

---

## Aufgabe 2.1 (☎TELEFONVERZEICHNIS)

### *Lernziele*

- ⇒ Objektvariablen
- ⇒ Collection-Objekt
- ⇒ Objekte erstellen
- ⇒ Klassenmodul und benutzerdefinierte Objekte

### *Wiederholungsfragen*

1. Ist Collection ein spezifischer oder ein allgemeiner Objekttyp? Begründen Sie Ihre Antwort.
2. Welche der folgenden Gruppen sind geordnet, welche ungeordnet? Begründen Sie Ihre Antwort.
  - a) Personen, die vor einem Bankschalter Schlange stehen.
  - b) Eine Gruppe von Schülern und Schülerinnen auf dem Pausenhof.
  - c) Bücher in einem Bücherregal.
  - d) Die Bände eines 20-bändigen Lexikons auf einem Bücherregal.
3. Was ist richtig? Eigenschaften lassen sich durch ...
  - a) private Variablen eines Klassenmoduls beschreiben.
  - b) öffentliche Variablen eines Klassenmoduls beschreiben.
  - c) private Prozeduren eines Klassenmoduls beschreiben.
  - d) öffentliche Prozeduren eines Klassenmoduls beschreiben.
4. Was geben Sie als Objekttyp an, wenn Sie eine Objektvariable für ein benutzerdefiniertes Objekt deklarieren möchten?
  - a) Class
  - b) Collection
  - c) Den Namen eines Klassenmoduls
  - d) Den Namen des Objekts mit einem 'C' davor
5. Was tun Sie, wenn Sie in VBA eine neue Objektklassen erstellen möchten.
  - a) Sie erstellen ein Klassenmodul mit dem Projektextplorer.
  - b) Sie erstellen ein Klassenmodul mit dem Objektbrowser.
  - c) Sie verwenden das New-Schlüsselwort in einer Set-Anweisung.
  - d) Sie verwenden die Add-Methode einer Auflistung.

### Vertiefungsfragen

Das Telefonverzeichnis-Projekt soll vollständig in VBA unter Excel implementiert werden. (Die Arbeitsmappe [LTelefonverzeichnis.xls](#) enthält die lauffähige Anwendung. Machen Sie sich mit der Benutzerschnittstelle und den Funktionen der Anwendung vertraut, bevor Sie mit der Aufgabe beginnen.)

### Vorbereitungen

- Gestalten Sie die *Benutzerschnittstelle* auf einem Tabellenblatt entsprechend Bild 11.
- Erstellen Sie im Projektexplorer ein neues *Standardmodul* und benennen Sie es mit 'Prozeduren'. Es wird alle Ereignisprozeduren aufnehmen.
- Benennen Sie die Ein- und Ausgabebereiche, und weisen Sie den Schaltflächen die entsprechenden (noch leeren) *Ereignisprozeduren* `hinzufügen_Klick()`, `suchen_Klick()` und `auflisten_Klick()` im Modul Prozeduren zu.

name	telefon
Heidi	43 43 03

Hinzufügen   Suchen   Liste

anzahl
4

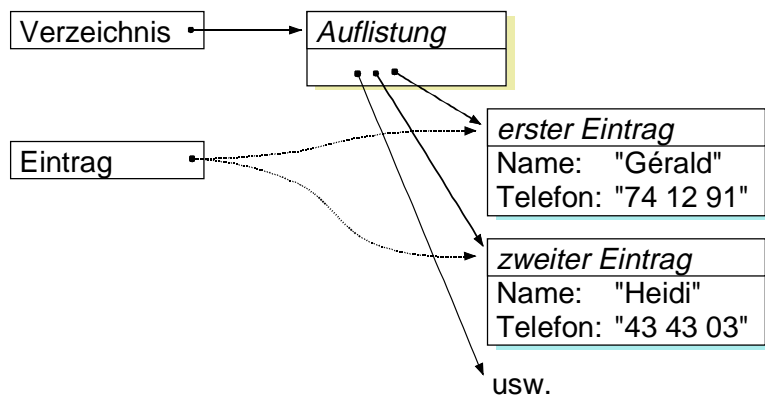
liste_name	liste_telefon
Gerald	39 72 51
Heidi	43 43 03
Peter	58 92 29
Sabine	78 98 12

**Bild 11:** Benutzerschnittstelle zu Beispiel ☞Telefonverzeichnis.xls.

Die Arbeitsmappe [☞TelefonverzeichnisSkelett.xls](#) enthält eine fertige Benutzerschnittstelle und ein Programmskelett. Bild 12 zeigt das Objektmodell der Anwendung.

Objektvariablen:

Objekte:

**Bild 12:** Objektmodell der Telefonverzeichnis-Anwendung.

a) Implementieren Sie das Verzeichnis.

- Vereinbaren Sie im Vereinbarungsabschnitt des Moduls Prozeduren eine Objektvariable Verzeichnis für das Telefonverzeichnis.

*Fragen:* Welchen Gültigkeitsbereich und welchen Datentyp geben Sie bei der Vereinbarung an? Begründen Sie Ihre Antwort.

- Implementieren Sie eine Prozedur initialisiere(), die das Verzeichnis-Objekt erstellt und der Variablen zuweist, falls das Objekt noch nicht existiert.

*Fragen:* Wie überprüfen Sie, ob ein Objekt existiert? Weshalb ist eine solche Überprüfung notwendig oder zumindest sinnvoll?

b) Implementieren Sie die benutzerdefinierte Objektklasse CEintrag für die Einträge.

- Erstellen Sie ein Klassenmodul, und benennen Sie es mit CEintrag.

*Fragen:* Weshalb sollte man ein Klassenmodul sofort sinnvoll benennen, nachdem man es erstellt hat? Ist es wirklich nötig, dem Namen ein 'C' voran zu stellen?

- Implementieren Sie die beiden Eigenschaften Name und Telefon.

*Fragen:* Welchen Gültigkeitsbereich und welchen Datentyp geben Sie bei der Vereinbarung der Variablen an? Begründen Sie Ihre Antwort.

c) Implementieren Sie die Ereignisprozedur `hinzufügen_Klick()`.

- Die Prozedur initialisiert zunächst das Verzeichnis (falls noch nicht initialisiert, siehe Aufgabe a),
- erstellt dann ein neues Eintrags-Objekt,
- liest die Eigenschaftswerte (Name und Telefon) vom Tabellenblatt ein und
- fügt das neue Objekt ins Verzeichnis.
- Schliesslich gibt sie die neue Anzahl Einträge auf dem Tabellenblatt aus.

*Hinweis: Sie benötigen eine lokale Variable. Vereinbaren Sie diese und implementieren Sie dann die fünf Punkte der Reihe nach (Sequenz von Anweisungen, vgl. 1. Teil, Abschnitt 3.1)*

d) Implementieren Sie die Ereignisprozedur `suchen_Klick()`.

- Die Prozedur initialisiert zunächst das Verzeichnis (falls noch nicht initialisiert, siehe Aufgabe a),
- durchläuft dann alle Einträge des Verzeichnisses und
- gibt die Telefonnummer aus, falls der Name des aktuellen Eintrags mit dem Namen auf dem Tabellenblatt übereinstimmt.

*Hinweis: Im Gegensatz zu Aufgabe c) lässt sich diese Aufgabe nicht mehr einfach als Sequenz von Anweisungen implementieren. Sie benötigen eine Wiederholung (vgl. 1. Teil, Abschnitt 3.1)*

*Fragen: Begründen Sie Ihre Wahl des Schleifentyps.*

*Was passiert, wenn Sie einen Namen angeben, der nicht in der Liste existiert? Was passiert, wenn ein Name mehrmals in der Liste vorkommt? Schlagen Sie sinnvolle Reaktionen auf diese Ausnahmefälle vor, und überlegen Sie sich, wie man Sie implementieren könnte.*

e) Implementieren Sie die Ereignisprozedur `auflisten_Klick()`. Sie soll vorläufig das noch unsortierte Verzeichnis ausgeben.

- Die Prozedur löscht zunächst den Ausgabebereich und initialisiert wieder das Verzeichnis.
- Danach durchläuft sie alle Einträge des Verzeichnisses und gibt Namen und Telefonnummern aus.
- Sie benötigen eine Zählvariable, um jedes Datenpaar auf einer neuen Zeile des Tabellenblatts ausgeben zu können.

*Hinweis: Benutzen Sie zum Löschen des Ausgabebereichs die `ClearContents-Methode` der `Range-Objekte`.*

- f) Implementieren Sie in der Prozedur `sortiere()` einen Sortieralgorithmus, der die Einträge im Verzeichnis alphabetisch in aufsteigender Reihenfolge sortiert.
- Den Pseudocode finden Sie unten. Arbeiten Sie mit den Positionen der Einträge in der Auflistung (Index), anstatt mit Objektverweisen.
  - Implementieren Sie zunächst die Funktion `kleinsterEintrag()`, die den Index (Position) des kleinsten Eintrags im Verzeichnis zurück gibt.
  - Implementieren Sie nun die Prozedur `sortiere()` unter Verwendung der Prozedur `kleinsterEintrag()`.
  - Binden Sie die Prozedur `sortiere()` in die Ereignisprozedur `auflisten_Klick()` ein.

*Frage:* Anstatt die Liste erst vor der Ausgabe zu sortieren, könnte man die Einträge bereits bei der Eingabe in der richtigen Reihenfolge einfüllen. Welche Änderungen müssten Sie am Programm vornehmen? Versuchen Sie diese Variante zu implementieren.

#### **sortiere durch Auswählen**

```
erstelle neues Verzeichnis (ohne Einträge)
SOLANGE noch Einträge im alten Verzeichnis
    suche kleinsten Eintrag im alten Verzeichnis
    füge Eintrag am Ende des neuen Verzeichnis ein
    entferne Eintrag aus altem Verzeichnis
ersetze altes Verzeichnis durch neues Verzeichnis.
```

#### **suche kleinsten Eintrag** im Verzeichnis

```
setze kleinster Eintrag = erster Eintrag
FÜR ALLE Einträge im Verzeichnis
    FALLS aktueller Eintrag < kleinster Eintrag
        setze kleinster Eintrag = aktueller Eintrag.
```

✓ [LTelefonverzeichnis.xls](#)

---