

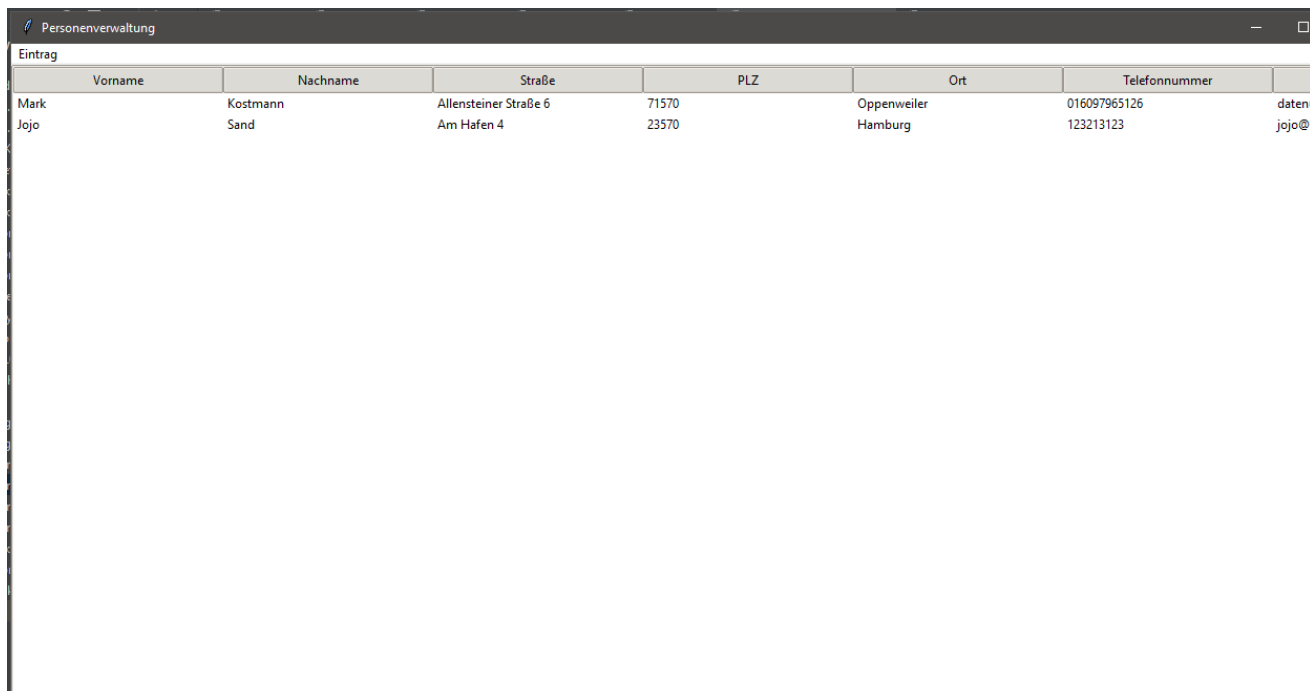
## Projekttitle: Ein Personalverwaltung

Teilnehmer: Simin Yazdi Nezhad

zusätzliche Unterstützung: Mark Kostmann

### Allgemeine Informationen:

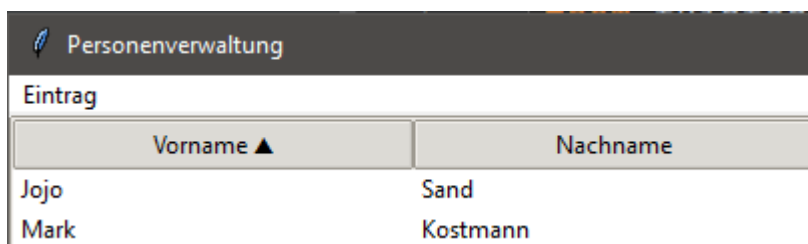
- Es wird Tabelle angezeigt, die personenbezogene Daten anzeigt.
- Im oberen Bereich, also im Tabellen-Header, werden die einzelnen Spaltennamen angezeigt.



The screenshot shows a web application window titled "Personenverwaltung". Below the title bar is a table with the following columns: Vorname, Nachname, Straße, PLZ, Ort, Telefonnummer, and a partially visible column on the right. The table contains two rows of data:

Vorname	Nachname	Straße	PLZ	Ort	Telefonnummer	
Mark	Kostmann	Allensteiner Straße 6	71570	Oppenweiler	016097965126	daten
Jojo	Sand	Am Hafen 4	23570	Hamburg	123213123	jojo@

- Wenn man auf eine Spalte klickt wird diese alphanumerisch sortiert. Klickt man noch einmal auf diese Spalte wird sie umgekehrt sortiert. Gleichzeitig erscheint in der Spalte ein Symbol, das die Sortierrichtung anzeigt.



The screenshot shows the same application window, but the table is now sorted by the 'Vorname' column. The 'Vorname' header has a small upward-pointing triangle icon next to it. The data rows are now ordered alphabetically by first name:

Vorname ▲	Nachname
Jojo	Sand
Mark	Kostmann

Vorname ▼	Nachname
Mark	Kostmann
Jojo	Sand

- Klickt man auf eine andere Spalte wird das Symbol an diese Spalte angezeigt.

Vorname	Nachname ▲
Mark	Kostmann
Jojo	Sand

- Klickt man im Menü auf Eintrag => Neu erscheint ein Popup in dem man Daten eintragen kann.

Vorname: Claudia

Nachname: Goodling

Straße: An der Pier 23

PLZ: 23569

Ort: Hamburg

Telefonnummer: 123234231

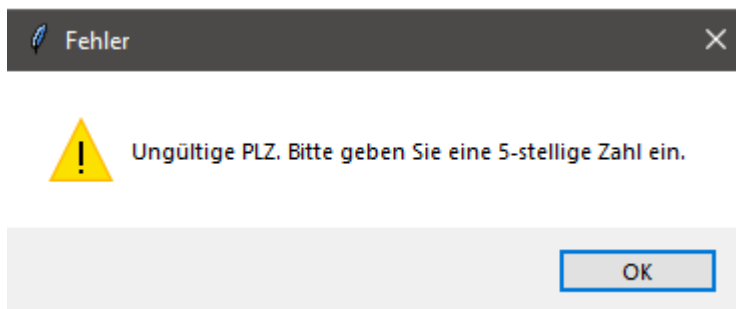
E-Mail: claudia@goodling.de

OK Abbrechen

- In den Postleitzahlen (PLZ) müssen 5 ganze Zahlen eingegeben werden. In dem Feld Telefonnummern dürfen nur ganze Zahlen und die Sonderzeichen „/“ und „-“ eingegeben werden. Im Feld E-Mail wird der Aufbau einer E-Mail Adresse rudimentär

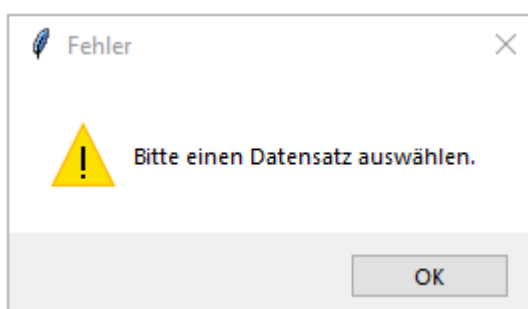
überwacht. Vor dem @ Zeichen dürfen alle möglichen Zeichen eingegeben werden. Dann muss ein @-Zeichen folgen und danach dürfen wieder beliebige Zeichen eingegeben werden. Danach muss man einen Punkt „.“ eintragen.

Gibt man in einem dieser Felder einen falschen Wert ein dann erscheint eine Fehlermeldung. In unserem Beispiel habe wir „aaa“ in s PLZ Feld eingegeben was zu einem Fehler führt:



Nachdem man das „OK“ von der Fehlermeldung geklickt hat wird das Eingabe Popup wieder angezeigt.

- Ein Doppelklick auf eine Datenzeile öffnet das Popup mit den Daten des entsprechenden Datensatzes. Somit kann man die Daten wieder bearbeiten.
- Gleichzeitig kann man aber auch eine Datenzeile markieren und im Menü Eintrag => Bearbeiten auswählen. Praktischer ist aber der Doppelklick auf eine Datenzeile, denn da wir das Popup genauso aufgerufen.
- Einen markierten Datensatz kann man löschen indem man im Menü Eintrag => Löschen auswählt.
- Wenn kein Datensatz ausgewählt wurde und man im Menü Eintrag => Bearbeiten oder Eintrag => Löschen auswählt erscheint eine Fehlerbox:



## Die Daten werden in der Datei **pspersonen.csv** gespeichert:

Jeder Datensatz besteht aus einer Zeile, die einzelnen Daten sind dort Komma separiert gespeichert. Am ende eines Datensatzes befindet sich kein Komma, :

vorn.,nachn.,xxx 6,12345,Yyy,0123,daten@datenmark.de

AA,BB,CCC 4,54321,Xxx,5678,aa@xx.yy

## Aufbau des Projekts

Wie wurde dieses Projekt aufgebaut?

Das Personenverwaltungs Tool verwendet zwei Klassen.

- Hauptfenster  
erbt von der tkinter Klasse die Tk-Klasse
- DialogNeuePerson  
erbt von der tkinter klasse die toplevel-Klasse

Die Klasse DialogNeuePerson wird in einer externen Datei gespeichert und in dem Programm eingebunden mit:

- from DialogNeuePerson import DialogNeuePerson
- **class Hauptfenster (tk.Tk)**
  - a. **\_\_init\_\_**  
zunächst wird der Konstruktor der Parent Klasse aufgerufen. Dann wird der Titel der Applikation eingetragen (hier: Personalverwaltung). Ferner werden die Höhen und Breitendaten eingetragen.  
zum Sortieren werden die Eigenschaften:
    - \* sort\_reversed und
    - \* sort\_statesgebraucht. Gleichzeitig werden Methoden aufgerufen, die zum Erstellen der Ansicht notwendig sind:
    - \* self.ladePersonen()
    - \* self.erstelleMenue()
    - \* self.erstelleTabelle()Des Weiteren werden noch diverse Funktionen aktiviert, die für die Farben des Tabellen Headers verantwortlich sind und dass ein Doppelklick auf der Tabelle möglich sein wird.
  - b. **erstelleMenue(self)**  
Diese Methode erstellt das Menü. Dies wird mit dem Package tkinter realisiert. Dort wird der Menüpunkt „Eintrag“ erstellt, der drei Submenüpunkte hatte:

- \* Neu
- \* Bearbeiten
- \* Löschen

c. **erstelleTabelle(self)**

Mittels dem Package „tk“ wird eine Tabellenstruktur erstellt. Das tk ist ein Widget aus der tkinter-Bibliothek, das eine tabellarische Ansicht darstellt.

d. **ladePersonen(self)**

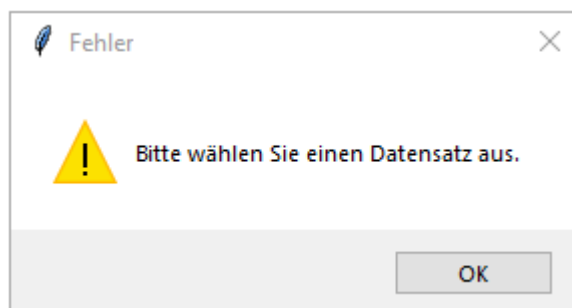
Hier werden die Personendaten aus der Datei personen.csv geladen. Die Daten werden mit dem Modul csv bearbeitet. Die Personendaten werden in einem Array (Liste) gespeichert und damit wird eine for-Loop durchgeföhren. In der for-Loop werden die Datenreihen eines jeden Datensatzes in die Tabelle eingetragen.

e. **anlegenNeuePerson(self)**

Hier wird mittels der Klasse „DialogNeuePerson“ ein Popup erzeugt.

f. **personBearbeiten(self)**

hier wird geprüft ob ein Datensatz selektiert wurde. Wenn diese Methode aufgerufen wird und kein Datensatz selektiert wurde erscheint eine Fehlermeldung.

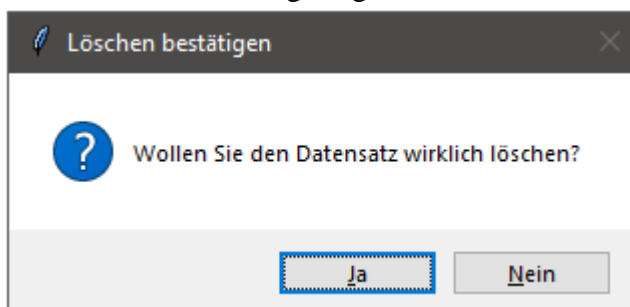


Wenn ein Datensatz ausgewählt wurde wird mit der Klasse DialogNeuePerson ein Popup erzeugt in dem die entsprechenden Daten dann in dem Popup angezeigt werden.

g. **personLoeschen(self)**

Wenn ein Datensatz ausgewählt wird kann mit dieser Methode ein Datensatz gelöscht werden. Ist kein Datensatz ausgewählt erscheint eine Fehlermeldung (s.o.).

Benutzer wird vorher gefragt, ob er den Datensatz löschen möchte.



Wenn der Benutzer dann den Ja-Button geklickt hat wird der Datensatz aus dem gespeicherten Array (Liste) entfernt. Anschließend wird die CSV-Datei

noch einmal geöffnet und der aktualisierte Datensatz aus dem Array (Liste) wird in der Datei gespeichert.

**h. beimTabellenHeaderKlick(self, col)**

Diese Methode realisiert das Sortieren der Tabellenspalte. Hier wird der Sortierzustand der Tabelle in `self.sort_states[col]` gespeichert, so dass ein weiterer Klick auf die Kopfspalte die Sortierung umkehrt.

**i. beimTabellenDoppelKlick(self, event)**

Realisiert dass bei einem Doppelklick auf einen Datensatz das Popup aufgerufen wird mit dem der Benutzer die Daten anpassen kann.

## 2. Class DialogNeuePerson

**a. \_\_init\_\_(self, elternteil, bearbeiten=False, personenDaten=None)**

Der Parameter `elternteil` repräsentiert das übergeordnete Fenster (Elternteil) des Dialogs. Wenn ein Dialogfenster erstellt wird (z. B. das Pop-up), wird das übergeordnete Fenster (das Hauptfenster) als "elternteil" für das Dialogfenster angegeben. Hier werden auch eine Art ID von den einzelnen Textfelder gespeichert, die das Popup repräsentieren.

**b. erstelleGUI(self)**

Erzeugt die GUI mit den Labels, input-Feldern, den „OK“ und „Abrechen“-Button

**c. validateTelefonNummer(self, telefonNummer)**

Hier wird die Telefonnummer mit einer Regular Expression geprüft. Telefonnummern dürfen nur ganze Zahlen enthalten. Ebenso sind die Zeichen „/“ und „-“ erlaubt.

**d. validatePLZ(self, plz)**

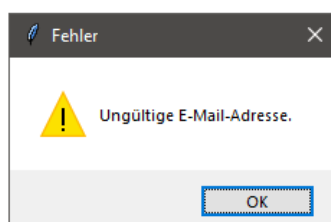
Validiert die PLZ mit einer Regular Expression. Hier müssen 5 ganze Zahlen eingegeben werden.

**e. validatEmail(self, email)**

Validiert die E-Mail mit einer Regular Expression. Hier dürfen beliebige Zeichen vor einem @-Zeichen eingegeben werden. Dann muss ein @-Zeichen eingegeben werden und danach wieder beliebige Zeichen. Zum Schluss muss ein Punkt eingegeben werden und danach wieder beliebige Zeichen.

**f. onOkButtonClick(self)**

Bei einem Klick auf den OK-Button des Popups werden die Daten, die eingegeben wurden in entsprechenden Variablen gespeichert. Dann werden die PLZ, die Telefonnummer und die E-Mail-Adresse validiert. Falls es dabei einen Fehler gibt wird eine Fehlermeldung angezeigt:



Wenn alle Daten in Ordnung sind werden die Daten in der personen.csv Datei gespeichert.