



Petra Kopp

Georg
Bartenschlager



VBA mit ^{Microsoft} Excel

FÜR KIDS



1

Von Ablaufplänen und Programmen

So, nun geht es endlich los, in diesem Kapitel zeigen wir dir, wie du Programme

- ⊙ planst,
- ⊙ programmierst,
- ⊙ abspeicherst und
- ⊙ ausführst.



1

Hier treffen wir unseren begnadeten Pizzabäcker Kurt. Seine Devise lautet: Programmieren ist wie Pizza backen, es fehlt nur der Duft.



In Kurts Backstube am Holzofen wird viel über dieses Thema diskutiert und du kannst nun dabei sein.

Wo wir gerade beim Thema wären: Wie backst du eigentlich Pizza?

Auch wenn du kein Pizza-Fan bist: Wie könnte es denn ablaufen? Betrachte es als eine Aufgabe, um dich in die Programmierung einzuarbeiten.

Pizzarezept als Ablaufplan

Dein Rezept wird vermutlich in etwa folgende Gestalt haben:

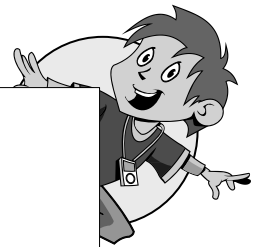
Für den Teig:

150 g Mehl
10 g Hefe
1/16 Liter warmes Wasser
1 Prise Salz
etwas Öl

Für den Belag:

Pilze,
Tomatensoße,
Gewürze wie z. B. Oregano,
Käse,
Peperoni
usw.

Pizzarezept als Ablaufplan



- Die Zutaten für den Teig verrühren wir in einer Schüssel.
- Den Teig lassen wir zugedeckt an einem warmen Ort ruhen.
- Danach kneten wir den Teig durch.
- Wir rollen daraus eine runde Teigplatte.
- Wir belegen diese mit den Zutaten.
- Anschließend wird die Pizza gebacken, bis der Teig knusprig und der Käse goldgelb verlaufen sind.

Nun überlegen wir einmal, welche Chancen eine Maschine deiner Meinung nach hätte, dieses Programm auszuführen. Stell dir vor, statt einer Maschine bekäme ein kleines Kind das Rezept vorgelesen und sollte es »umsetzen«.

Die unserer Meinung nach kritischen Punkte dieses Unterfangens sind unten aufgelistet. Unter uns: Es ist schwer an dieser Stelle eine Grenze zu ziehen, was wir einem kleinen Kind zutrauen. Was wir einer Maschine zutrauen können, werden wir später erfahren.

Damit haben wir schon etwas Wichtiges gelernt: Ein Computer kann einen Plan nur dann korrekt ausführen, wenn alle Anweisungen dieses Plans präzise in einer für den Rechner verständlichen Sprache formuliert sind. Die menschliche Umgangssprache ist für diese Anforderung zu allgemein/diffus.



Nun unsere Vorschläge, was mit diffus gemeint ist:

- ◇ Bei den Zutaten gibt es problematische Formulierungen wie »etwas Öl« oder wer weiß schon, wie viel Gramm eine »Prise Salz« sind, oder welche Vorlieben z. B. deine Gäste bei einem Belag haben, sodass »usw.« schwer umsetzbar sein wird. Genauso kann es auch mit Formulierungen wie »gehäufter Esslöffel« o. Ä. schwierig werden.
- ◇ Manchmal ist die Reihenfolge der Aktionen wichtig (wie z. B. erst rühren, dann backen).
- ◇ Manchmal ist die Reihenfolge auch egal (erst Tomaten, dann Peperoni oder umgekehrt). In der Programmierung müssen wir uns dann willkürlich für eine Reihenfolge entscheiden.
- ◇ Dann stecken noch so einige Anweisungen »zwischen den Zeilen«. Wenn wir einen Hefeteig backen, dann sollten die Zutaten Zimmertemperatur haben, weil sonst die Hefe »erschrickt«, anstatt wie gewünscht aufzugehen.



1

- ❖ *Wenn* der Teig nach längerer Zeit genug gegangen ist, *dann* können wir weitermachen, *sonst* stimmt mit dem Teig etwas nicht (z. B. Zutaten sind verdorben oder die Heizung ist defekt). Dann müssen wir das Backen verschieben. Diese Abhängigkeit mit *wenn* nennen wir *bedingte Anweisung*.

Mehr zur *bedingten Anweisung* und wo diese uns bisher schon in Excel begegnet sein könnte, findest du auf Seite 255 in Anhang A.

- ❖ »Zwischen den Zeilen« unseres Rezepts finden wir Wiederholungsanweisungen. Wir müssen mitunter mehrfach nachsehen, ob der Käse nun endlich die gewünschte Bräunung hat. *Wenn nicht*, sehen wir später noch mal nach. Mitunter müssen wir mehrfach nachsehen. Dieses Wiederholen des immer gleichen Vorgangs nennen wir in der Programmierung eine Schleife.

Erste Schritte

Nun probieren wir einmal aus, wie wir unseren schicken Ablaufplan namens »Pizzarezept« in ein Programm umsetzen können.

Wir versuchen, ein Programm zu schreiben, das uns Schritt für Schritt anweist, was beim Pizzabacken zu erledigen ist. Wenn wir die jeweilige Anweisung weggeklickt haben, dann soll die nächste am Bildschirm erscheinen.

Dazu erledigen wir folgende Schritte:

- Wir starten Excel.
- Wir wechseln in die Entwicklungsumgebung, in der wir programmieren können.
- Wir legen ein »Blatt« an, auf dem wir unser Programm schreiben können.
- Wir schreiben unser Programm.
- Wir führen das Programm aus, um uns an den Bildschirmhinweisen zu erfreuen und sie in der Küche anzuwenden.

Und nun die Umsetzung:



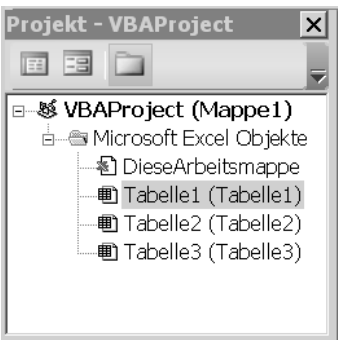
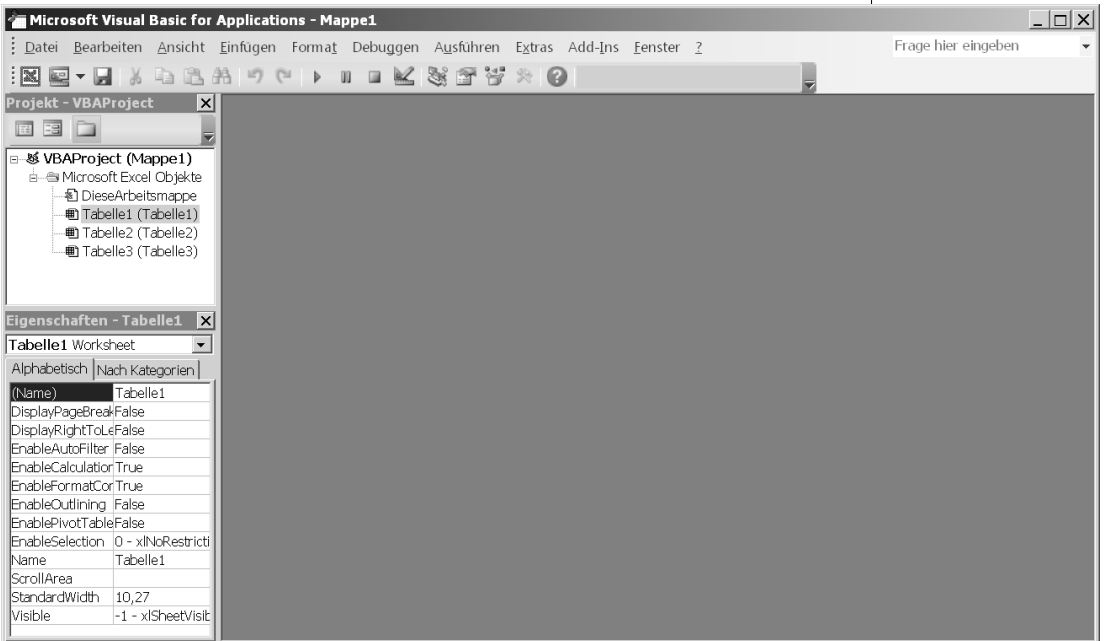
Excel starten

Excel starten wir z. B. über den Windows-Startknopf links unten am Bildschirm. In der Liste der Programme suchen wir uns das Office-Produkt Excel heraus.

In die Entwicklungsumgebung wechseln

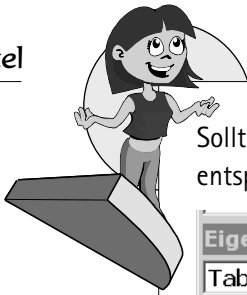
Mit der Tastenkombination **Alt + F11** wechseln wir von Excel aus in den *Visual Basic-Editor*. Das ist die Entwicklungsumgebung für Makros. Wer eine Alternative zu der Tastenkombination sucht, kann auf Seite 259 in Anhang A nachsehen.

Die Entwicklungsumgebung sollte bei dir etwa folgendermaßen aussehen:

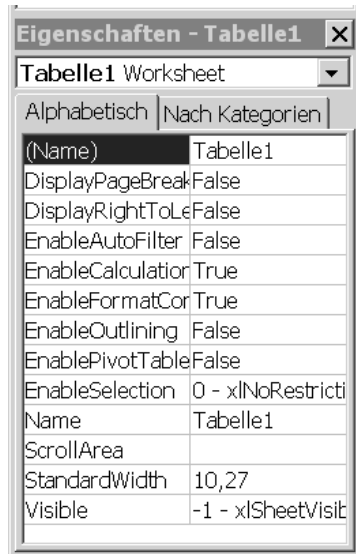


Projekt-Explorer

1



Sollte sich der *Projekt-Explorer* verabschiedet haben, kannst du ihn mit der entsprechenden Schaltfläche in der Symbolleiste wieder »hervorzubern«.



Eigenschaften



Links unten siehst du das *Eigenschaften-Fenster*. Sollte dieses nicht vorhanden sein, kannst du es mit dem Knopf »Hand und Karte« zum Vorschein bringen.



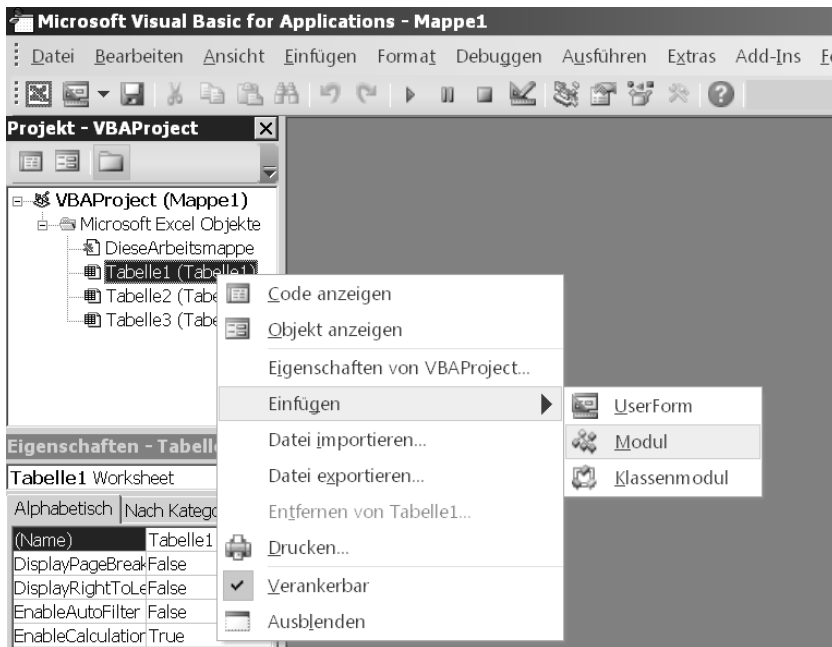
Die linke Schaltfläche öffnet den *Projekt-Explorer*, die rechte das *Eigenschaften-Fenster*

Die graue, »gähnende Leere« rechts zeigt uns, dass wir erst noch eine Schreibmöglichkeit herbeischaffen müssen, um unser Programm tippen zu können.

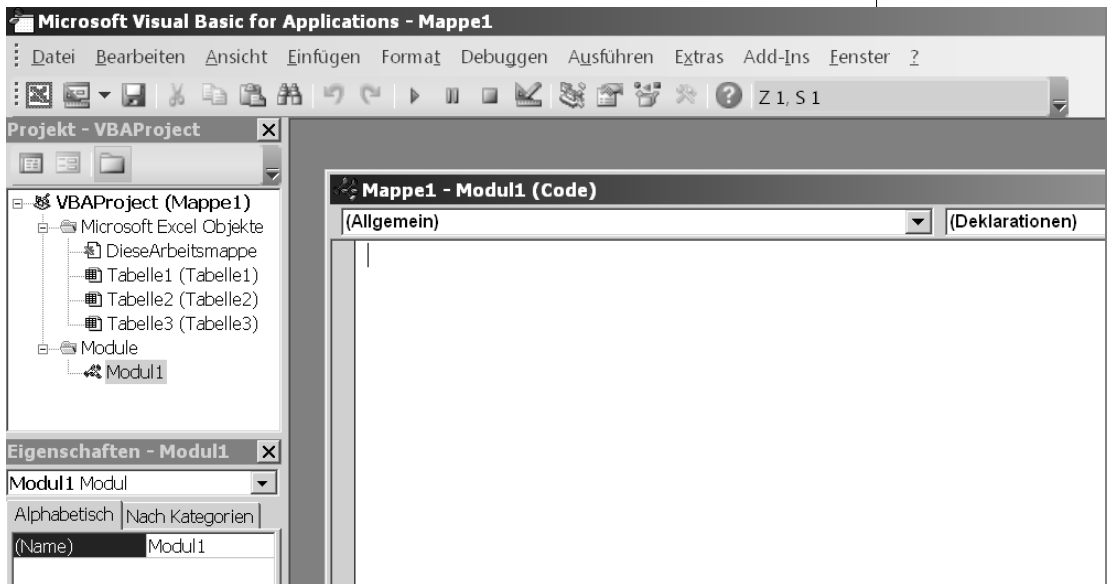
Wir holen uns ein »Blatt«, um unser Programm zu schreiben



Meistens nennen wir unsere »Schreibblätter« *Module*. Diese Module unterscheiden wir aufgrund ihres Namens. Klicke doch einmal mit der rechten Maustaste (für Linkshänder umgekehrt, je nach Einstellung der Maus) in den *Projekt-Explorer*, z. B. auf *Tabelle1*, und wähle im Kontextmenü **EINFÜGEN|MODUL**. Dann erkennst du, dass sich im *Projekt-Explorer* ein `Modul1` hinzugesellt hat.



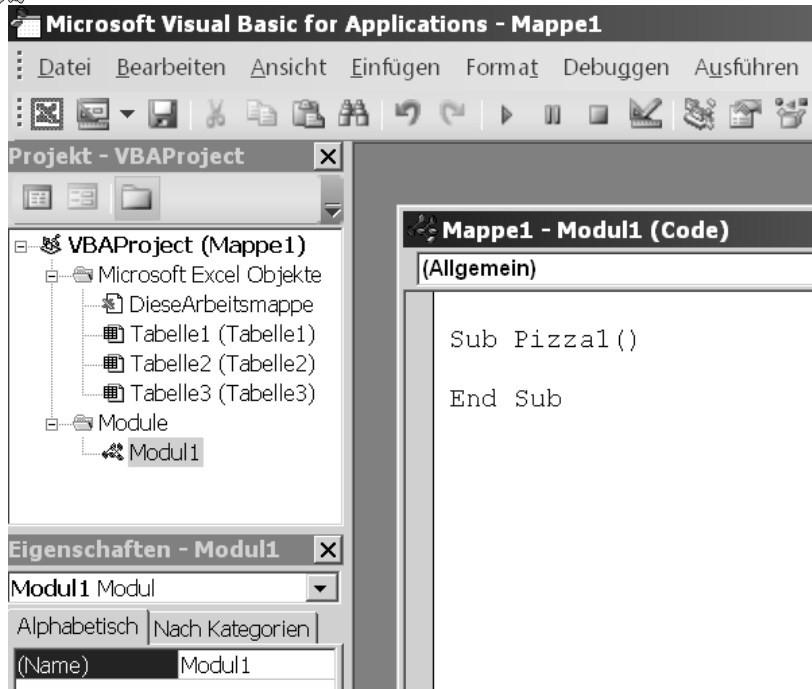
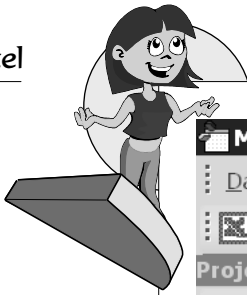
Modul einfügen



Das neue Schreibfenster

In diesem Fenster schreiben wir unser erstes *VBA*-Programm (*VBA* steht für »Visual Basic for Applications«). Es beginnt mit dem kleinen Wort *Sub*, gefolgt von einem Namen bzw. Bezeichner, den wir selbst auswählen, sowie einer öffnenden und einer schließenden runden Klammer.

1



Unser erster Programmierversuch

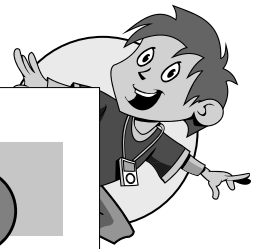


Beachte nun, dass der Editor (gemeint ist dieses neue Fenster mit dem Programmtext) dir beim Tippen hilft.

Es genügt, wenn du `sub Pizza1` tippst und dann die »Bestätigungstaste« (oder auch *Enter* oder *Return* genannt) drückst. Der Editor übernimmt dann das Einfärben der Zeichen, das Ergänzen der Klammern und das Einfügen von `End Sub` sowie das Großschreiben von `Sub`.

Die blau geschriebenen Wörter verraten uns, dass das etwas ist, was der Rechner kennt. Die Bezeichnung `Pizza1` für unser Programm ist selbst ausgedacht (eine »Eigenkreation«). Diese kennt der Rechner nicht, deshalb bleibt sie schwarz. Für diese Eigenkreationen gibt es auch gewisse Regeln, nämlich dass wir einen solchen Namen einmal vergeben müssen, sonst wäre das Programm fehlerhaft, zum anderen gelten gewisse Regeln für diese Bezeichner.

- ◇ Das erste Zeichen ist ein Buchstabe.
- ◇ Danach können nach Belieben Buchstaben, Ziffern und der Unterstrich »_« folgen.



Es muss alles ohne Leerzeichen zusammengeschrieben sein.
Pizza 1 wäre demnach falsch.



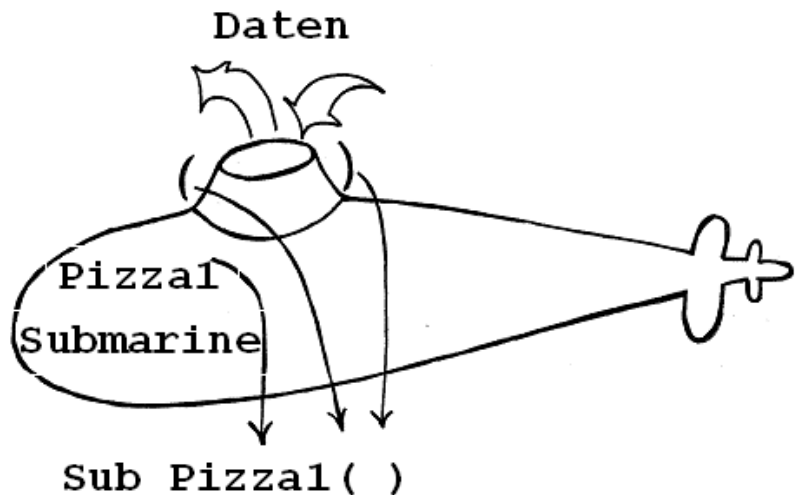
Wir programmieren unser Pizzarezept

Sub steht dabei als Abkürzung für *Subroutine*, übersetzt *Unterprogramm* (sub = unter). Unser Unterprogramm heißt `Pizza1`.

Als Gedächtnisstütze hat sich die Vorstellung eines U-Boots (engl. *Submarine* für *Unterseeboot*) bewährt. Das U-Boot hat den seltsamen Namen `Pizza1`. Die Klammern umschließen den Ausstieg. Damit soll eine Parallele gezogen werden:



- ◇ Durch den Ausstieg im U-Boot werden Personen und Gegenstände »ausgetauscht«.
- ◇ Beim Programm werden durch die Informationen, die in die Klammern geschrieben werden können, Daten ausgetauscht.



Den Rest (End Sub) erledigt Visual Basic für uns, wenn wir die Zeile umbrechen, wie wir bereits gesehen haben.

Nun kommen wir zu den Befehlen, die unser Programm abarbeiten soll.

Die Ausgabe am Bildschirm

Zur Ausgabe am Bildschirm benutzen wir die *Messagebox*, sozusagen die Nachrichtenbox.



1

Die MessageBox gibt den Text, der in Anführungszeichen steht, in einem Rechteck am Bildschirm aus.

```
Sub Pizzala()
  MsgBox "Dieser Text wird ausgegeben."
End Sub
```

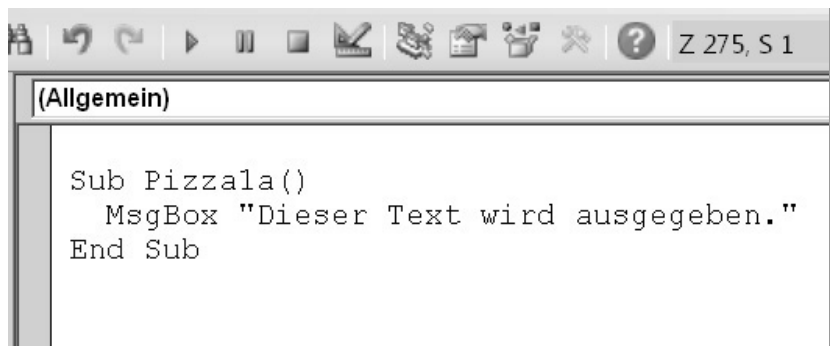
Obiges Programm führt nach einem Programmstart zu folgendem Ergebnis:



Mit dieser Methode können wir das komplette Pizzarezept der Reihe nach am Bildschirm ausgeben, was wir jetzt auch machen.

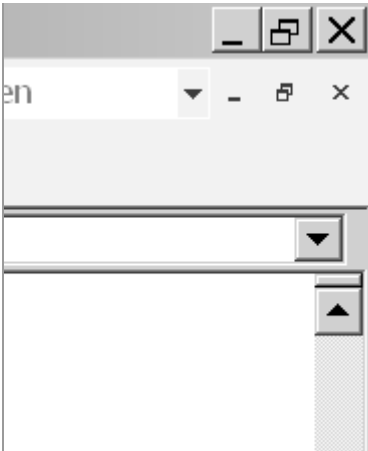
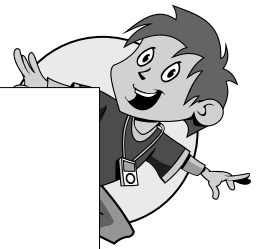
Das Programm ausführen

Gestartet wird ein Programm mit dem grünen Dreieck:



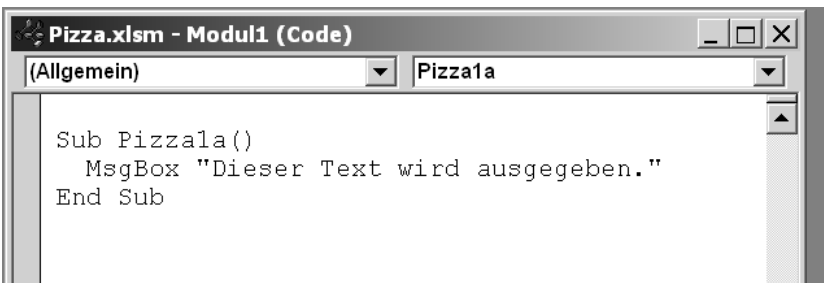
Programmstart mit dem grünen Dreieck in der Symbolleiste

Wenn das Editorfenster bereits auf Vollbild eingestellt sein sollte, kannst du es mit dem Symbol der überlappenden Blätter in der unteren Symbolleiste verkleinern.



Vollbild zurücksetzen

Wenn das Editorfenster zu klein ist, kannst du wie gewohnt das Fenster auf Vollbild stellen.

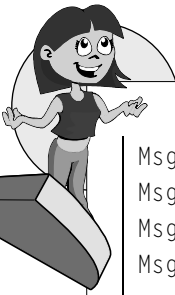


Fenster auf Vollbild stellen

Das vorläufige Programmresultat siehst du als »Listing« (so nennt man Programmausdrucke) `Pizza1` unten abgedruckt. Vorläufig deshalb, weil wir noch einige Verbesserungen vornehmen wollen.

Zum Experimentieren kannst du die Excel-Mappe `Pizza.xlsm` unter www.mitp.de/9205 aus dem Internet herunterladen.

```
Sub Pizza1()  
  
MsgBox "Für den Teig: 150 g Mehl"  
MsgBox "Für den Teig: 10 g Hefe"  
MsgBox "Für den Teig: 1/16 Liter warmes Wasser"  
MsgBox "Für den Teig: 1 Prise Salz"  
MsgBox "Für den Teig: etwas Öl"  
  
MsgBox "Für den Belag: Pilze"  
MsgBox "Für den Belag: Tomatensoße"
```



1

```

MsgBox "Für den Belag: Gewürze wie Oregano"
MsgBox "Für den Belag: Käse"
MsgBox "Für den Belag: Peperoni"
MsgBox "Für den Belag: alles Weitere nach Wunsch"

MsgBox "Die Zutaten für den Teig verrühren wir in einer ↵
Schüssel."
MsgBox "Wir lassen den Teig zugedeckt an einem warmen ↵
Ort ruhen."
MsgBox "Danach kneten wir den Teig durch und rollen ↵
daraus eine runde Teigplatte."
MsgBox "Diese Teigplatte belegen wir."
MsgBox "Wir backen die Pizza so lange, bis der Teig ↵
knusprig und der Käse goldgelb verlaufen ist."

End Sub

```

Um unser Programm zu starten, klicken wir auf das bereits erwähnte grüne Dreieck mit der Spitze nach rechts.



Das Zeichen ↵ schreibst du natürlich nicht mit. Es bedeutet, dass die Zeile an dieser Stelle eigentlich noch weitergeht. Da das Buch nicht so breit ist, müssen wir leider einen Umbruch einfügen. Du jedoch schreibst in solchen Fällen stets auf der gleichen Zeile weiter!

Nun experimentiere bitte ein wenig mit deinem Programm.

- ◇ Du kannst z. B. für den Belag einige Zutaten ergänzen.
- ◇ Versuche aufgrund des Programmtexts vorherzusagen, was als Nächstes ausgegeben wird.
- ◇ Was passiert, wenn du Zeilen austauschst? Stimmt dann noch die Logik oder darf man das so nicht machen? Welche Zeilen darfst du z. B. nicht tauschen. Denkt dabei an die Abfolge: Erst rühren, dann backen. Umgekehrt geht nicht. Bei der Zugabe der Zutaten ist die Reihenfolge eher zweitrangig, denn es wird ja doch alles verrührt. Wenn Salz und Hefe direkt aufeinandertreffen, kann Salz die Zellen der Hefe zerstören. Daher solltest du Salz im schon verrührten Teig zuletzt zugeben und dann nochmals rühren.

Wenn es bei dir nicht funktioniert, überprüfe, ob du alles richtig geschrieben hast.

Zusammenfassung



Wenn es »nur« Rechtschreibfehler im Ausgabertext unserer Messagebox sind, dann ist es aus Sicht von VBA noch in Ordnung. Der Anwender wird sich allerdings möglicherweise daran stören.

Anders ist es, wenn du Befehle falsch schreibst. Sollte sich der Tippfehler in dem Wort `MsgBox` eingeschlichen haben, dann hat VBA keine Idee mehr, wie es weitergehen soll.

Wenn du z. B. ein oder mehrere Anführungszeichen um den Text vergisst, dann wird VBA die Zeile rot färben, um dich auf den Fehler hinzuweisen.

Doch wenn du Zeile für Zeile überprüfst, wirst du den Fehler schnell finden.

Eine ausführlichere Liste der Fehler, die in solchen Programmen auftreten können, findest du auf Seite 265 in Anhang A.

Noch während du an deinem Programm arbeitest, solltest du es zur Sicherheit abspeichern.

Denk daran: Ein lauffähiges Programm ist ein wertvolles Gut.

Für das Abspeichern gibt es in deiner Entwicklungsumgebung links oben ein Diskettensymbol. Wenn du darauf klickst, wirst du gefragt, wo deine Excel-Mappe gespeichert werden soll, denn das Programm ist in der Mappe enthalten.

Wenn deine Excel-Mappe bereits gespeichert ist und du nur Ergänzungen vornimmst, dann wird der Inhalt der alten Mappe mit dem neuen Inhalt überschrieben.

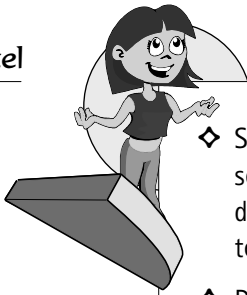
Das ist so wie immer in Excel, ob du nun Zahlen in Excel ergänzt oder Programmzeilen.

Zusammenfassung

Jetzt kannst du schon selbst ein Programm schreiben, auch wenn bislang nur wenig dabei passiert, es ist ein Programm. Deshalb schreiben wir dir die wichtigsten Punkte, mit denen du ein Programm schreibst, noch mal auf:

- ◇ Du solltest gelernt haben, was unter einem Ablaufplan zu verstehen ist, nämlich eine Folge von Arbeitsschritten (Befehlen), die in einer bestimmten Reihenfolge abgearbeitet werden sollten.
- ◇ Um ein Programm zu schreiben, wechselst du mit der Tastenkombination `Alt+F11` in den Visual Basic-Editor. Dort kannst du ein Programm schreiben, wenn ein Fenster (Editor) dafür geöffnet ist.

1



- ❖ Solltest du ein Editorfenster im Visual Basic-Editor selbst öffnen müssen, klickst du mit der rechten Maustaste auf den Projekt-Explorer. Über den Kontextmenübefehl EINFÜGEN wählst du MODUL aus und dein Fenster zum Programmieren ist offen.
- ❖ Das VBA-Programm beginnt mit der Programmzeile `Sub AusgedachterName ()`. Dabei ist `Sub` die Abkürzung für Subroutine, ein Unterprogramm.
- ❖ Die `MessageBox` gibt den Text in Klammern aus.
- ❖ Das Programm endet mit `End Sub`.
- ❖ Das Programm startest du mit dem grünen Dreieck mit der Spitze nach rechts.

Ein paar Fragen ...

Frage 1: Weißt du noch, was VBA bedeutet?

Frage 2: Was verstehst du denn unter einem Programm?

... und ein paar Aufgaben

Die Lösungen zu den Aufgaben findest du im entsprechenden Anhang.

Kapitel 1, Aufgabe 1

Schreibe ein kleines Programm `Sub Ausgabe()` unter dem soeben geschriebenen Programm `Pizza1`; ja genau in demselben Editorfenster, in dem du gerade bist.

Programmiere da die Zeile `MsgBox "Text"`.

Finde heraus, was passiert,

1. wenn du ein Anführungszeichen weglässt,
2. wenn du beide Anführungszeichen weglässt,
3. wenn du `msgbox` kleinschreibst.