Revision

1. Create class “triangle”
   properties:
   - a
   - b
   - c

   methods:
   - per() - to calculate perimeter.
   - check() - to check whether it is possible to build a triangle.
   - area()- to calculate the area of a triangle.

1. Create class “math”:
   No properties
   methods:
   - abs_val()- to find absolute value.
   - pow()-to to return the value of x raised to power y.
   - factor- to return factorial N.

Activity 1

Create a Class with instance attributes

Write a Python program to create a **Car** class with *`max_speed`* and *`mileage`* instance attributes. Methods:

- move()
- stop()
- acceleration()

***don't lost solution it will be useful during the lesson**

descriptors:

- *create class Car*
- *create class instance attributes*
- *define class methods*

Inheritance

NIS Astana, Grade 11

2024

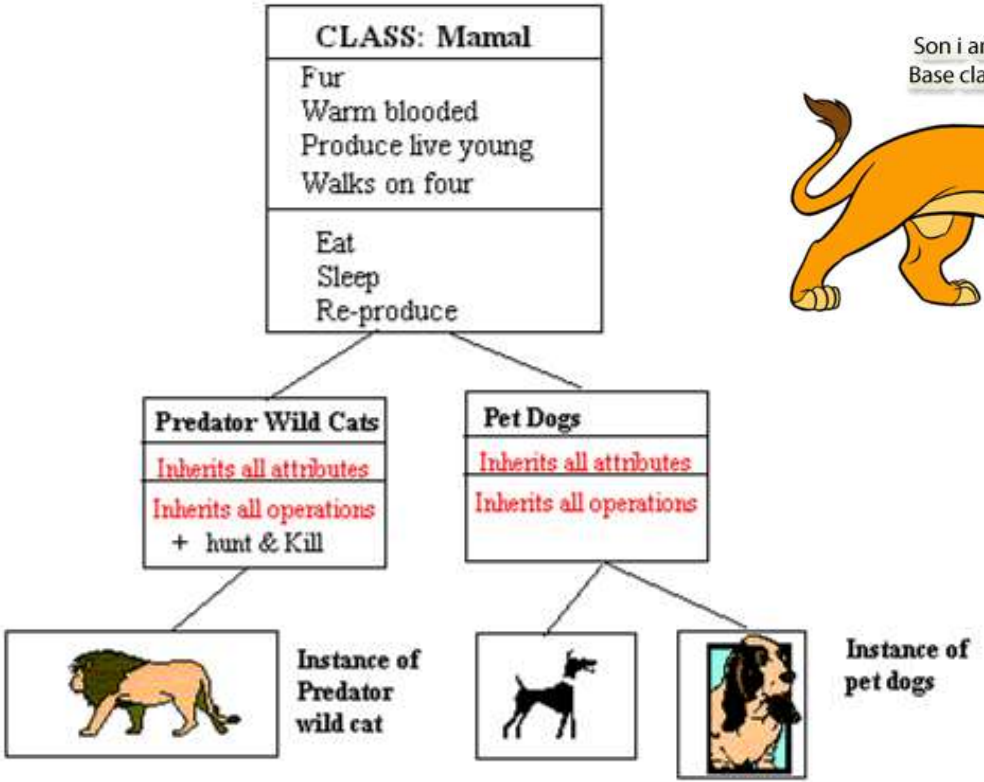
Learning objectives

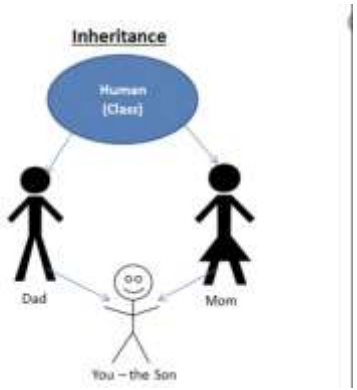
- 11.4.1.4 create a class hierarchy;
- 11.4.1.5 define class and instance identifiers in the proposed code snippet.

Success criteria:

- create child class
- able to know parent and child classes
- distinguish between parent and child class properties and methods
- solve problems by using inheritance in OOP

- explain the concept of inheritance with examples





Python Inheritance

Read information about Inheritance from next link.

descriptors:

- *able to define child class*
- *know how to use the super() function*
- *know how to add properties in child class*
- *know how to add methods in child class*

Practical task 1

Create classes “Mercedes”, “BMW”, “Bentley”, “Volvo”, ”Nissan”, “Audi”, ”Volkswagen”, “Toyota”, “Lexus” with ***max_speed*** and ***mileage*** instance attributes. **Methods:**

- move()
- stop()
- acceleration()

by using inheritance.

descriptors:

- *create child classes*
- *define properties of the child classes*
- *define methods of the child classes*
- *create objects*
- *call object methods*

Practical task 2


```
class algo:
        def isPrime(self,n):
                i=2
                while (i*i<=n):
                        if n%i==0:

        return "No"
                        i+=1
                return "Yes"
```

Create child class "algo2" that inherits from class "algo" and add two methods that return **largest common divisor (GCD)** and **smallest common multiple (LCM)**

descriptors:

- *create child class*
- *define properties of the child class*
- *define methods of the child class*
- *create object*
- *call object methods*

Practical task 3

Create a Python class called **Library** that represents a library. The class should have the following attributes and methods:

1. Attributes:
    - **books**: A list that stores the books available in the library.
    - **members**: A list that stores the members registered with the library.
2. Methods:
    - **add_book(self, book_title: str)**: Adds a book to the library. The method should take one parameter: **book_title** (a string representing the title of the book). The method should update the **books** list.
    - **remove_book(self, book_title: str)**: Removes a book from the library. The method should take one parameter: **book_title** (a string representing the title of the book to remove). If the book is found in the **books** list, it should be removed. If the book is not found, the method should print a message saying the book is not available.
    - **register_member(self, member_name: str)**: Registers a new member with the library. The method should take one parameter: **member_name** (a string representing the name of the member to register). The method should update the **members** list.
    - **print_books(self)**: Prints the list of books available in the library.
    - **print_members(self)**: Prints the list of members registered with the library.

3 mins

Post-lesson Reflect on the Learning

What concept or skill did you learn today?

What is unclear or confusing?

Does the information you learned today connect to something else you've learned in the past?

If you were going to teach this to someone else, how would you teach it?

What concepts, information, or skills would you like more time discussing, practice, or working with me to better understand?