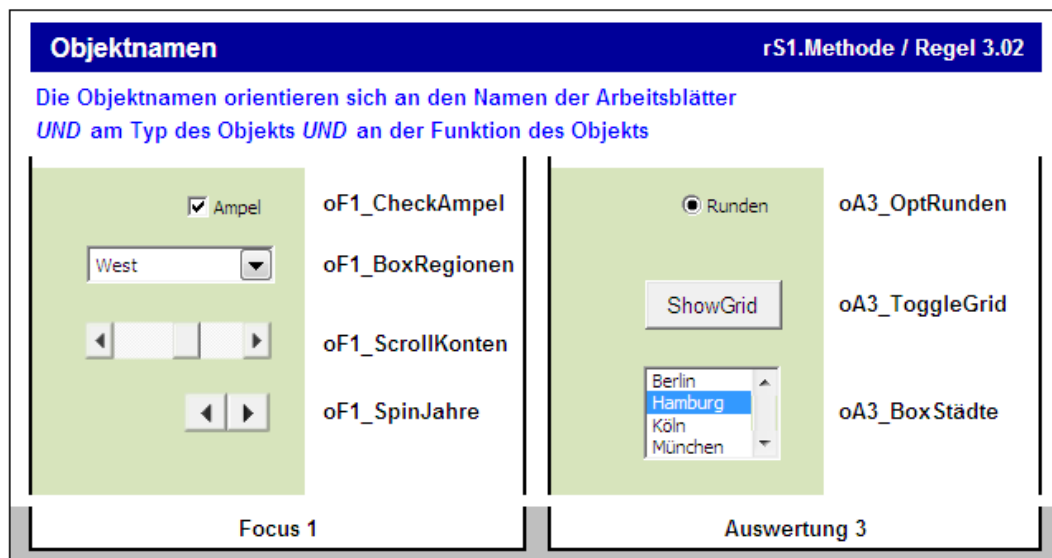


Abbildung 16: Im Objektnamen ist auch die Art des Objekts erkennbar

Die Art des Objekts wird im ersten Teil des Suffixes bezeichnet. In Abb. 16 sehen Sie sieben Steuerelemente und jeweils rechts daneben deren exemplarische Objektnamen. (Mehr zu Steuerelementen im Abschnitt 3.4.4.) Der erste Teil des Namenssuffixes bezieht sich auf die *Art* des Objekts:

Links, von oben nach unten:

- *CheckBox* (Kontrollkästchen), das Suffix beginnt mit *Check*
- *ComboBox* (Kombinationsfeld), das Suffix beginnt mit *Box*
- *ScrollBar* (Bildlaufleiste), das Suffix beginnt mit *Scroll*
- *SpinButton* (Drehfeld), das Suffix beginnt mit *Spin*

Rechts, von oben nach unten:

- *OptionButton* (Optionsfeld), das Suffix beginnt mit *Opt*
- *ToggleButton* (Umschaltfläche), das Suffix beginnt mit *Toggle*
- *ListBox* (Listenfeld), das Suffix beginnt mit *Box*

### ComboBox und ListBox

Da die Verwaltung und Programmierung dieser beiden Elemente nahezu identisch ist, muss auch bei der Namensvergabe kein Unterschied gemacht werden: Beide Suffixe beginnen also mit *Box*.

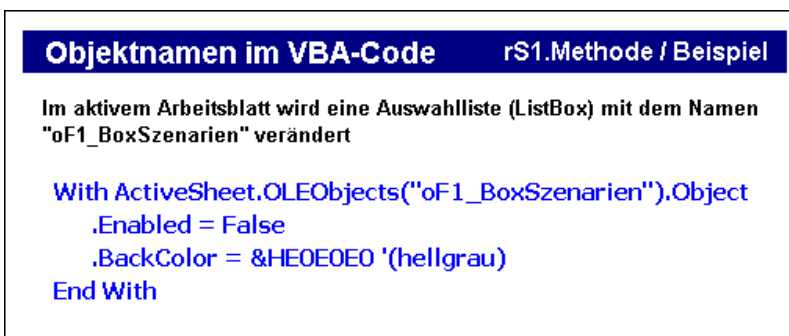


Abbildung 17: Objektnamen im Programmcode

## 3.4 Besonderheiten, Empfehlungen und Hinweise

In diesem Abschnitt – er gehört immer noch zum Thema „Arbeitsblätter“ – erfahren Sie einiges über bestimmte Details und Besonderheiten der rS1.Methode. Weiterhin werden Sie über nützliche Vorgehensweisen informiert.

### 3.4.1 K11 als linke obere Zelle/Hilfszeilen und Hilfsspalten

Alle Blätter einer rS1-Standard-Arbeitsmappe haben Hilfszeilen und Hilfsspalten. Dies sind Bereiche, die während und auch nach der Entwicklungsphase funktional unterstützende Strukturen aufnehmen. Dazu gleich mehr. Die Hilfszeilen 1 bis 10 haben die Zeilenhöhe 8. Die Hilfsspalten A bis J haben die Spaltenbreite 1. Aus dieser Struktur ergibt sich, dass die oberste linke Zelle des eigentlichen Funktionsbereichs einer Tabelle die Zelle *K11* ist.

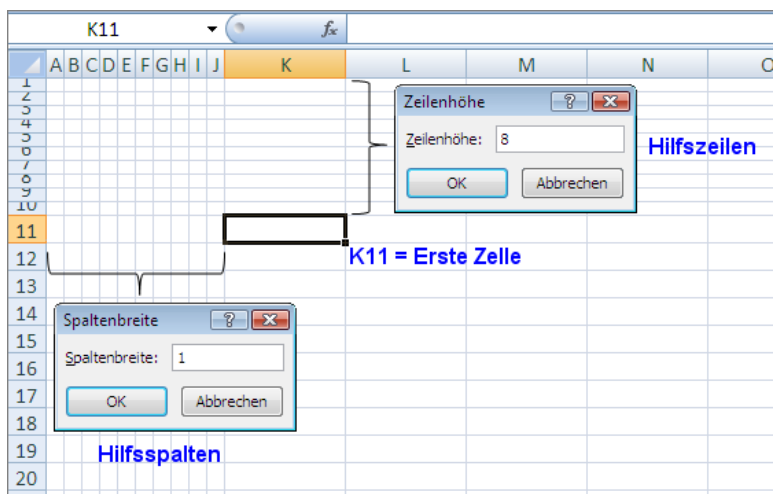


Abbildung 18: Hilfszeilen und Hilfsspalten/K11 ist obere linke Zelle

Weiter oben wurde schon als wichtig erwähnt, dass besonders in großen Kalkulationsmodellen, die meisten Formeln die Werte ihrer Argumente von außerhalb beziehen. „Außerhalb“ – das sind z. B. auch die Hilfsspalten oder Hilfszeilen. Die Formeln des Kalkulationsmodells stehen unterhalb und rechts von Zelle *K11*. Werte, die von den Formeln zur Veränderung ihrer variablen Argumente benötigt werden, stehen oberhalb und links von *K11*. Nach Abschluss der Entwicklung sind die Hilfszeilen und Hilfsspalten nicht mehr sichtbar. Sie werden ausgeblendet, mit identischen Schrift- und Hintergrundfarben unsichtbar gemacht und gelegentlich auch mit anderen Strukturen, z. B. mit Textfeldern, überdeckt. Eine Momentaufnahme aus der Entwicklungszeit selbst sehen Sie in *Abb. 19*. Hier sind jene Strukturen noch sichtbar, die dem Anwender normalerweise verborgen bleiben:



### 3.4.3 Berechnungsmodus

Excel führt bei vielen Aktionen eine Neuberechnung der gesamten Mappe durch, auch dann, wenn es (zunächst) nicht logisch erscheint. Jede Neuberechnung braucht ihre Zeit und bei Modellen mit vielen Tausend Formeln kann das schon ein paar Sekunden dauern. Dennoch sollte dieser Automatismus nicht unterdrückt werden, denn die meisten rS1-Kalkulationsmodelle benötigen wegen der manchmal großflächigen Vernetzung ihrer Formeln eine ständige, heißt automatische Berechnung.

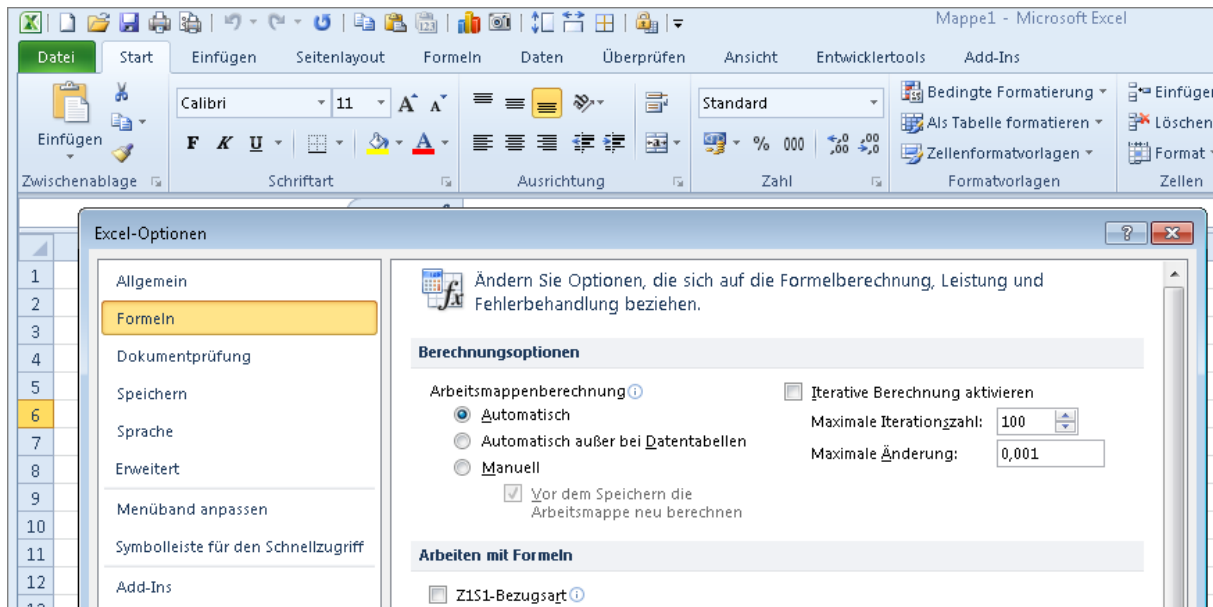


Abbildung 21: Automatische Berechnung ist meistens erforderlich

Die Festlegung treffen Sie, wie in Abb. 21 angedeutet, mit dem Befehlsweg Registerkarte *Datei/Optionen/Formeln/Abschnitt Berechnungsoptionen/Automatisch*. Dann sollten Sie Ihr Modell mehrfach testen um zu sehen, ob die Rechenzeiten tolerabel sind. Bei modernen Computern mit schnellen Prozessoren, ausreichendem Arbeitsspeicher und leistungsfähigen Grafikkarten erzeugen auch 5.000 oder mehr kompliziert aufgebaute Formeln kein echtes Zeitproblem. Sollten Sie jedoch zu der Entscheidung kommen, dass Ihre Lösung wegen der ständigen Neuberechnung zu langsam ist, kann nur zweierlei helfen. Erstens: Sie werfen Ballast über Bord. Da Sie aber nach allen Regeln der Kunst und obendrein nach der rS1.Methode gearbeitet haben, finden Sie kaum Ballast. Dann also zweitens: die Programmierung. Hier können Sie dann die automatische Berechnung ausschalten und per Programmcode festlegen, wann und was berechnet wird.

### 3.4.4 Steuerelemente

Wenn Sie nach den Regeln der rS1.Methode arbeiten, sind die in Excel verfügbaren Steuerelemente von zentraler Bedeutung für die Dynamisierung Ihrer Lösungen. Das dabei benutzte Prinzip ist recht einfach: Sie treffen in oder mit einem Steuerelement durch Mausklick eine Auswahl und erzeugen dadurch einen Ausgabewert, den Sie in eine von Ihnen bestimmbare Zelle lenken. Solche Ausgabewerte sind z. B.:

- variable Zahlen, die Sie direkt oder indirekt als Argument einer Formel benutzen können *oder*
- logische Wahrheitswerte (WAHR oder FALSCH), die Sie direkt oder indirekt mit z. B. einer WENN-Formel verarbeiten können *oder*
- Texte, den Sie z. B. direkt oder indirekt als das Argument Suchkriterium einer Formel (wie etwa in SVERWEIS oder SUMMEWENN) benutzen können.

Weiter oben haben Sie, im Zusammenhang mit Objektnamen (Abb. 16), schon eine kurze Information zu diesen hervorragenden Arbeitshilfen erhalten. Hier nun weitere Details:

### Zugang zu Steuerelementen

Die Befehle zur Einrichtung und Gestaltung von Steuerelementen befinden sich im Menüband/Registerkarte *Entwicklertools*/Gruppe *Steuerelemente*.

Die Registerkarte *Entwicklertools* wird nach der Installation von Excel 2010 nicht automatisch angezeigt. Sie kann jedoch auf dem folgenden Befehlsweg dauerhaft verfügbar gemacht werden: Registerkarte *Datei/Optionen/Menüband anpassen*, dann unter *Hauptregisterkarten* das Kontrollkästchen *Entwicklertools* aktivieren.

Einrichtung unter Excel 2007: *Office-Schaltfläche/Excel-Optionen/Registerkarte Häufig verwendet/Entwicklerregisterkarte ... anzeigen*.

Zugang unter Excel 2003: Symbolleiste *Formular* und Symbolleiste *Steuerelement-Toolbox*.

Grundsätzlich können Sie mit zwei verschiedenen Arten von Steuerelementen Ihren Anwendungen „Leben einhauchen“. Bei der rS1.Methode kommen beide zum Einsatz:

- Verwenden Sie die schlichten und kaum formatierbaren *Formularsteuerelemente* für einfache Aufgaben in solchen Lösungen, die keine aufwändige und formal anspruchsvolle Gestaltung benötigen. Diese Elemente sind in jederlei Hinsicht relativ anspruchslos und lassen bei ihrer Funktionalisierung und Nutzung kaum Probleme erwarten.
- Verwenden Sie die umfangreich gestaltbaren *ActiveX-Steuerelemente* in solchen Lösungen, die hohe Anforderungen an das Erscheinungsbild stellen und deshalb auch diverse Formatierungsmöglichkeiten von Steuerelementen verlangen. Das Arbeiten mit Elementen dieser Gruppe ist relativ anspruchsvoll und kann bei einigen Systemumgebungen unter bestimmten Bedingungen auch zu kleinen Darstellungsfehlern führen. Sie bieten jedoch in den so genannten „programmfreien Anwendungen“ weit gefächerte Möglichkeiten der Gestaltung und lassen sich überdies hervorragend mit Programmcode beeinflussen (*ActiveX-Steuerelemente* wurden primär für den Einsatz in programmierten Modellen geschaffen, ihre Verwendung zur Dynamisierung nicht programmierter Lösungen gehört zu den Eigenarten der rS1.Methode.)

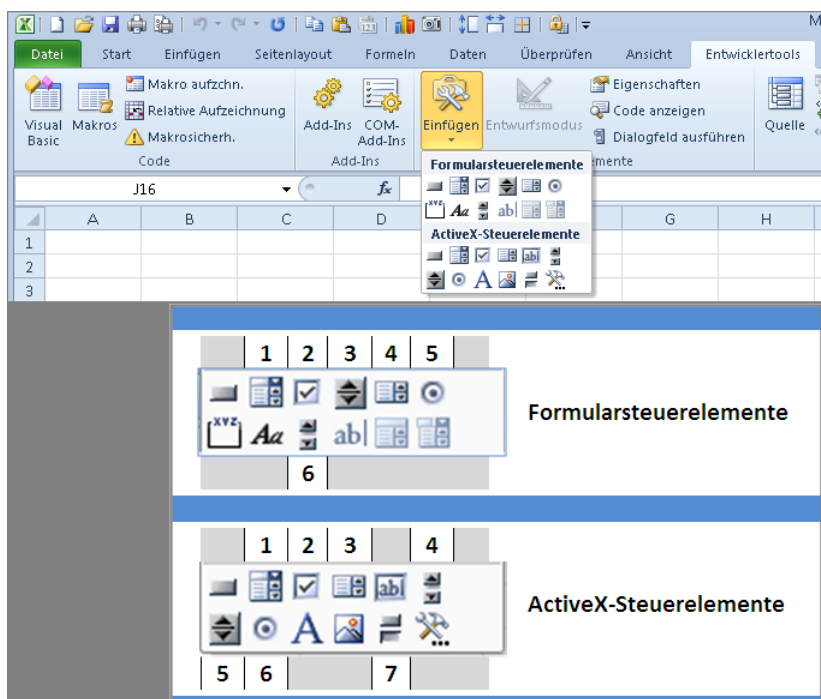









Abbildung 22: Die verfügbaren Steuerelemente

Im nachstehenden Text werden für die *Formularsteuerelemente* in erster Linie die deutschen Bezeichnungen und für die *ActiveX-Steuerelemente* in erster Linie die englischsprachigen Bezeichnungen benutzt. Letzteres deswegen, weil Sie bei Nutzung der *rS1.Methode* und bei der Verwendung in Excel auch mit diesen Namen umgehen und arbeiten müssen.

In Abb. 22 mit Nummern versehen und der folgenden Tabelle aufgelistet sind nur solche Elemente, die in nach der *rS1.Methode* angefertigten, nicht programmierten Lösungen zum Einsatz kommen.

<i>Formularsteuerelemente</i>	<i>ActiveX-Steuerelemente</i>
1 Kombinationsfeld ( <i>ComboBox</i> )	1 ComboBox (Kombinationsfeld)
2 Kontrollkästchen ( <i>CheckBox</i> )	2 CheckBox (Kontrollkästchen)
3 Drehfeld ( <i>SpinButton</i> )	3 ListBox (Listenfeld)
4 Listenfeld ( <i>ListBox</i> )	4 ScrollBar (Bildlaufleiste)
5 Optionsfeld ( <i>OptionButton</i> )	5 SpinButton (Drehfeld)
6 Bildlaufleiste ( <i>ScrollBar</i> )	6 OptionButton (Optionsfeld)
	7 ToggleButton

Welches Steuerelement Sie für welchen Zweck benutzen, ist natürlich Ihnen überlassen. Sie sollten jedoch bestimmte Grundregeln beachten und sich dabei an der Konstruktionsart dieser Werkzeuge orientieren. Dazu einige Anmerkungen:

<i>Steuerelement</i>	<i>Aktionen</i>
 ComboBox (Kombinationsfeld)	Aus einer Liste von Elementen ein Element auswählen. (Es handelt sich um ein Textfeld mit Dropdownliste. Optional ist eine manuelle Eingabe als Auswahl-Vorgabe möglich.)
 CheckBox (Kontrollkästchen)	Eine Option durch Aktivierung oder Deaktivierung ein- oder ausschalten.
 ListBox (Listenfeld)	Aus einer Liste von Elementen ein Element auswählen. (Alle Elemente sind sichtbar.)
 ScrollBar (Bildlaufleiste)	Einen Wert per Mausklick in Einzelschritten erhöhen oder vermindern oder mit gedrückter Maustaste großflächig und schnell durch komplexe Wertebereiche blättern.
 SpinButton (Drehfeld)	Einen Wert per Mausklick in Einzelschritten erhöhen oder vermindern.
 OptionButton (Optionsfeld)	Aus einer Gruppe von Optionen eine Option auswählen.
 ToggleButton	Eine Option ein- oder ausschalten.

### Weitere Informationen

zur Einrichtung von und zur Arbeit mit Steuerelementen finden Sie in dem Buch, dem dieses Dokument beigegeben ist.

### 3.4.5 Namen im Arbeitsblatt *Listen 1*

Die vorangestellte Kurzbeschreibung der Steuerelemente war auch erforderlich, um die jetzt folgenden Ausführungen verständlicher zu machen. Denn die Steuerelemente haben – ihrer Bedeutung entsprechend – in jeder nach der *rS1.Methode* gestalteten Arbeitsmappe eigene Definitionsbereiche. Diese Bereiche werden vorzugsweise im Arbeitsblatt *Listen 1* angelegt. Hier gelten weitere eindeutige Konventionen.

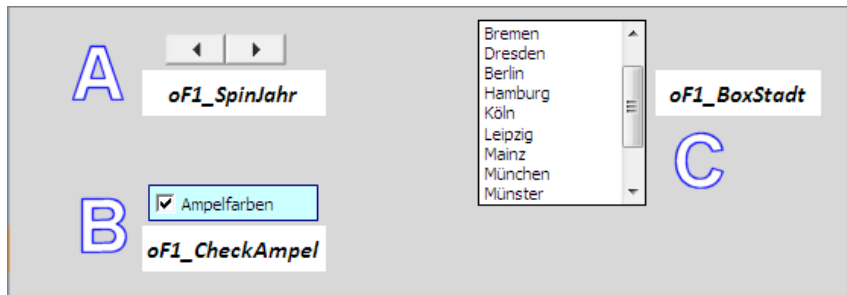


Abbildung 23: Steuerelemente und deren Objektnamen

Wie Sie beim Thema *Objektnamen* (Abschnitt 3.3) bereits erfahren haben, gelten für die Benennung von Steuerelementen spezifische Regeln. Drei der Elemente sehen Sie exemplarisch in *Abb. 23*, zusammen mit den dafür vergebenen Namen. Dabei wurde unterstellt, dass die Steuerelemente in einem Arbeitsblatt mit dem Namen *Focus 1* angelegt wurden, deshalb also das Namenspräfix *oF1*.

- **A:** Mit einem als *oF1\_SpinJahr* benannten *SpinButton* klickt sich der Anwender durch mehrere Jahre. Im Hintergrund könnten dadurch z. B. Budgetdaten verschiedener Geschäftsjahre angesprochen und in den Focus eingelesen werden. Ein Klick auf einen *SpinButton* verändert Schritt für Schritt den Ausgabewert dieses Steuerelements. Der Ausgabewert ist eine Zahl, die in eine definierbare Zelle gelenkt wird. Diese Zelle wird bei Verwendung der *Formularsteuerelemente* als *Zellverknüpfung* bezeichnet, bei den *ActiveX-Steuerelementen* hingegen ist es die sog. *LinkedCell*.
- **B:** Mit der als *oF1\_CheckAmpel* benannten *CheckBox* entscheidet der Anwender z. B., ob seine Daten im Focus mit Ampelfarben angezeigt werden, um bestimmte Probleme zu signalisieren. Ein Klick auf eine *CheckBox* verändert den Ausgabewert dieses Steuerelements. Der Ausgabewert ist ein logischer Wahrheitswert (also WAHR oder FALSCH), der in die *Zellverknüpfung* bzw. in die *LinkedCell* gelenkt wird.
- **C:** Mit der als *oF1\_BoxStadt* benannten *ListBox* trifft der Anwender eine Auswahl unter verschiedenen Städten. Im Hintergrund könnten dadurch z. B. regionale Verkaufserlöse eines Produkts angesprochen und in den Focus eingelesen werden. Ein Klick auf ein Element einer *List-Box* verändert den Ausgabewert dieses Steuerelements. Der Ausgabewert ist (falls bei Verwendung von *ActiveX-Steuerelementen* nicht anderes bestimmt wurde) eine Zahl, die in die *Zellverknüpfung* bzw. in die *LinkedCell* gelenkt wird.

Der Anwender entscheidet also etwas, er trifft per Mausklick eine *Auswahl*. Sein Auswahlergebnis wird in die *Zellverknüpfung* bzw. in die *LinkedCell* gelenkt und anschließend von Formeln verwertet. Solche Zellen sollen, wie weiter oben schon beschrieben, immer die Suffixergänzung *Ausw* (für *Auswahl*) haben und sind normalerweise im Blatt *Listen 1* angesiedelt.

Für alle Steuerelemente, die nicht *ListBox* oder *ComboBox* sind, befinden sich die *Zellverknüpfungen* bzw. *LinkedCells* in Spalte L des Blattes *Listen 1* (oder, falls erforderlich, in *Listen 2* und so weiter). Meistens oberhalb der Zelle steht nachrichtlich der Name des Steuerelements, zu dem diese *Zellverknüpfung* bzw. *LinkedCell* gehört. Für die Benennung der Zelle selbst gilt folgende Syntaxregel:

- Präfix aus Blattname, also in der Regel *rLI*, dann als Trennzeichen der Punkt
- Suffix aus Benennung des verbundenen Steuerelements, abschließend die Zeichenfolge *Ausw*.

(vgl. auch die Informationstexte in *Abb. 24*, Spalte K)

Das hört sich komplizierter an, als es ist. Denn folglich gehört zum Objekt *oF1\_SpinJahr* eine *LinkedCell* mit dem Namen *rL1.SpinJahrAusw*. Die Auswahl des Steuerelements *oF1\_CheckAmpel* wird in eine *LinkedCell* mit dem Namen *rL1.CheckAmpelAusw* gelenkt. Und ein *ToggleButton* mit dem Namen *oF1\_ToggleModus* wäre also mit einer *LinkedCell* zu verbinden, die den Namen *rL1.ToggleModusAusw* hat. Die Logik dieser Namensgebungen wird sich Ihnen in der täglichen Arbeit rasch erschließen und Sie werden sich dann in vielen Zusammenhängen, ganz besonders aber während der Konstruktion – oder auch bei einer notwendigen Rekonstruktion – Ihrer Lösung über die Nützlichkeit dieser Namensgebungen freuen können.

	FECEEF	G	H	I	J	K	L	M	N	O	P	Q
1												
2												
3												
4												
5												
6												
7									5		← rL1.StadtAusw	muss
8												
9									12		← rL1.StadtAnzahl	kann
10												
11									Stadt		← rL1.StadtKopf	kann
12		1						Bremen				
13		2						Dresden				
14		3				rL1.SpinJahrAusw →	1	Berlin				
15		4						Hamburg				
16		5						Köln				
17		6				rL1.CheckAmpelAusw →	WAHR	Leipzig			← rL1.StadtListe	muss
18		7						Mainz				
19		8						München				
20		9						Münster				
21		10						Nürnberg				
22		11						Rostock				
23		12						Stuttgart				

Abbildung 24: rS1-Bereichsnamen im Blatt *Listen 1*

Damit zu den Listendefinitionsbereichen auf der rechten Seite von Abb. 24. Derartige Listendefinitionen werden für die Steuerelemente *ComboBox* (*Kombinationsfeld*) und *ListBox* (*Listenfeld*) benötigt.

- Es gibt einspaltige und mehrspaltige Listendefinitionsbereiche. Die Abbildung zeigt den einspaltigen Standard. Der Titel der Liste soll möglichst kurz sein und wird ohne Leerzeichen oder Sonderzeichen in die Zeile 11 der jeweiligen Spalte eingetippt. Diese Zelle bildet den Listenkopf und erhält (optional) als Bereichsnamen das Standardpräfix, also *rL1*, gefolgt vom Standard-Trennzeichen, gefolgt vom Text, der in der Zelle steht, gefolgt von der Suffixergänzung *Kopf*. Die Zelle *N11* mit dem Text *Stadt* bekommt also den Namen *rL1.StadtKopf*. Stünde in einer solchen Listenkopfzelle z. B. der Text *Monate02*, bekäme sie folglich den Namen *rL1.Monate02Kopf*.  
Diese Namensvergabe für den Listenkopf ist ein KANN. Der Name ist in programmfreien Anwendungen in aller Regel überflüssig. In programmierten Lösungen kann er verschiedentlich von großem Nutzen sein.
- Unterhalb des Kopfes folgt in lückenloser Zeilenreihe der eigentliche Inhalt der Liste, der *Eingabebereich* (wie es etwas unglücklich bei den *Formularsteuerelementen* heißt) bzw. die *ListFillRange* (wie es eindeutiger bei Verwendung der *ActiveX-Steuerelemente* genannt wird). Dieser Bereich erhält nach Präfix und Trennzeichen als Namenstext den Textinhalt des Listenkopfes, gefolgt von der Zeichenfolge *Liste*. Im abgebildeten Beispiel heißt dieser Definitionsbereich also recht logisch *rL1.StadtListe*. Stünde in der Listenkopfzelle *Monate02*, bekäme die *ListFillRange* folglich den Namen *rL1.Monate02Liste*. Die Namensvergabe für eine *ListFillRange* ist bei Anwendung der *rS1.Methode* ein MUSS. Der Name wird sowohl in programmfreien Anwendungen wie auch in programmierten Lösungen benötigt.

- In Zeile 9 der betreffenden Spalte wird mit einer Formel gezählt, aus wie vielen Elementen die Liste besteht; anders gesagt, wie viele Einträge in der *ListFillRange* vorhanden sind. Benutzt wird dazu die Funktion *ANZAHL2*. Im abgebildeten Beispiel lautet die entsprechende Formel `=ANZAHL2(rL1.StadtListe)`. Die Zelle *N9* erhält als Namenstext den Textinhalt des Listenkopfes, gefolgt von der Zeichenfolge *Anzahl*, also *rL1.StadtAnzahl*. Diese Namensvergabe ist ein KANN. Der Name ist in programmfreien Anwendungen überflüssig. In programmierten Lösungen kann er von Nutzen sein.

### Die Funktion ANZAHL2

Die Funktion `=ANZAHL2(Bezug)` zählt die Anzahl von Daten (Zellinhalten) in Bezug, unabhängig von der Art der Inhalte. Gezählt werden also beliebige Textzeichenfolgen und Zahlen. Die verwandte Funktion *ANZAHL* hingegen zählt nur die Anzahl von *Zahlen* in einem Bereich.

- In Zeile 7 der betreffenden Spalte schließlich befindet sich die *Zellverknüpfung* bzw. *Linked-Cell* der Liste. Hier erscheint nach jedem Klick auf einen Eintrag der *ComboBox* (*Kombinationsfeld*) bzw. *ListBox* (*Listenfeld*) eine Zahl, die der Auswahl des Anwenders entspricht. Die Zelle *N7* erhält als Namenstext den Textinhalt des Listenkopfes, gefolgt von der Zeichenfolge *Ausw*, im Beispiel also *rL1.StadtAusw*. Die Namensvergabe für diese Zelle ist bei Anwendung der *rS1.Methode* ein MUSS.

### Empfehlungen für den Überschriftentext im Listenkopf

Wie Sie gelesen haben, ist der Texteintrag im Listenkopf das zentrale Element für alle Benennungen und weiteren Vorgänge, seien sie manuell oder automatisch durchgeführt. Der Eintrag sollte relativ kurz sein. Verwenden Sie Abkürzungen dennoch nur dann, wenn sie von hoher Eindeutigkeit sind. Der Text soll weiterhin prägnant den Inhalt der Liste bezeichnen und keine Leer- oder Sonderzeichen enthalten. Sie werden in einigen Beispielen sehen, dass es nötig sein kann, inhaltlich identische Listen doppelt zu führen. So etwa, wenn in Vergleichsberichten Daten unterschiedlicher Monate per Mausklick nebeneinandergestellt werden sollen. Für diesen Zweck muss es zwei Boxen mit Monatsnamen geben, also auch zwei Definitionsbereiche für die entsprechenden Listen. In solchen Fällen wird den Texten des Listenkopfs eine Nummer angehängt. Mit *Monate01*, *Monate02* und ähnlichem lassen sich die erforderlichen Unterscheidungen treffen.

## 3.4.6 Namen definieren (festlegen)

Bisher war schon recht viel von Namenskonventionen die Rede. Ergänzend am Beispiel der *Abb. 25* einige Anmerkungen zu den Methoden der Namensfestlegung.

Die Zelle *K11* soll den Namen *rD1.Knoten* erhalten. Markieren Sie die Zelle und verfahren Sie dann wie folgt.

- Variante A: Geben Sie den Namen in das *Namenfeld* auf der linken Seite der Bearbeitungsleiste ein und legen Sie ihn mit Drücken der Taste [Enter] fest. (Unsichere Methode, hier nur der Vollständigkeit halber erwähnt.)
- Variante B: Aktivieren Sie im Menüband die Registerkarte *Formeln* und klicken Sie in der Gruppe *Definierte Namen* auf den Eintrag *Namen definieren*. Geben Sie anschließend den Namen im Dialogfeld *Neuer Name* ein. In diesem Dialogfeld können Sie auch noch weitere Entscheidungen zur Verwaltung dieses Namens treffen.

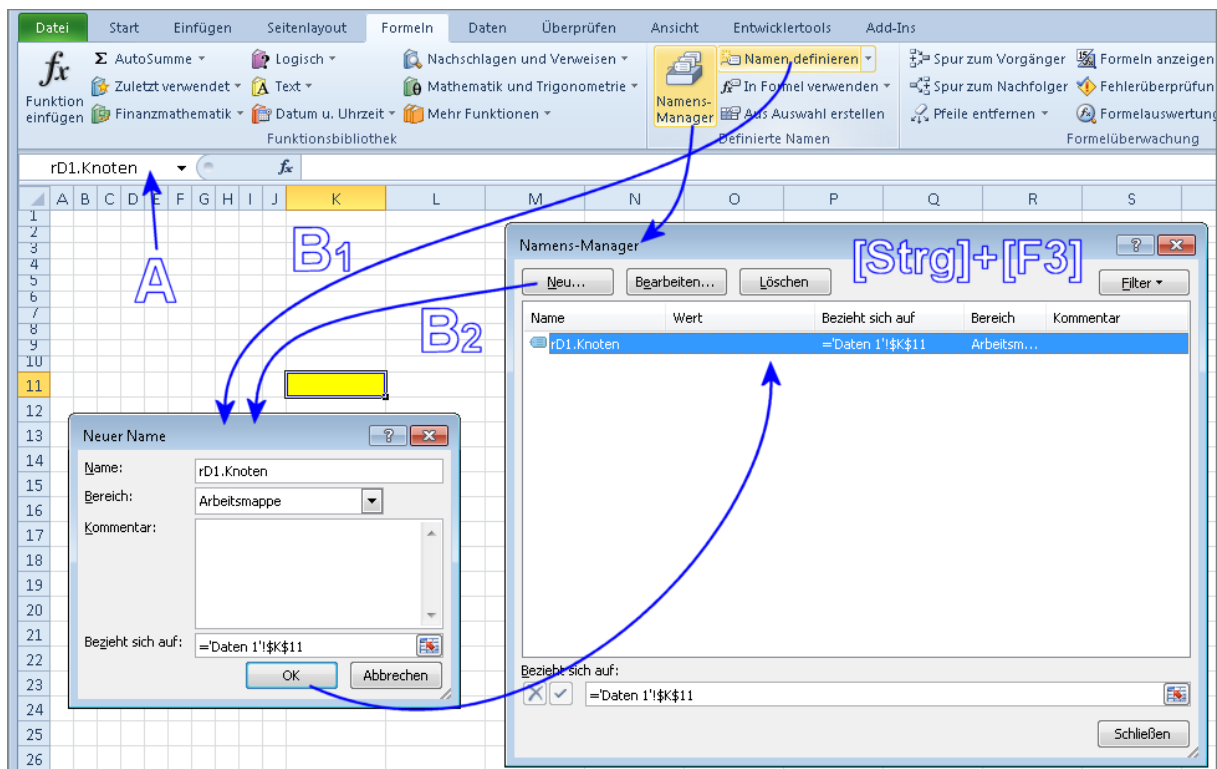


Abbildung 25: Namen festlegen für eine Zelle bzw. einen Zellbereich

- Variante C: Aktivieren Sie im Menüband die Registerkarte *Formeln* und klicken Sie in der Gruppe *Definierte Namen* auf den Eintrag *Namensmanager*. Alternativ benutzen Sie die Tastenkombination [Strg]+[F3]. Klicken Sie dann im Dialogfeld *Namensmanager* auf die Schaltfläche *Neu*. Dann weiter wie bei Variante B. Im Dialogfeld *Namensmanager* können Sie überdies auch komplexe Entscheidungen zur Verwaltung der Namen Ihrer Arbeitsmappen treffen.

Bei der Festlegung von Bereichsnamen sind die Varianten B und C aus Gründen der Flexibilität und Arbeitssicherheit der Variante A auf jeden Fall vorzuziehen.

Bei der Festlegung von Objektnamen (also z. B. zur Benennung von Steuerelementen) benutzen Sie nach dem Markieren des Objekts die Variante A (Eingabe des Namens im Namenfeld und anschließend Drücken der Taste [Eingabe])

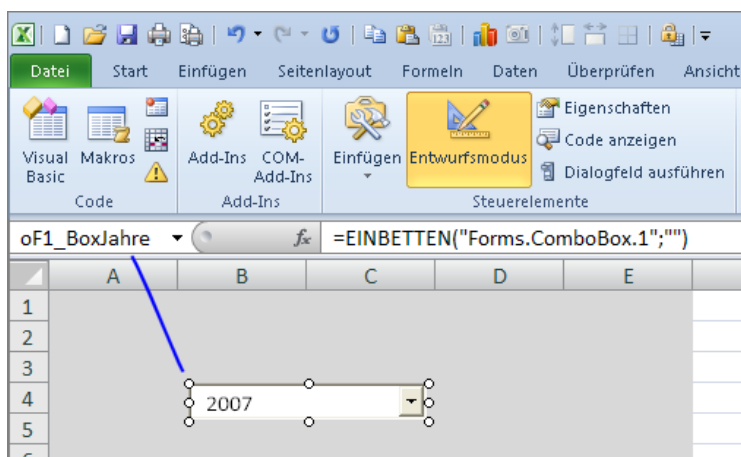


Abbildung 26: Objektnamen festlegen

### 3.4.7 Namen in Formeln einfügen

Die rS1.Methode verlangt die konsequente Verwendung von Bereichsnamen als Formelargumente. Das Einfügen solcher Namen in Formeln ist denkbar einfach und sehr komfortabel. In Zusammenhang mit Abb. 27:

- Variante A: Schreiben Sie die Formel bis zu der Stelle, an der ein Bereichsname einzufügen ist. Drücken Sie dann die Taste [F3]. Es öffnet sich ein Dialogfeld, in dem alle vergebenen Namen aufgelistet sind. Da die Anwendung der rS1.Methode mit ihren Namenspräfixen eine sehr übersichtliche Ordnung herstellt ist der benötigte Name leicht zu finden und mit Doppelklick oder mit Auswahl und abschließendem OK in die Formel einzufügen.
- Variante B: Sie kennen die Anfangsbuchstaben des benötigten Namens auswendig? Bei Anwendung der rS1.Methode ist das ziemlich sicher! Schreiben Sie die Formel bis zu der Stelle, an der ein Bereichsname einzufügen ist. Schreiben Sie die ersten Buchstaben des Namens, z. B. *rd*. Unterhalb der Bearbeitungsleiste erscheint nun eine Auswahlliste (des Features *AutoVervollständigen-Formel*), in der jetzt alle Namen aufgelistet sind, die mit *rd* anfangen. (Hätten Sie nur ein *r* eingegeben würden alle Formeln *und* alle Namen aufgelistet, die mit *R* anfangen.) Klicken Sie doppelt auf den einzufügenden Namen oder markieren Sie diesen Namen in der Liste und drücken Sie dann die Taste [Tab].

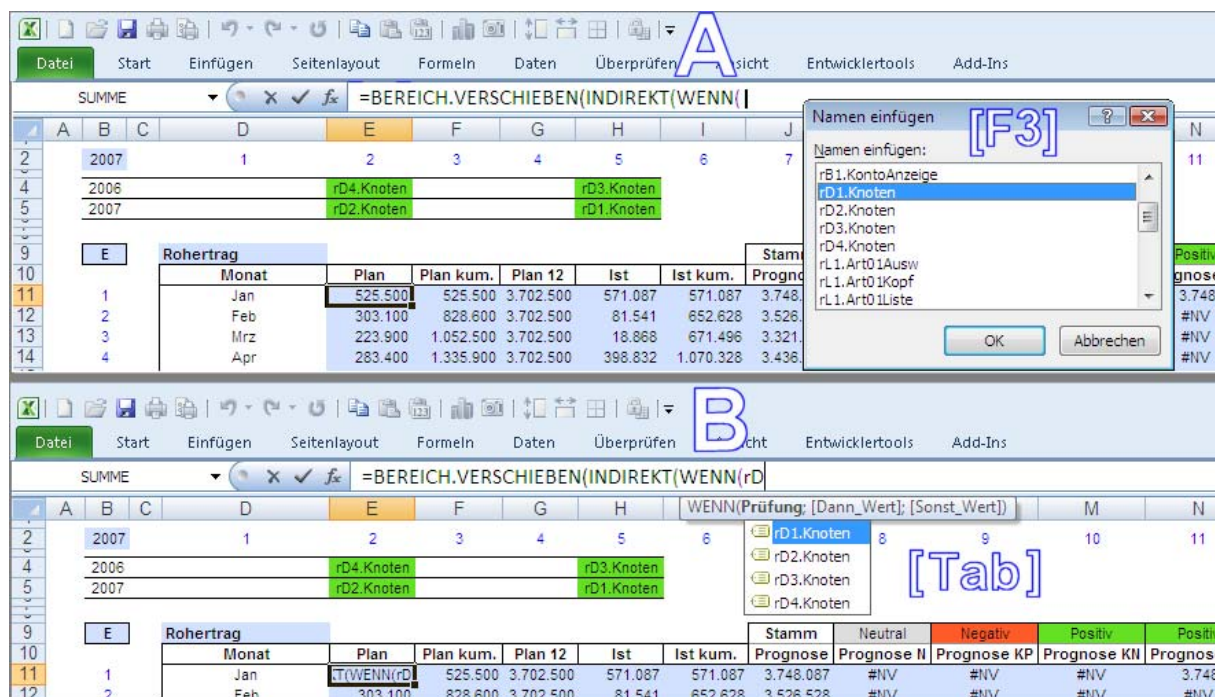


Abbildung 27: Namen in Formeln einfügen

Das Feature **AutoVervollständigen-Formel** machen Sie auf folgendem Befehlsweg verfügbar:

Excel 2010: Registerkarte *Datei/Optionen/...*

Excel 2007: *Office-Schaltfläche/Excel-Optionen/...*

dann in beiden Versionen weiter mit *Formeln/Abschnitt Arbeiten mit Formeln/Option AutoVervollständigen-Formel* aktivieren.

### 3.4.8 Benannte Bereiche ansteuern

Auch als Navigationshilfe sind die Bereichsnamen sehr hilfreich. Öffnen Sie im *Namenfeld* links neben der *Bearbeitungsleiste* die Dropdownliste der in der Mappe definierten Namen. Mit Klick auf den gewünschten Namen wird der benannte Bereich angesteuert und markiert. Da die Namen, wenn Sie nichts anderes festgelegt haben, in der gesamten Mappe gelten, wechseln Sie somit auch automatisch zum entsprechenden Arbeitsblatt.

Eine andere Methode ist die Navigation mit Unterstützung des Dialogfelds *Gehe zu*, das Sie z. B. mit [Strg]+[G] öffnen oder, schneller noch, mit der Taste [F5] zur Anzeige bringen.

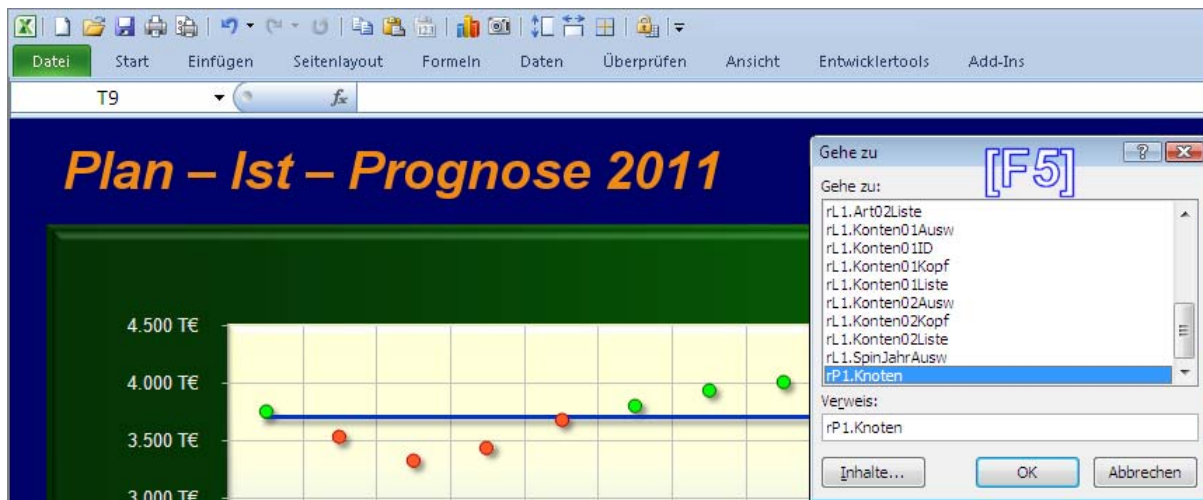
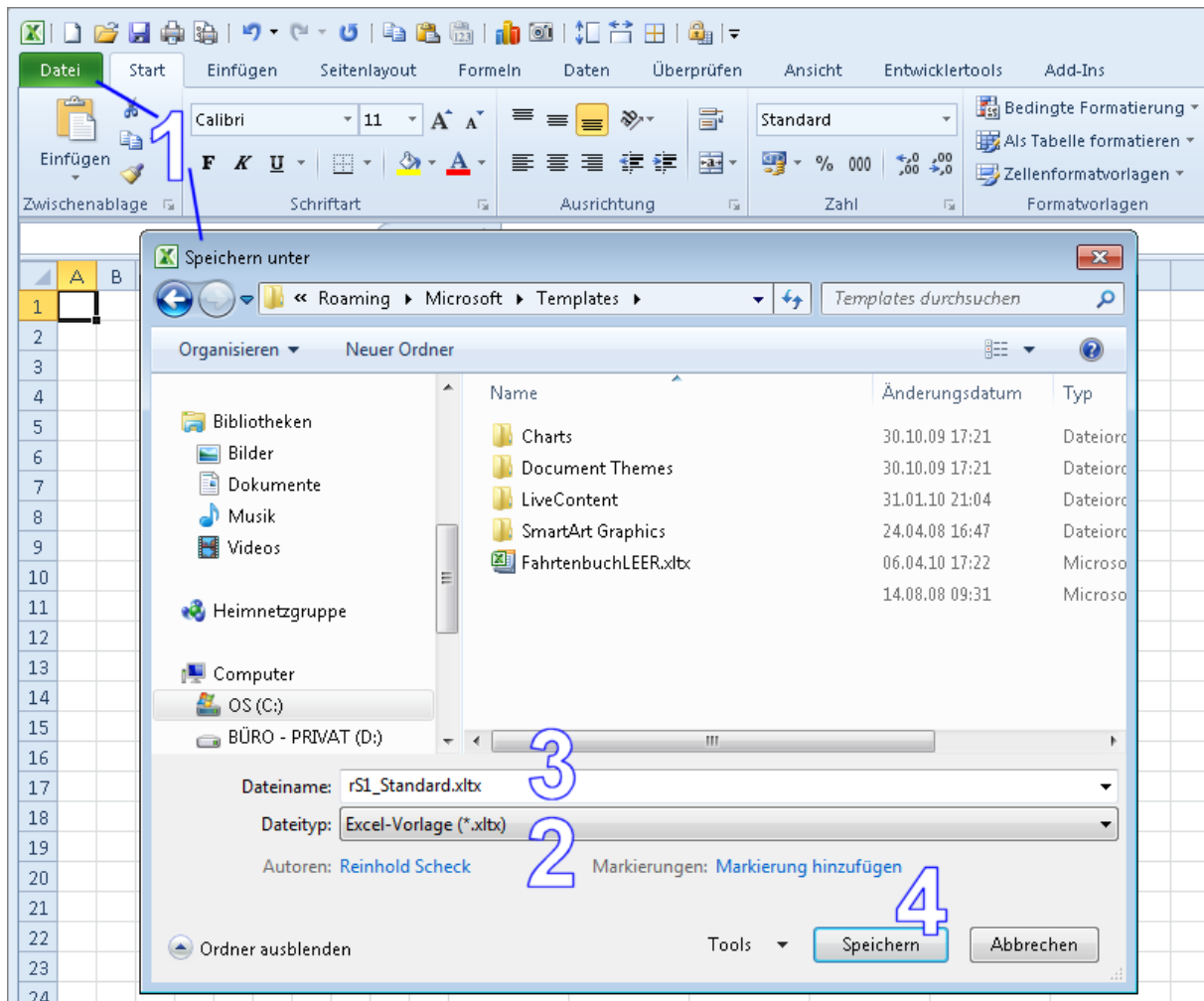


Abbildung 28: Bereiche ansteuern mit *Gehe zu* (Taste [F5])

Auch in der Auflistung dieses Dialogfelds finden Sie alle Bereichsnamen und kommen deswegen mit *OK* oder mit Doppelklick schnell zum gewünschten Ziel.

## 3.5 Excel-Vorlagen

Eine standardisierte Vorgehensweise, wie Sie die *rS1.Methode* verlangt, ist natürlich umso effizienter, wenn Sie auf bereits entsprechend vorbereitete Arbeitsmappen zugreifen können (vgl. Abschnitt 3.1.2). Die benutzerdefiniert einrichtbare *Excel-Vorlage* (in den früheren Versionen wurde sie *Muster-vorlage* genannt) ist eine spezifische Excel-Speicherform und entspricht nach ihrer Art und Verwendung der *Dokumentvorlage* in Word. Achten Sie bei der Verwendung solcher Vorlagen auf Gestaltungsfreiheit. Definieren Sie das „Große und Ganze“, vermeiden Sie aber spezielle strukturelle Vorgaben. Denn mit einem Zuviel an Festlegungen werden Sie oft Ihre Arbeit eher einschränken als erleichtern – es sei denn, Sie erzeugen eine ganz spezielle Vorlage für einen ganz speziellen, häufig wiederkehrenden Zweck. Ansonsten aber sollte Ihre Excel-Vorlage normalerweise nur wenige oder keine Funktionselemente enthalten, wohl aber sämtliche Basisstrukturen, die z. B. eine Entwicklung unter Anwendung der *rS1.Methode* erleichtern (benannte Arbeitsblätter, Spaltenbreiten und Zeilenhöhen der Hilfsspalten und Hilfszeilen, Standardformatierungen im Blatt *Listen 1*). Bei der *Excel-Vorlage* handelt es sich also um eine jederzeit und mühelos reproduzierbare Arbeitsmappe, die den immer wieder gleichartigen Aufbau eines standardisierten Grundgerüsts erspart.



**Abbildung 29: Datei als Mustervorlage speichern.**

So – in Zusammenhang mit Abb. 29 – erstellen Sie, eine *Excel-Vorlage*:

- 1 Die entsprechend vorbereitete und eingerichtete Standard-Exceldatei (Dateityp *xlsx*) ist geöffnet. Nach dem Aktivieren der Registerkarte *Datei* wählen Sie den Befehl *Speichern unter*.
- 2 Stellen Sie im Dialogfeld als Dateityp die *Excel-Vorlage (\*.xltx)* ein.
- 3 Geben Sie bei *Dateiname* den gewünschten Namen ein.
- 4 Nach dem Klick auf *Speichern* wird die Datei als Vorlage (Dateityp *xltx*) abgelegt und ist reproduzierbar.

Möchten Sie nun ein Projekt beginnen, für das Sie diese *Excel-Vorlage* benötigen, sind folgende Schritte erforderlich:

- 1 Nach dem Aktivieren der Registerkarte *Datei* (Excel 2010) bzw. nach Klick auf die *Office-Schaltfläche* (Excel 2007) wählen Sie den Befehl *Neu*.
- 2 Wählen Sie im dann erscheinenden Fenster/Dialogfeld den Befehl *Meine Vorlagen*.
- 3 Danach öffnet sich das Dialogfeld *Neu*, auf dessen Registerkarte *Persönliche Vorlagen/Meine Vorlagen* die verfügbaren, benutzerdefinierten *Excel-Vorlagen* aufgelistet sind. Wählen Sie die Vorlage aus und klicken Sie dann auf *OK*. Es wird daraufhin nicht die Vorlage selbst, sondern eine Kopie davon geöffnet.
- 4 Wenn Sie nun diese Kopie der Vorlage speichern wollen, wird von Excel als Speichervariante bezüglich des Typs automatisch wieder die Standardvorgabe eingestellt, Sie speichern also routinemäßig die Kopie der *xltx*-Vorlage als eine »normale« Arbeitsmappe, als eine *xlsx*-Datei. Dieses Verfahren schützt die gespeicherte Vorlage vor versehentlichen Änderungen und Beschädigungen.