

- Im **fünften Teil des Praktikums** ergänzen wir das Programm durch eine Konfigurationsdatei im JSON-Format. Dadurch wird es möglich Eingabeparameter zu ändern, ohne das Programm anzupassen.
- Die folgenden Parameter sollen aus der JSON-Datei geladen werden:
 - destinations: Liste $[(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$ von Zielen, die angeflogen werden sollen
 - height: Flughöhe
 - protocol: Ein Wahrheitswert, der angibt, ob das Protokoll geplottet werden soll
 - routing: "xcoord", "distance from zero" oder shortest", je nachdem welche Routing-Strategie gewählt werden soll
 - view_scaling: Skalierung der View
 - speedup: Speedup-Faktor der Simulation

- Eine Konfigurationsdatei könnte folgendermaßen aussehen:

```
{  
  "destinations": [  
    [ 55.0, 20.0 ],  
    [ -116.5, 95.0 ],  
    [ -10.0, -40.0 ],  
    [ -115.0, 95.0 ]  
  ],  
  "height": 10,  
  "protocol": true,  
  "routing": "shortest",  
  "view_scaling": 2,  
  "speedup": 10  
}
```

- Der Name der Konfigurationsdatei wird als Kommandozeilenparameter übergeben. Fehlt er, sollen feste Parameter verwendet werden, z.B. Parameter für einen kurzen Testflug.
- Ob eine angegebene Datei tatsächlich existiert und ob sie eine korrekte JSON-Datei ist, muss nicht geprüft werden. Dann gibt es halt Laufzeitfehler, z.B. `FileNotFoundException` oder `JSONDecodeError`.
- Die Konfigurationsdatei wird der Einfachheit halber nur gelesen, aber nicht geschrieben.



1. Erstellen Sie in einem Texteditor eine Konfigurationsdatei ufo_config.json.
2. Erstellen Sie eine neue, zunächst leere Datei ufo_config.py.
3. Implementieren Sie in ufo_config.py die folgende Funktion:

```
def load_config(fname):
```

Parameter: str fname : Name und ggf. Pfad der Konfigurationsdatei

Rückgabewert: dict: Inhalt der Konfigurationsdatei

4. Fügen Sie in ufo_main.py einen Aufruf von load_config ein, falls ein Dateiname als Kommandozeilenparameter angegeben wurde. Anderenfalls verwenden Sie feste Parameter.
5. Ändern Sie ufo_main.py, so dass die Konfigurationswerte zur Initialisierung entsprechend verwendet werden.

- Abgabe:
 - Testen Sie das fertige Programm ausgiebig. Wenn alle Tests erfolgreich waren, verpacken Sie die fünf py-Dateien `ufo_routing.py`, `ufo_autopilot.py`, `ufo_protocol.py`, `ufo_config.py` und `ufo_main.py` und die Konfigurationsdatei `ufo_config.json` in eine zip-Datei. Laden Sie die zip-Datei anschließend in Moodle hoch. Packen Sie Datei, die sich nicht geändert haben, trotzdem mit ein.
 - Bitte laden Sie die zip-Datei rechtzeitig hoch.
- Checkliste: Überprüfen Sie vor dem Hochladen der Abgabe, ob Folgendes erfüllt ist:
 - Die Abgabe ist eine zip-Datei, die fünf py-Dateien und eine json-Datei enthält.
 - Die py-Datei `ufo_config.json` enthält lediglich eine Funktion und eine import-Anweisung.
 - Das Programm hat keine Syntaxfehler.
 - Das Programm ist mit verschiedenen Konfigurationsdateien, aber auch ohne Angabe einer Konfigurationsdatei getestet.