

# Übungsaufgaben 6

## Classes - Object-oriented programming.

Abgabe bis: 21.11.2022 (Moodle)

### Coding!

#### 1

In der Vorlesung haben wir die Klasse `Point` definiert und dafür auch eine *magic method* hinzugefügt die festlegt wenn zwei Punkte als gleich gelten sollen (`__eq__()`). Programmieren Sie jetzt die unten angefangene Klasse `Circle` so weiter, dass zwei Objekte der Klasse als gleich gelten, wenn die x,y-Koordinate **und** der Radius gleich sind.

```
import math

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __eq__(self, p):
        return (self.x == p.x) and (self.y == p.y)

    def center_distance(self):
        return math.sqrt(self.x ** 2 + self.y ** 2)

class Circle(Point):
    def __init__(self, x, y, radius):
        super().__init__(x, y)
        self.radius = radius
```

Anwendungsbeispiel:

```
a = Circle(5, -3, 2.5)
b = Circle(5, -3, 2.5)
print(a == b)  # => True

a = Circle(5, -3, 2.5)
b = Circle(5, -3, 25.0)
print(a == b)  # => False
```

## 2

Wir starten mit dem folgenden Code:

```
class Person:
    def __init__(self, name, mail, password):
        self.__name = name
        self.__mail = mail
        self.__check_password(password)
        self.__password = password
        self.__failed_attempts = 0

    def __check_password(self, password):
        assert len(password) >= 8, \
            "Password must have at least 8 characters"

    def get_name(self):
        return self.__name

    def get_mail(self, password):
        # eigenen Code hierhin
```

Schreiben sie eine Methode `get_mail()` fertig, so dass die folgenden Beispiele funktionieren:

```
p1 = Person("Monica", "m.thomson@gmail.com", "SuperSecret")

p1.get_mail("FirstGuess") # => Password incorrect.
p1.get_mail("SecondGuess") # => Password incorrect.
p1.get_mail("ThirdGuess") # => Password incorrect.
p1.get_mail("FourthGuess") # => Too many failed attempts.
```

Falls aber einmal das korrekte Passwort eingegeben wurde, wird der Zähler für die Fehlversuche (`__failed_attempts`) wieder auf 0 gesetzt:

```
p1 = Person("Monica", "m.thomson@gmail.com", "SuperSecret")
p1.get_mail("FirstGuess") # => Password incorrect.
p1.get_mail("SecondGuess") # => Password incorrect.
p1.get_mail("SuperSecret") # => m.thomson@gmail.com
p1.get_mail("FirstGuess") # => Password incorrect.
p1.get_mail("SecondGuess") # => Password incorrect.
p1.get_mail("ThirdGuess") # => Password incorrect.
p1.get_mail("SuperSecret") # => Too many failed attempts.
```