
Vorlesung: Programmieren in Anwendungen
Übungsblatt 2

http://softech.informatik.uni-kl.de/Homepage/PIA_SS14

- Bitte bearbeiten Sie **alle** Übungen dieses Blattes. Nur die letzte Aufgabe ist optional.
- Alle Aufgaben können in der VBA-Entwicklungsumgebung für Word bzw. Excel bearbeitet werden.
- Lösungen können häufig im Direktfenster getestet werden.

Daten aus Excel in Word integrieren

Gegeben sei eine Excel-Datei, die Daten von Teilnehmern einer Klausur enthält, in dem folgenden Format enthält:

- Die erste Spalte enthält die Nachnamen der Teilnehmer.
- Die zweite Spalte enthält die Vornamen der Teilnehmer.
- Die dritte Spalte enthält die erreichte Punktzahl des jeweiligen Teilnehmers in der Klausur.

Erstellen Sie ein Word-Dokument, das eine Liste der Teilnehmer enthält, die die Klausur bestanden haben. Bestanden gilt die Klausur, wenn mehr als 40 Punkte erreicht wurden.

1. Öffnen Sie zuerst ein Excel-Dokument und füllen Sie es mit einigen Testdatensätzen. Speichern Sie das Dokument unter dem Namen *Klausurergebnisse.xlsx*.
2. Öffnen Sie dann ein Word-Dokument und speichern Sie das Dokument unter dem Namen *Aushang.docm*.
3. Erstellen Sie eine Prozedur `Datenimport`, die die gewünschte Funktionalität implementiert.
4. Achten Sie darauf am Ende die entsprechenden Objekte wieder freizugeben.
5. Testen Sie die Prozedur.
6. Ergänzen Sie das Makro um einen Dialog, der beispielsweise die Bestehensschranke anpasst oder den Pfad der Klausurergebnisse vom Benutzer erfragt.

Buchstaben zählen

Schreiben Sie ein Makro, welches das Vorkommen eines Buchstabens in einem Satz/Wort (allgemein: in einem String) zählt.

Designen Sie dazu eine UserForm mit den folgenden Komponenten:

- Textfeld zur Eingabe eines Strings
- Textfeld zur Eingabe eines Buchstabens
- CheckBox zur Abfrage, ob der Buchstabe in Groß- und Kleinschreibung gesucht werden soll
- CommandButtons um das Zählen zu starten, die Eingaben zurückzusetzen und abzuberechnen
- Label zur Anzeige des Ergebnisses

Datenimport aus .csv - Datei

Schreiben Sie ein Macro, das es erlaubt, eine .csv - Datei (CSV = comma separated values) in Excel zu importieren und einem bereits bestehenden Tabellenblatt hinzuzufügen. Dabei sollen duplizierte Datenreihen entfernt werden, d.h. Datenreihen, bei denen die Einträge aller Spalten gleich sind.

Beispiel: Nach dem Entfernen der Duplikate soll aus der Tabelle:

A	4	5
B	4	5
B	2	5
B	4	5

folgende Tabelle übrigbleiben:

A	4	5
B	4	5
B	2	5

- Verwenden Sie einen geeigneten FileDialog, um die Datei zu auswählen.
- Verwenden Sie die Funktion `Range.RemoveDuplicates`, um die Duplikate zu entfernen. Sie können dabei davon ausgehen, dass die Datensätze jeweils 3 Spalten umfassen.
- Auf der Homepage finden Sie eine .csv-Beispieldatei. Laden Sie diese bitte herunter und testen Sie Ihren Code damit.

Diagrammimport aus Excel

Erweitern Sie das Beispiel aus der Vorlesungen zu Diagrammen in Word um einen UserForm-Dialog, der es erlaubt, zwischen verschiedenen Diagrammtypen zu wählen (z.B. Kreisdiagramm und Säulendiagramm).

Kontaktdaten importieren

Erstellen Sie einen UserForm-Dialog, der es erlaubt aus den Kontaktdaten in Outlook einzelne Adressen auszuwählen und diese in einem Word-Dokument automatisch einzufügen.

Game of Life (optional)

Allgemeine Informationen zu Game of Life (Quelle: Wikipedia)

Conway's Game of Life ist ein vom Mathematiker John Horton Conway 1970 entworfenes System, basierend auf einem zweidimensionalen zellulären Automaten. Das Spiel kann manuell auf einem Stück Papier oder mit Computerhilfe simuliert werden.

Das Spielfeld ist in Zeilen und Spalten unterteilt und im Idealfall unendlich groß. Jedes Gitterquadrat ist ein Zellulärer Automat (Zelle), der einen von zwei Zuständen einnehmen kann, welche oft als lebendig und tot bezeichnet werden. Zunächst wird eine Anfangsgeneration von lebenden Zellen auf dem Spielfeld platziert. Jede lebende oder tote Zelle hat auf diesem Spielfeld genau acht Nachbarzellen, die berücksichtigt werden. Die nächste Generation ergibt sich durch die Befolgung einfacher Regeln. Da ein reales Spielfeld immer einen Rand hat, muss das Verhalten dort festgelegt werden. Eine Möglichkeit ist ein Torus-förmiges Spielfeld, bei dem alles, was das Spielfeld nach unten verlässt, oben wieder herauskommt und umgekehrt, und alles, was das Spielfeld nach links verlässt, rechts wieder eintritt und umgekehrt.

Die Folgegeneration wird für alle Zellen gleichzeitig berechnet und ersetzt die aktuelle Generation. Der Zustand einer Zelle, lebendig oder tot, in der Folgegeneration hängt nur vom Zustand der acht Nachbarzellen dieser Zelle in der aktuellen Generation ab.

Regeln

- Eine tote Zelle mit genau drei lebenden Nachbarn wird in der Folgegeneration neu geboren.
- Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit.

- Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt in der Folgegeneration lebend.
- Lebende Zellen mit mehr als drei lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung.

Mit diesen vier einfachen Regeln entsteht aus bestimmten Anfangsmustern im Laufe des Spiels eine Vielfalt komplexer Strukturen. Einige bleiben unverändert, andere oszillieren und wieder andere wachsen oder vergehen. Manche Strukturen, sogenannte Gleiter, bewegen sich auf dem Spielfeld fort.

Implementierung von Game of Life in Excel

Mit Hilfe von Excels Tabellenblättern lässt sich Game of Life relativ einfach schrittweise implementieren.

1. Definieren Sie die Grösse des Spielfeldes, z.B. 20 x 20 Zellen.
2. Definieren Sie eine Farbe für tote und einen für lebendige Zellen.
3. Erzeugen Sie zwei Tabellenblätter, für den aktuellen Zustand und den Zustand des vorangegangenen Schritts.
4. Schreiben Sie eine Prozedur `toggle`, die den Style des aktuell markierten Feldes wechselt.
5. Schreiben Sie eine Prozedur `step`, die den Zustand des Feldes im Schritt bestimmt. Definieren Sie dazu Hilfsfunktionen zur Berechnung von Nachbarfeldern (insbesondere bei Randfeldern) und zum Zustand einer Zelle im nächsten Schritt basierend auf den oben genannten Regeln.
6. Schreiben Sie eine Initialisierungsfunktion, die das Spielfeld in einen Anfangszustand versetzt und dann zwei Ereignisprozeduren registriert:

```
Application.OnKey "s", "Step"
Application.OnKey "t", "Toggle"
```

Diese Prozeduren führen dann bei entsprechender Eingabe den Farbwechsel bzw. nächsten Schritt aus.

7. Starten Sie die Initialisierungsfunktion und testen Sie Ihr Programm. Auf Wikipedia (http://de.wikipedia.org/wiki/Game_of_Life) finden sich einige interessante Startmuster.
8. Fügen Sie einen passenden UserForm-Dialog hinzu, der die Funktionalität des Programms komfortabel steuert (z.B. Wahl der Spielfeldgrösse, Farbauswahl, Anzahl der auszuführenden Schritte).