

Name:

Gruppenpartner:

Hier ist Platz für **Hinweise von euch!**

4. Übung

Objekte und Klassen

Hausaufgabe: Löse mindestens **eine der folgenden Aufgaben oder schreibe eine Klasse die dur dir selbst ausgedacht hast**. Ihr dürft die Aufgaben in Gruppen bearbeiten.

Abgabe im ISIS-Kurs, spätestes Abgabedatum wird dort angegeben, Gruppenpartner bitte bei der Abgabe vermerken.

Klasse Würfel

Erstelle eine Klasse `Die` (Würfel) mit einem Attribut namens `sides`, das standardmäßig Wert 6h hat. Schreibe eine Methode namens `roll_die()`, die eine Zufallszahl ausgibt, zwischen zwischen 1 und der Anzahl der Seiten, die der Würfel hat. Erstelle auch 10-seitige und eine 20-seitige Würfel.

Würfel jeden Würfel 10 mal.

Restaurant

Erstelle eine Klasse mit dem Namen `Restaurant`. Die Methode `__init__()` für ein Restaurant sollte zwei Attribute speichern: den Namen des Restaurants und die Küchenart (italienisch, indisch, ...). Erstelle eine Methode namens `describe_restaurant()`, die diese beiden Informationen ausgibt und eine Methode namens `open_restaurant()`, die aussagt, dass das Restaurant geöffnet ist.

Erstelle eine Instanz der Klasse und teste alle Funktionen.

Eine Klasse für Daten

Definiere eine Klasse `Data`, die als Datencontainer dienen soll. Sie soll eine Methode `titles()` haben, die eine Liste der Bezeichnungen der bisher gespeicherten Daten enthält, sowie eine Methode `set_data`, die einen Namen und ein Datenobjekt übergeben bekommt und dieses zu dem Namen in einem Wörterbuch speichert, sowie eine Methode `get_data`, die einen Namen übergeben bekommt und das Datenobjekt zurückgibt.

Zusatz:

Weiterhin soll es noch eine Methode `set_properties` und eine Methode `get_properties` geben, die zu einem Namen ein Dictionary von Eigenschaften speichert. Außerdem soll die Klasse Methoden `pickle` und `unpickle` haben, die einen Dateinamen übergeben bekommen und das Objekt mit Hilfe des Moduls `pickle` speichern bzw. lesen.

Wenn eine besondere Art von Daten gespeichert werden soll, kann man von dieser Klasse eine andere ableiten, z.B. `EKGData`, und dort noch spezielle Methoden zum Lesen von Daten in gewissen Formaten und zum Auswerten, Anzeigen, etc. hinzufügen. Dabei ist es wichtig, sich die Eigenschaften der entsprechenden Daten zu überlegen (z.B. bei einer Tabelle die Namen der Spalten oder der Zeilen, Einheiten, ...), um diese Eigenschaften in den Auswertungsfunktionen benutzen zu können.

Eine Klasse für Himmelskörper

Definiere eine Klasse `Body`, die als Attribute die Masse, den Radius, den Namen des Himmelskörpers enthält. Der Konstruktor sollte diese Parameter übergeben bekommen. Weitere mögliche Attribute: Bahnparameter, um die Keplerschen Bahnen zu berechnen, oder die Position und Geschwindigkeit zu einem gewissen Zeitpunkt, was ebenfalls die Berechnung der Bahnen erlaubt.

Zusatz:

Definiere eine Klasse `Viewer`, die in ihrem Konstruktor eine Liste von Himmelskörpern bekommt und Methoden `visualize` und `update`, die diese Himmelskörper 3d visualisieren, bzw. die Positionen aktualisieren. Da in galaktischen Maßstäben die Himmelskörper oft furchtbar klein sind, sollte die Methode `visualize` einen optionalen Paramter `scale` haben, mit dem man die Himmelskörper vergrößern kann. Am besten eignet sich dafür das Modul `vpython` (es heißt auch 'visual python' oder 'visual'.)

Game of Life

Programmieren Sie das Game of Life. Dafür ist es angemessen, eine Klasse `Welt` zu definieren, die den Zustand der Welt in einem Attribut speichert und eine Methode `schritt` hat, die einen Entwicklungssschritt beschreibt. Weiterhin kann die Klasse eine Methode `anzeige` haben, die mit Hilfe von matplotlib den Zustand anzeigt.