MyRepos\KI-Kurs-Mystro\Exercise_27062024.py

```
"""
#Exercise: Zoo Management System

Objective:
Create a simple Zoo Management System using inheritance. You will create a base class
Animal and derive specific animal classes from it.

"""

"""
1. Create a base class Animal:
This class should have the folowing attributes:
--> name: the name of the animal (string)
--> age: the age of the animal (integer)
--> species: the species of the animal (string)
This class should have the following methods:
-->   --init_-(self, name, age, species): constructor to initialize the attributes.
--> make_sound (self):a method that prints a general sound, like
-->  "Some  generic animal sound".
"""
class Animal:
  def __init__(self, name, age, species):
    self.name = name
    self.age= age
    self.species = species

  def make_sound(self):
    return("Some generic animal sound")


"""
2. Greate subclasses for specific animals:
Create at least three subclasses that inherit from Animal.
For example, Lion, Elephant, and Monkey.

Each subclass should overide the make_sound method to print a sound specific
to that animal.

For example,
--> a lion might roar,
--> an elephant might trumpet,
--> and a monkey might chatter.

Add an additional attribute or method to each subclass that is specific to that
animal. For example,
--> Lion could have an attribute mane_size and a method hunt,
--> Elephant could have an attribute trunk_length and a method swing_trunk, and
--> Monkey could have an attribute tail_length and a method swing.

"""

# 01
class Lion(Animal):
  def __init__(self, name, age, species, mane_size ):
    super().__init__(name, age, species)
    self.mane_size= mane_size

```

```
  def make_sound(self):
    return(f"{self.name} roar")

  def hunt(self):
    return(f"{self.name} hunts")

#02
class Elephant(Animal):
  def __init__(self, name, age, species, trunk_length):
    super().__init__(name, age, species)
    self.trunk_length= trunk_length

  def make_sound(self):
    return (f"{self.name} trumpet")

  def swing_trunk(self):
    return (f"{self.name}  trunks")

#03
class Monkey(Animal):
  def __init__(self, name, age, species, tail_length ):
    super().__init__(name, age, species)
    self.tail_length= tail_length

  def make_sound(self):
    return (f"{self.name}  chatter")

  def swing(self):
    return (f"{self.name}  swings")


def main():

    #  call the amimals:
    l= Lion("Borry", 10, "  Lion", "small")
    e= Elephant("Rambo", 50, " elephant", "big")
    m= Monkey("sosy", 5, " Monkey", " long")

    print(f"My Animal is {l.species} , his Name is {l.name} and his age = {l.age} years")
    print(f" {l.hunt()}  and   {l.make_sound()} " )
    print ("")
    print(f"My Animal is {e.species} , his Name is {e.name} and his age = {e.age} years")
    print(f" {e.swing_trunk()}   and {e.make_sound()}" )
    print("")
    print(f"My Animal is {m.species} , his Name is {m.name} and his age = {m.age} years")
    print(f" {m.swing()} and {m.make_sound()} " )


if __name__ == "__main__":
    main()
```