

# Heiratsproblem – Programmierübung

## Programmieren mit Visual Basic in Microsoft Excel

### Einleitung

Du kennst das Heiratsproblem und den Algorithmus zum Finden einer Mann- oder Frau-optimalen stabilen Heirat. Wie wir gesehen haben, ist es sehr mühsam, diesen Algorithmus von Hand durchzuführen. Diese mühsame Arbeit soll uns nun der Computer abnehmen. Der Algorithmus soll dazu mit Visual Basic in Microsoft Excel ausprogrammiert werden.

### Vorbereitung

Bevor wir mit Programmieren loslegen können, benötigen wir natürlich die Tabellen mit den Frauen und Männern, auf welchen unser Programm operieren soll. Öffne also die Datei *heiratsproblem.xls*. Sie enthält bereits alle nötigen Tabellen.

Mache dich zuerst etwas mit den beiden Tabellen „Männer“ und „Frauen“ vertraut. Sie sind ähnlich gestaltet, wie wir es uns schon von der Gruppenarbeit gewohnt sind.

Jede Person hat eine Nummer und einen Namen. Die „Präferenzen“ stellen eine Liste mit den Nummern aller möglichen Partner des anderen Geschlechts dar. Die vorderste Nummer in dieser Präferenzliste verweist auf den Partner, den die Person am liebsten hätte. Je weiter hinten also jemand in dieser Liste steht, desto weniger wird er oder sie begehrt.

Noch leer sind die beiden Spalten „Partner“ und „probiert bis“.

In „Partner“ werden wir uns später jeweils die Nummer des Partners notieren, mit dem jemand verheiratet ist. Die „Partner“-Spalte soll also am Ende das Resultat enthalten.

Neu und noch ungewohnt ist die Spalte „probiert bis“. Dies ist eine Hilfsspalte, welche unser Programm benötigen wird. Wir müssen uns nämlich irgendwie merken, wie weit wir bei einer Person in der Präferenzliste bereits gegangen sind, welche möglichen Partner also bereits probiert wurden. Anstatt dazu die Einträge in den Präferenzen durchzustreichen, werden wir uns einfach die Nummer des zuletzt probierten Präferenzeintrages in der Spalte „probiert bis“ merken.

### Aufgabe 1 – Erste Schritte

Öffne die Visual Basic Programmierumgebung, in dem du aus dem Menü „Tools“ unter „Macro“ den „Visual Basic Editor“ auswählst.

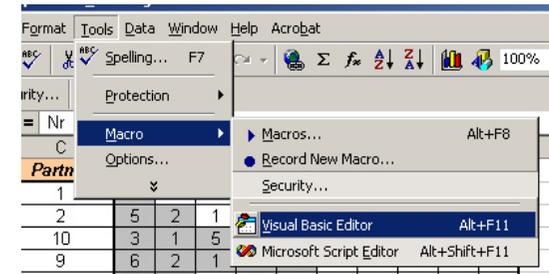


Abbildung 1: der „Visual Basic Editor“, das Tor zur Programmierung in Excel

Beginne ein neues Programm, in dem du „Module“ im Menü „Insert“ auswählst. Gib deinem Modul noch einen passenden Namen. Dazu gibst du einfach im Properties-Fenster ein beliebiges Wort als Namen ein, also zum Beispiel „Heiratsproblem“.

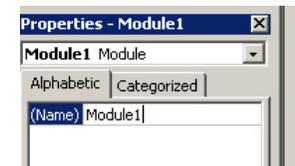


Abbildung 2: Hier kannst du den Namen des Moduls ändern.

Nun kannst du im grossen Textfenster mit Programmieren loslegen. Erzeuge dazu zuerst ein neues Unterprogramm. Platziere einfach den Cursor an einer beliebigen Stelle im Textfenster und wähle dann „Procedures ...“ aus dem Menü „Insert“.

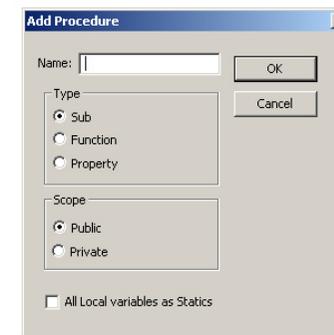


Abbildung 3: Erzeuge eine neue Prozedur.

Gib einen geeigneten Namen ein, zum Beispiel „LoeseHeiratsproblem“. Alle anderen Einstellungen lässt du so wie in Abbildung 3. Durch einen Klick auf „OK“ wird ein leeres Unterprogramm erstellt.

Der erste Teil unseres Programms besteht darin, die Spalten „probiert bis“ und „Partner“ zu initialisieren. Am Anfang haben ja alle noch keinen Partner, dies entspricht dem Wert 0. Und es wurde auch noch kein Partner ausprobiert, also ebenfalls jeweils eine 0.

Schreibe einfach mal folgendes Unterprogramm ab, welches die zwei Spalten in beiden Tabellen auf 0 setzen soll:

```
'Initialisiere():
' Setze alle "Partner" und "probiert bis" auf 0.
Public Sub Initialisiere()
    pos = 1
    While Not IsEmpty(Worksheets("Frauen").Cells(pos + 1, 1).Value)
        Worksheets("Frauen").Cells(pos + 1, 3).Value = 0
        Worksheets("Frauen").Cells(pos + 1, 4).Value = 0
        Worksheets("Männer").Cells(pos + 1, 3).Value = 0
        Worksheets("Männer").Cells(pos + 1, 4).Value = 0
        pos = pos + 1
    Wend
End Sub
```

Abbildung 4: Unterprogramm zum Initialisieren der Tabellen

Versuche das Programm auch gleich zu verstehen. Beachte, dass die Spalten 3 und 4 den Spalten „Partner“ und „probiert bis“ entsprechen. Der verwendete Befehl „IsEmpty“ überprüft ob ein Tabellenfeld leer ist.

Füge nun in die noch leere Prozedur „LoeseHeiratsproblem“ einen Aufruf von „Initialisiere“ ein. Dazu schreibst du einfach den Namen der Prozedur hinein, also einfach „Initialisiere“.

Um das Programm auszuführen, gibt es mehrere Möglichkeiten. Du kannst im Visual Basic Editor einfach den Cursor in eine Prozedur platzieren und diese mit einem Klick auf  starten. Dies ist aber viel zu mühsam, wenn wir unser Programm oft aufrufen müssen. Deshalb wollen wir uns in Excel einen Knopf basteln, welcher unser Programm aufruft, ohne dass wir dafür in den Programmcode gehen müssen.

10	9	Jennifer	8	6	10
11	10	Nancy	3	2	2
12	Löse Heiratsproblem				
13					
14					

Abbildung 5: ein Knopf in unserer Excel Tabelle

Gehe also zu den Excel Tabellen zurück um einen Knopf wie in der Abbildung 5 zu erstellen. Klicke dazu mit der rechten Maustaste auf die Menüleiste oder die Toolbar und wähle „Forms“ aus, um die Formular-Toolbar anzuzeigen.



Abbildung 6: Die Formular-Toolbar

Um einen Knopf zu erzeugen, klickst du einfach auf  in der Formular-Toolbar und anschliessend irgendwo auf das Tabellenblatt, wo du gerne den Knopf erzeugen möchtest. Es erscheint ein Fenster, in dem du die Prozedur auswählen kannst, die beim Drücken auf diesen Knopf ausgeführt werden soll. Hier wählst du natürlich die Prozedur „LoeseHeiratsproblem“ aus.

Nun kannst du noch den Text auf dem Knopf abändern, in dem du ihn anklickst und bearbeitest. Ausserdem kannst du die Grösse des Knopfes noch etwas anpassen.

Sobald dir dein Knopf gefällt wie er ist, so klickst du einfach irgendwo neben dran und kannst nun durch Anklicken des Knopfes dein Programm ausführen lassen. Es sollten nun die beiden Spalten sowohl in der „Männer“- als auch in der „Frauen“-Tabelle auf 0 gesetzt werden.

## Aufgabe 2 – Zählen

Als nächstes sollst du selber eine Funktion schreiben, welche die Männer oder die Frauen zählt. Wir benötigen nämlich diese Anzahl der möglichen Ehepaare später in unserem Programm.

Eine Funktion ist ein Unterprogramm, welches einen Wert zurückgibt. Erzeuge zuerst eine neue Funktion mit dem Befehl „Procedures ...“ aus dem Menü „Insert“ im Visual Basic Editor. Wähle diesmal „Function“ als Type aus und gib der Funktion den Namen „AnzahlPaare“.

Verwende nun in der Funktion „AnzahlPaare“ eine Variable „anz“, um die Anzahl der Männer oder Frauen zu berechnen. Am Ende schreibst du dann einfach noch den Befehl „AnzahlPaare = anz“ hin, um den berechneten Wert zurück zu geben.

```
'AnzahlPaare:
' Gibt die Anzahl der Maenner respektive Frauen,
' und somit die Anzahl der gesuchten Paare zurück.
Public Function AnzahlPaare()
    anz = 0
    ' Hier kommt dein Programmteil:
    ' Berechne die Anzahl Männer in der Variable anz!
    AnzahlPaare = anz
End Function
```

Abbildung 7: Grundgerüst für die Funktion "Anzahl"

Kleiner Tipp: Beachte die Prozedur „Initialisiere“. Ganz ähnlich funktioniert auch das Zählen.

Um die Funktion zu testen, kannst du sie direkt in einem Excel-Tabellenfeld ausprobieren. Schreibe dazu einfach in ein beliebiges leeres Tabellenfeld die Formel „=AnzahlPaare()“.

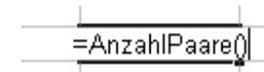


Abbildung 8: Funktionen können direkt in Tabellen verwendet werden.

### Aufgabe 3 – Die Liebe

Zum Finden einer stabilen Heirat müssen wir immer wieder entscheiden ob ein Mann eine Frau lieber hat als eine andere. Dazu definieren wir uns am einfachsten eine Funktion "Liebe", welche uns angibt wie gross die Liebe von einem Mann zu einer Frau ist.

```
'Liebe (i, j):
'   Wie stark liebt der i-te Mann die j-te Frau?
Public Function Liebe(mann, frau)

End Function
```

Abbildung 9: Grundgerüst für die Funktion „Liebe“.

Am einfachsten lässt du die Funktion ein Mal durch die Präferenzliste des Mannes gehen, bis du auf die entsprechende Frau triffst. Je weiter hinten die Frau steht, desto kleiner soll der Rückgabewert sein. Als maximale Liebe können wir hier gerade die Funktion *AnzahlPaare()* verwenden.

Teste auch diese Funktion wieder in einem Tabellenblatt. Lasse dir für alle Männer ihre Liebe zu allen Frauen in einer Tabelle ausrechnen. Das Resultat sollte ungefähr der folgenden Abbildung entsprechen.

Nr	Name	Partner	probiert bis	Präferenzen
1	Arnold	1	0	1 5 7 6 4 2 10 9 8 3 10 9 8 7 6 5 4 3 2 1
2	Eros	2	0	5 2 1 9 7 6 10 8 3 4 10 9 8 7 6 5 4 3 2 1
3	Hansruedi	10	0	3 1 5 10 8 9 7 6 2 4 10 9 8 7 6 5 4 3 2 1
4	Robbie	7	0	6 2 1 5 9 7 3 10 8 4 10 9 8 7 6 5 4 3 2 1
5	Leonardo	8	0	5 3 6 2 8 1 7 4 9 10 10 9 8 7 6 5 4 3 2 1
6	Mick	6	0	10 2 6 4 3 8 9 5 1 7 10 9 8 7 6 5 4 3 2 1
7	Rnman	4	0	7 2 5 4 10 3 6 1 8 9 10 9 8 7 6 5 4 3 2 1

Abbildung 10: Einfacher Test der Funktion "Liebe".

### Aufgabe 4 – Die Nächste bitte...

Als nächstes brauchen wir eine Funktion, welche uns jeweils sagt, welches gerade die oberste Frau ist, welche noch keinen Mann hat.

Programmiere also eine Funktion "NaechsteFrau", welche jeweils die Nummer der obersten unverheirateten Frau zurückgibt.

Hinweis:  
Falls alle Frauen verheiratet sind, soll die Funktion trotzdem abrechnen und einfach eine Nummer zurückgeben, welche grösser als die Anzahl Frauen ist. In der While-Bedingung verwenden wir dazu eine And-Verknüpfung.

### Aufgabe 5 – jetzt wird geheiratet

Nun geht es darum das Programm zum Finden einer stabilen Heirat zu implementieren. Natürlich werden wir dabei die bisher programmierten Funktionen zu Hilfe nehmen.

Programmiere also die bereits angefangene Prozedur "LoeseHeiratsproblem" gemäss folgendem Algorithmus aus.

#### Algorithmus zum Finden einer Frau-optimalen stabilen Heirat

Initialisiere die Spalten "Partner" und "probiert bis" in den Tabellen "Frauen" und "Männer" mit dem Wert 0.

Merke Dir in einer Variable *frau* welche Nummer die oberste noch unverheiratete Frau hat.

Solange eine Frau mit der Nummer *frau* existiert, tue folgendes:

- Merke dir in einer Variablen *mann* die Nummer des Mannes, welcher bei der Frau mit der Nummer *frau* zuvorderst in der Präferenzliste steht (gemäss „probiert bis“). Erhöhe anschliessend den Wert in der Spalte "probiert bis" bei der Frau um eins.
- Nun müssen wir noch herausfinden ob und mit wem der Mann bereits verheiratet ist. Merke dir deshalb die Nummer seiner Partnerin in einer Variablen *partnerin*.
- **Ist der Mann noch unverheiratet (*partnerin* = 0)?**
  - Falls ja:
    - Frau und Mann heiraten  
→ in beiden Tabellen entsprechend vermerken!
    - Setze Variable *frau* auf nächste unverheiratete Frau.
  - Falls nein:
    - **Ist *Liebe(mann, frau)* grösser als *Liebe(mann, partnerin)*?**
      - Falls ja:
        - Mann lässt sich von seiner aktuellen Ehefrau scheiden  
→ in Tabelle „Frauen“ entsprechend korrigieren!
        - Frau und Mann können anschliessend heiraten  
→ in beiden Tabellen entsprechend notieren!
        - Setze Variable *frau* auf soeben geschiedene Frau (= *partnerin*).
      - Falls nein:
        - Sie können nicht heiraten, wir müssen gar nichts tun.
        - Mache mit der selben Frau weiter.

## Aufgabe 6 – Wer heiratet mit wem?

Wenn dein Programm funktioniert, so wird es nun zwar wunderbar die "Partner"-Spalten mit Zahlen füllen. Es ist aber nicht ganz einfach, daraus die Ehepaare abzulesen.

Ergänze deshalb das Programm noch so, dass am Ende alle Ehepaare mit Namen in die Tabelle "Lösung" geschrieben werden.

	A	B
1	Melissa	Arnold
2	Pamela	Eros
3	Britney	Hugo
4	Julia	Romeo
5	Claudette	Kevin
6	Victoria	Mick
7	Demi	Robbie
8	Amanda	Leonardo
9	Jennifer	Bruce
10	Nancy	Hansruedi
11		

Abbildung 11: So sollte schlussendlich die Lösung aussehen.

Nun kannst du noch ein paar eigene Heiratsprobleme erfinden, in dem du eigene Männer- und Frauen- Tabellen produzierst und diese dann mit deinem Programm lösen lässt.