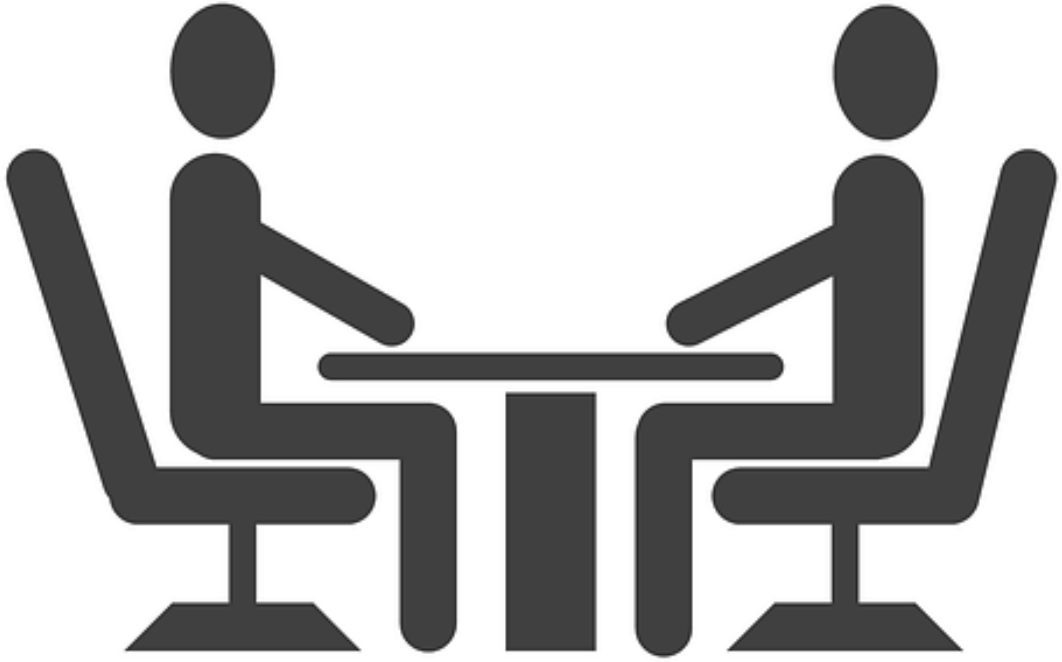
Lecture

OO Analysis

Example

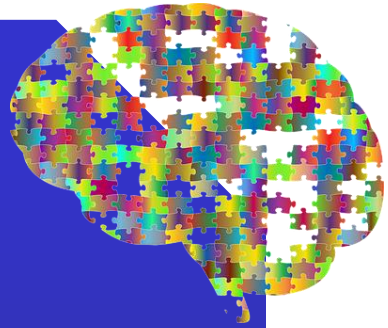
OO Analysis



OO Analysis

OO Analysis

- The shelter takes care of dogs, cats, parrots, lizards, snakes.
- There are 6 paid employees in the staff.
- The same 55 volunteers work in the shelter every week.
- 2 managers control the logistics of the shelter and the payroll.
- Donors are vital for the shelter, so the program should keep a record of them as well.



Think about...

Animal Shelter

OO Analysis

- The shelter takes care of dogs, cats, parrots, lizards, snakes.
- There are 6 paid employees in the staff.
- The same 55 volunteers work in the shelter every week.
- 2 managers control the logistics of the shelter and the payroll.
- Donors are fundamental for the shelter, so the software should keep a record of them as well.

OO Analysis

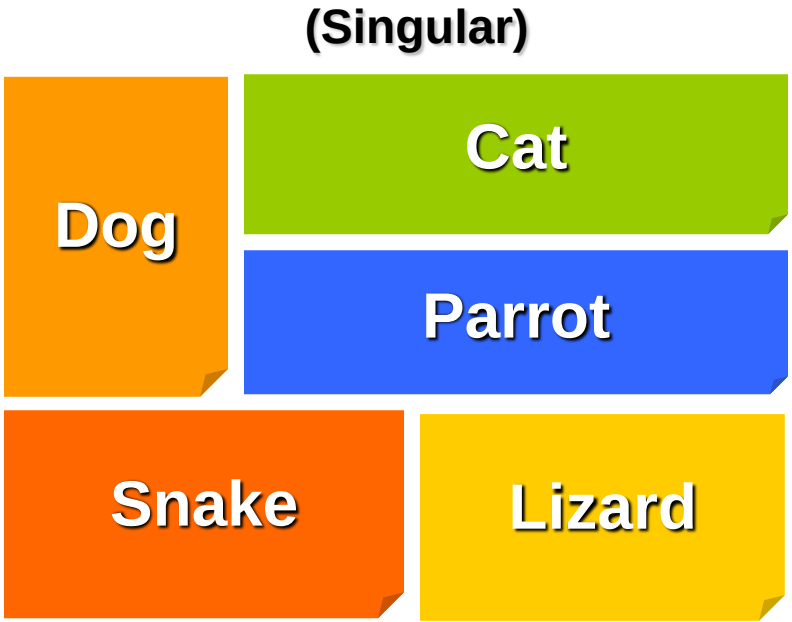
- The shelter takes care of **dogs**, **cats**, **parrots**, **lizards**, **snakes**.
- There are 6 paid **employees** in the staff.
- The same 55 **volunteers** work in the shelter every week.
- 2 **managers** control the logistics of the shelter and the payroll.
- **Donors** are fundamental for the shelter, so the software should keep a record of them as well.

OO Analysis

- **Animals:**
  - Dogs
  - Cats
  - Parrots
  - Lizards
  - Snakes
- **People:**
  - Employees
  - Volunteers
  - Managers
  - Contributors

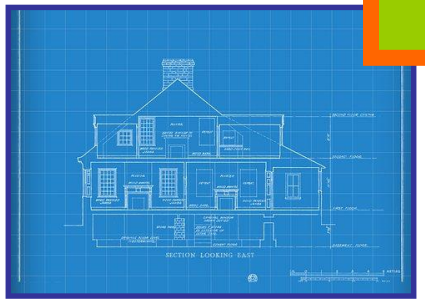
OO Analysis

- **Animals:**
  - Dogs
  - Cats
  - Parrots
  - Lizards
  - Snakes

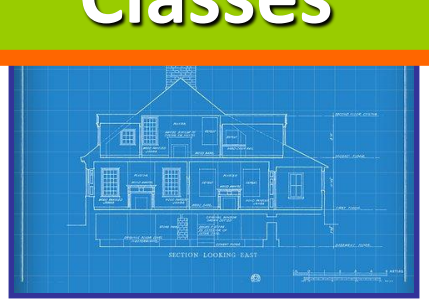


OO Analysis

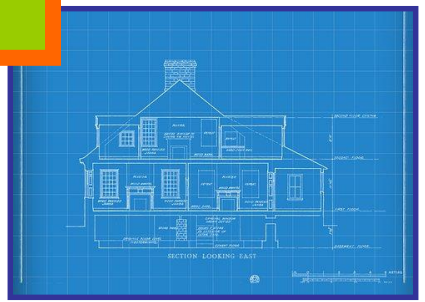
Classes



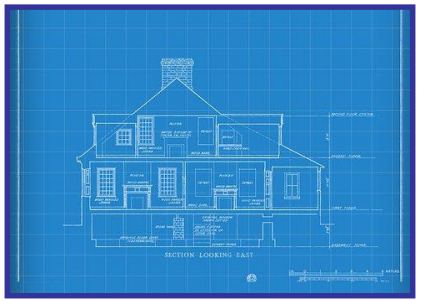
Dog



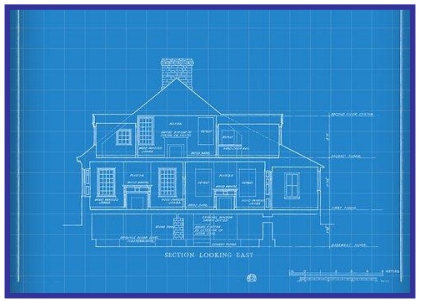
Cat



Parrot



Snake



Lizard

OO Analysis

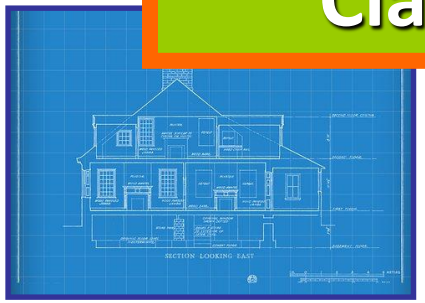
- **People:**
  - Employees
  - Volunteers
  - Managers
  - Donors

(Singular)

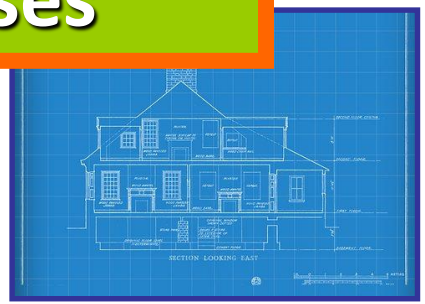
- Employee
- Volunteer
- Manager
- Donor

OO Analysis

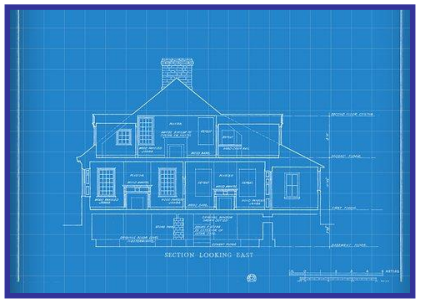
Classes



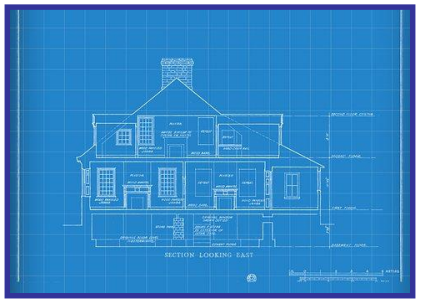
Employee



Volunteer



Manager



Donor

Class

Name

```
class Dog:

    def __init__(self, name, age, vaccines):
        self.name = name
        self.age = age
        self.vaccines = vaccines

    def display_name(self):
        print(self.name)
```

Class

```
class Dog:
    def __init__(self, name, age, vaccines):
        self.name = name
        self.age = age
        self.vaccines = vaccines

    def display_name(self):
        print(self.name)
```

Body

Class

```
class Dog:

    def __init__(self, name, age, vaccines):
        self.name = name
        self.age = age
        self.vaccines = vaccines

    def display_name(self):
        print(self.name)
```

Class

Body

```
class Dog:

    def __init__(self, name, age, vaccines):
        self.name = name
        self.age = age
        self.vaccines = vaccines

    def display_name(self):
        print(self.name)
```

Time to Practice