# 25 File Handling (Dateioperationen)

## 25.1 Beschreibung

- eine Datei ist eine Sammlung von Daten bzw. Datensätzen
- jeder Datensatz kann aus mehreren Komponenten bestehen
- das Sichern von Daten auf externen Speichern (Festplatte, USB-Stick, usw.) erfolgt in Form von solchen Dateien
- die Verwaltung der Dateien übernimmt das Betriebssystem
- für Dateien in der sequenziellen Organisationsform gilt:
  - die Daten werden nacheinander in die Datei geschrieben und auch nur in dieser Reihenfolge gelesen
  - einfacher Dateizugriff
  - Nachteil:
    langsamer Zugriff zum Lesen eines gesuchten Datensatzes sind erst alle davorliegenden Datensätze zu lesen
- Dateioperationen sind kritisch, Dateifehler sollten immer mit der Ausnahmebehandlung abgefangen werden try
  except
- vor dem Zugriff auf eine Datei muss die sie geöffnet werden open()
  - anzugeben sind Dateiname und Modus, in dem Datei geöffnet werden
  - Zugriffs-Modus
    - **"r"** Datei zum Lesen öffnen
      Fehler, falls die Datei nicht existiert
    - **"w"** Datei zum Schreiben öffnen
      Datei mit gleichem Namen wird überschrieben
    - **"a"** Datei zum Schreiben öffnen
      Datensätze werden hinten angefügt
- einzelnen Datensatz von einer existierenden Datei einlesen readline()
- alle Datensätze von einer existierenden Datei einlesen readlines()
- gesamte Datei einlesen read()
- schreiben einer Zeichenkette in Datei write()
  Zeilenvorschub muss explizit angegeben werden
- schreiben einer Liste von Zeichenkette in Datei writelines()
  Zeilenvorschub muss explizit angegeben werden
- Datei schließen close()
  - zu viele offene Dateien können ein Problem für das Betriebssystem sein

- Beispiel:

```
# Textdatei zeilenweise lesen
try:
    datei = open("dateiname.txt", "r")
    for zeile in datei:
        print(zeile)
    datei.close()
except FileNotFoundError:
    print("Datei nicht gefunden:")
except:
    print("sonstige Dateifehler")


# Textdatei komplett einlesen in eine Variablen vom Datentyp Liste
try:
    datei = open("zahlen.txt", "r")
    zeilen = datei.readlines()
    datei.close()
except FileNotFoundError:
    print("Datei nicht gefunden:")
except:
    print("sonstige Dateifehler")

# Variable (Datentyp Liste) in einer for-Schleife ausgeben
for zeile in zeilen:
    print(zeile, end="")

# alle Zeilen aus der Datei lesen => große Zeichenkette
datei = open("zahlen.txt", "r")
zeilen = datei.read()
for z in zeilen:
    if z.isalpha():
        print(z)
datei.close()

# in eine Textdatei schreiben
# falls die Datei schon existiert, wird sie komplett überschrieben
datei_neu = open("zahlen_neu.txt", "w")
zeile = "hallo\n"
datei_neu.write(zeile)
zeile = "fünf"
datei_neu.write(zeile)
datei_neu.close()
```

## Aufgabe

1. In einer Datei stehen zeilenweise die Länder (z.B. Italien, Frankreich, Spanien und Belgien) Diese Date ist einzulesen und auf der Konsole auszugeben

2. Für alle gelesenen Länder ist in einer Schleife die Eingabe der jeweiligen Hauptstadt über die Konsole anzufordern. In einer Variablen von Datentyp Dictionary sind Land und Hauptstadt anzulegen und dann auf der Konsole zu protokollieren.

3. Weitere Ausbaustufe: Über einen Dialog sind zwei weitere Länder mit ihren Hauptstädten zum bestehenden Dictionary zu ergänzen.
   Das Dictionary ist in einer neuen Datei auszugeben, z.B.:
   *Italien Rom*
   *Frankreich Paris*
   *Spanien Madrid*
   *Belgien Brüssel*
   usw.

4. Die Datei der Aufgabe 1 und die neue Datei, erstellt in der Aufgabe 3, sind jeweils einzulesen. Durch Abgleich der Datensätze soll ermittelt werden, welche Länder (über die Aufgabe 2) ergänzt wurden. Diese Länder sind auf der Konsole zu protokollieren