

## 2. Klassen und Objekte

### Klassen

- Baupläne von Objekten
- definiert Attribute
- definiert Methoden

Beispiel: Klasse GameCharacter

```
1 from dataclasses import dataclass
2 @dataclass
3
4 class GameCharacter:
5     name : str
6     strength : int
7     hitpoints : int
```

### Objekte

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

Instanzen erzeugen:

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

Ausgabe der Instanzen:

```
1 talia
```

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

```
1 pemen
```

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

## Methoden

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

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

Die Methode `dec_hp` soll die Hitpoints des `GameCharacter`s um die vorgegebene Zahl reduzieren.

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

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

```
GameCharacter(name='Talía', strength=12, hitpoints=6)
```