2. Klassen und Objekte

Klassen

- Baupläne von Objekten
- definiert Attribute
- definiert Methoden

Beispiel: Klasse GameCharacter

Beispiel: Klasse GameCharacter

```
from dataclasses import dataclass

dataclass

class GameCharacter:
    name : str
    strength : int
    hitpoints : int
```

Objekte

- Instanz der Klassen
- wird entsprechend Bauplan erzeugt.
- Insanzierung sorgt für Initialisierung der Attribute

Instanzen erzeugen:

Instanzen erzeugen:

```
pemen = GameCharacter("Pemen", 20, 10)
talia = GameCharacter(name = "Talia", strength=12, hitpoints=10)
```

Ausgabe der Instanzen:

Ausgabe der Instanzen:

1 talia

GameCharacter(name='Talia', strength=12, hitpoints=10)

Ausgabe der Instanzen:

1 talia

GameCharacter(name='Talia', strength=12, hitpoints=10)

pemen

GameCharacter(name='Pemen', strength=20, hitpoints=10)



Methoden

- Funktionen, die auf ein Objekt selbst angwandt werden.
- innerhalb der Klassenumgebung
- Veränderung der geschützten Attribute möglich. (= Datenkapselung)
- Methoden benötigen in der Parameterliste mindestens das Attribut self.

```
from dataclasses import dataclass
   @dataclass
3
   class GameCharacter:
       name : str
5
       strength: int
       hitpoints : int
8
       def dec_hp(self):
9
           pass
10
```

Die Methode dec_hpsoll die Hitpoints des GameCharacters um die vorgegebene Zahl reduzieren.

Die Methode dec_hpsoll die Hitpoints des GameCharacters um die vorgegebene Zahl reduzieren.

```
from dataclasses import dataclass
   @dataclass
3
   class GameCharacter:
       name : str
5
       strength: int
       hitpoints : int
7
8
       def dec hp(self, damage):
9
           self.hitpoints = self.hitpoints - damage
10
```

```
# initialize two objects
pemen = GameCharacter("Pemen", 20, 10)
talia = GameCharacter(name = "Talia", strength=12, hitpoints=10)
# use the method dec_hp for the thalia object
talia.dec_hp(4)
# print the Gamecharacter objetc named talia
talia
```

GameCharacter(name='Talia', strength=12, hitpoints=6)