

## Übung 2: Objektorientierte Programmierung in Python

### Abgabe:

Die Übung ist als \*.zip-Datei via Moodle oder als Teil eines git-Repositories (Link via Mail) einzureichen. Das späteste Abgabedatum ist:

**Dienstag, der 13.05.2025, 23:59 Uhr**

Der Dateiname muss euren Vor- und Nachnamen enthalten:

**vorname\_nachname.zip**

### Aufgabe 01: Implementieren der Klasse "Point"

Implementiere eine Klasse mit dem Namen "Point":

- Die Objekte die über die Klasse "Point" instanziiert werden können, müssen eine x und y Koordinate haben
- Implementiere eine Methode für die Klasse "Point" zur Berechnung der euklidischen Distanz zwischen diesem und einem *anderen* Punkt ("other")

Die Formel für die euklidische Distanz lautet wie folgt:

$$\begin{aligned}dx &= x_2 - x_1 \\dy &= y_2 - y_1 \\d &= \sqrt{dx^2 + dy^2}\end{aligned}$$

- Implementiere darüber hinaus eine Methode der Klasse "Point" mit dem Namen "area", welche den Wert 0 zurück gibt, da Punkte keine Fläche haben.

### Aufgabe 02: Implementieren der Unterklasse "Geographic Point"

- Implementiere eine Unterklasse "Geographic Point" der Oberklasse "Point" (Inheritance)
- Zeige, dass die Punkte der Klasse Point vererbt werden (bspw. mit `mro()`)
- Implementiere eine Methode für die Klasse "GeographicPoint", welche die Distanz auf der Kugeloberfläche mit der folgenden Formel (s. Haversine-Formel) berechnet. Dabei steht  $\varphi$  für die geographische Breite (latitude),  $\theta$  für die geographische Länge (longitude) und  $r$  für den Radius der Erde (verwende hier  $r = 6371$ ).  $\Delta$  steht für die Differenz aus Breitengraden und Längengraden.

$$\begin{aligned}a &= \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_0 * \cos\varphi_1 * \sin^2\left(\frac{\Delta\theta}{2}\right) \\c &= 2 * \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)}) \\d &= r * c\end{aligned}$$

## Weitere Informationen:

Solltet ihr beim Lösen der Aufgaben auf Schwierigkeiten stoßen, schreibt mir eine Mail:  
gueren.dinga@hcu-hamburg.de

Ein Beispiel für die Implementierung einer Methode mit "other" könnte so aussehen. Der nachfolgende Code-Block berechnet die Altersdifferenz zwischen zwei Instanzen der Klasse "Student":

```
1 class Student:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5
6     def age_diff(self, other):
7         return self.age - other.age
8
9 std01 = Student("A", 25)
10 std02 = Student("B", 23)
11 print(std01.age_diff(std02))
```