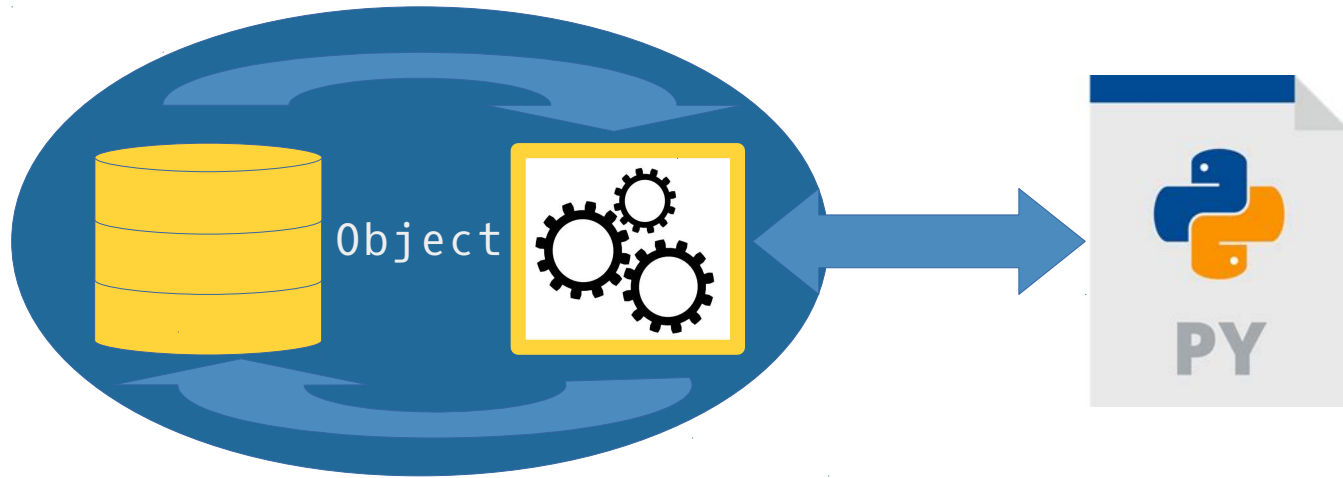


Objects



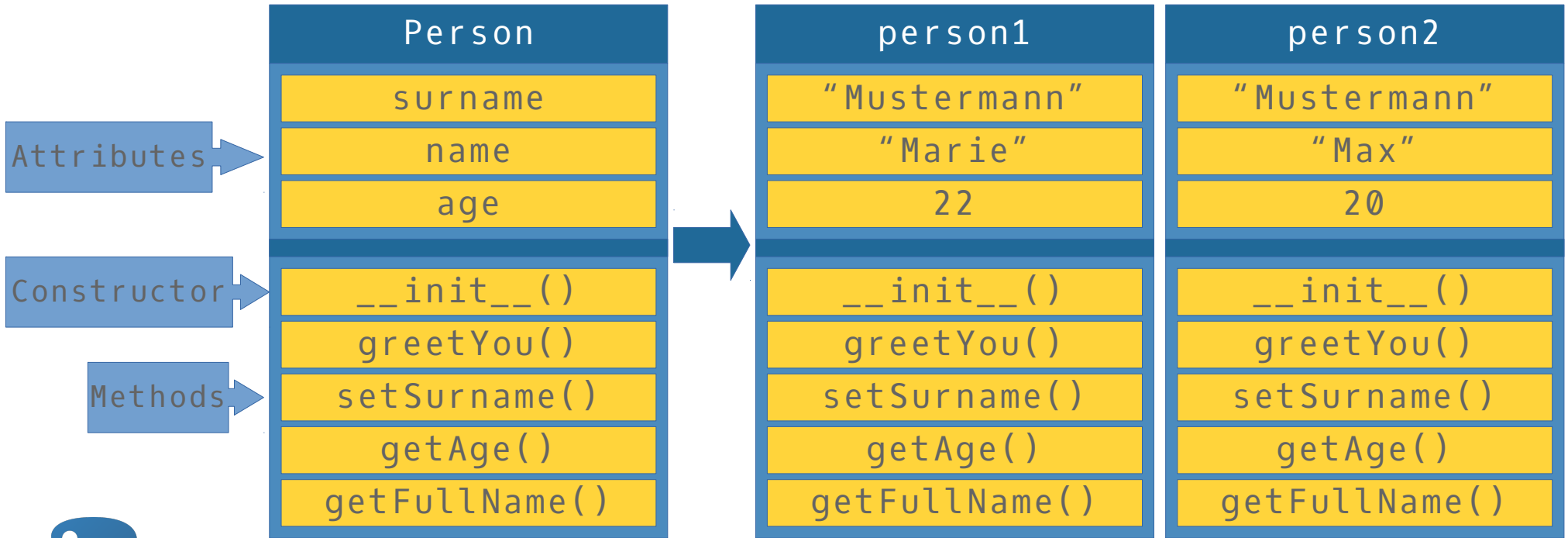
Why (not) Objects?



- + Encapsulate data
- + Reusable
- + Organise Data with their functions
- More complexity
- Data not free available



Structure of Objects



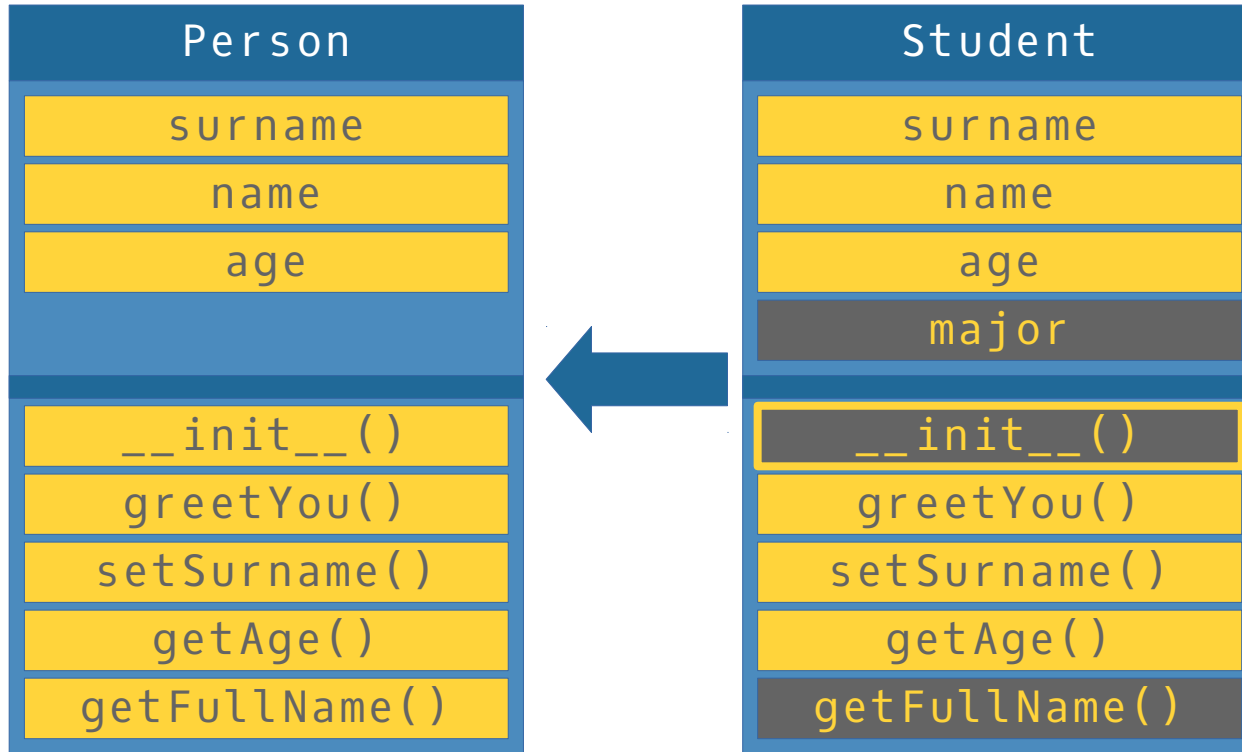
Declaring Objects/Instances

```
4 # IMPORT MODULE
5 # annotations enables you to use the class Person as a data type inside the class itself
6 from __future__ import annotations
7
8 class Person:
9     """
10     This class stores information about a person
11
12     Methods
13     -----
14     __init__(self, name, surname, age)
15     getAge()
16     getFullName()
17     setSurname(surname)
18     greet(other)
19     """
20
21     def __init__(self, name: str, surname: str, age: int):
22         """
23         Create an instance of the Person class
24
25         Parameters
26         -----
27         name : str
28             Name of the person.
29         surname : str
30             Surname of the person.
31         age : int
32             The persons age.
33         """
34         # Save the parameters of the methods as attributes of the class
35         self.name = str(name).strip()
36         self.surname = str(surname).strip()
37         self.age = int(age)
38
39     def greet(self) -> int:
40         """
41         Return the age of a person.
42
43         Returns
44         -----
45         age : int
46             The age of the person.
47         """
48         return self.age
49
```

```
50
51 def getFullName(self) -> str:
52     """
53     Return the full name of a person.
54
55     Returns
56     -----
57     str
58         The full name of the person.
59     """
60     return " ".join([self.name, self.surname])
61
62 def setSurname(self, new_surname: str):
63     """
64     Change the surname of a person.
65
66     Parameters
67     -----
68     new_surname : str
69         The new surname of the person.
70     """
71     self.surname = str(new_surname).strip()
72
73 def greetYou(self, other: Person):
74     """
75     Greet another person.
76
77     Parameters
78     -----
79     other : Person
80         An instance of the Person class.
81     """
82     print(f"Hi, {other.getFullName()}! My name is {self.getFullName()}.")
83
84 # Create instances of the class
85 person1 = Person("Marie", "Mustermann", 22)
86 person2 = Person("Max", "Mustermann", 20)
```



Inheritance



Inheritance

```
4 # Import the Person class from previous example 2
5 from example_02_object_person import Person
6
7 class Student(Person):
8     """
9     This class inherits from class Person and does stuff.
10    """
11
12    def __init__(self, name: str, surname: str, age: int, major: str):
13        """
14        Initialise class
15
16        Parameters
17        -----
18        name : str
19            Name of the person.
20        surname : str
21            Surname of the person.
22        age : int
23            The persons age.
24        major : str
25            The students major.
26
27        Returns
28        -----
29        Instance of the class Student.
30        """
31
32        # Call the constructor of the base class
33        Person.__init__(self, name, surname, age)
34        # Save the parameter as attribute
35        self.major = str(major).strip()
36
37    def getFullName(self):
38        """
39        Return the full name and major of the student.
40
41        Returns
42        -----
43        str
44            The full name and major of the student.
45
46        """
47        return f"{self.name} {self.surname} ({self.major})"
```

```
...: student1 = Student("Marie", "Müller", 22, "Mathematics")
...: student2 = Student("Max", "Mustermann", 20, "Engineering")

In [54]: print(student2.getFullName())
Max Mustermann (Engineering)

In [55]: student2.setSurname("Schmidt")

In [56]: student2.greetYou(student1)
Hi, Marie Müller (Mathematics)! My name is Max Schmidt (Engineering).

In [57]: student1.greetYou(Person("Herbert", "Schmidt", 21))
Hi, Herbert Schmidt! My name is Marie Müller (Mathematics).
```



Exercise 9: Objects

Create a class with some attributes and methods.

Create a second class inheriting from the first class.

Feel free to come up with some classes on your own or get some inspiration from my earlier examples.

