Name: Sicheco, Vincent Rhian S.
Section: C203

Final Task 3. Polymorphism.

Requirements.

Finals Task 3. Simple Polymorphism

Problem. Chirp and Tweet

Create a simple program to demonstrate basic polymorphism with bird sounds.

Class - Bird:

- Methods:
  - `def make_sound(self) -> None`: An abstract method that represents making a sound. It doesn't have a specific implementation in the base class `Bird`.

Class - Sparrow (extends Bird):

- Methods:
  - `def make_sound(self) -> None`: Overrides the `make_sound` method from the base class `Bird`. It prints the sound "Chirp Chirp" when called.

Class - Parrot (extends Bird):

- Methods:
  - `def make_sound(self) -> None`: Overrides the `make_sound` method from the base class `Bird`. It prints the sound "Tweet Tweet" when called.

Class - BirdCage:

- Methods:
  - `def make_bird_sounds(self, birds: List) -> None`: Accepts a list of `Bird` objects as input. Iterates through the list of birds and calls the `make_sound` method on each bird to make its sound.

Note:

- *The test cases are not outputs of your main file but of a hidden test file. Create and implement the classes instructed to test your code.*
- *Each class should be defined in its own file, with the file name following camelCase conventions (e.g., `bankAccount.py`).*

TEST CASES:

Code.

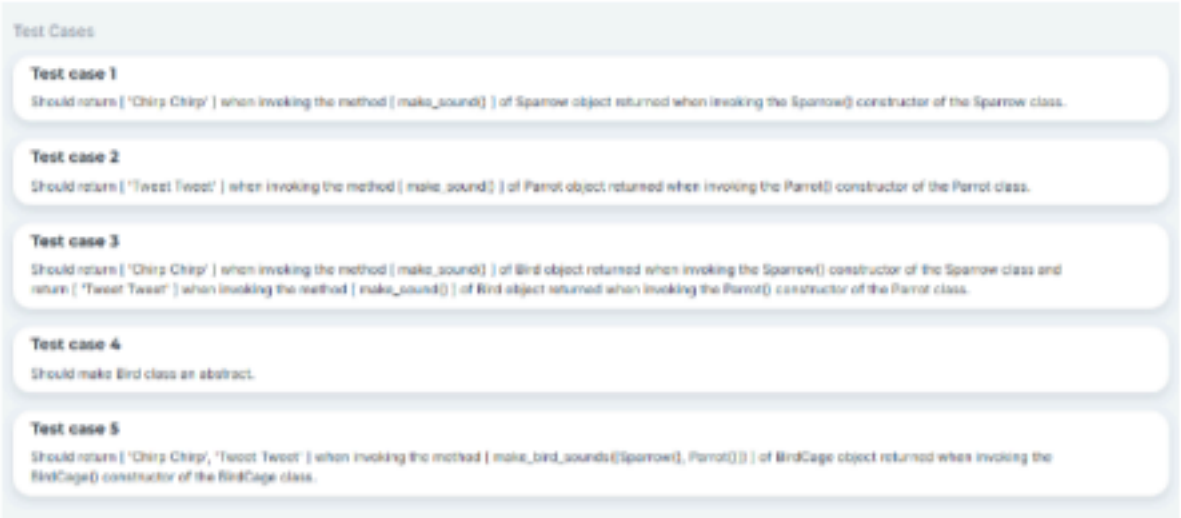
Birds.

```
3 usages
class Bird:
    1 usage (1 dynamic)
    def makeSound(self):
        pass


2 usages
class Sparrow(Bird):
    2 usages (1 dynamic)
    def makeSound(self):
        print("CHIRP CHIRP")


2 usages
class Parrot(Bird):
    2 usages (1 dynamic)
    def makeSound(self):
        print("TWEET TWEET")


2 usages
class BirdCage():
    1 usage
    def makeBirdSounds(self, birds: list):
        for birb in birds:
            birb.makeSound()
```

testBirb.

```
from bird import Bird, Sparrow, Parrot, BirdCage


1 usage
def test():
    birb1 = Sparrow()
    birb1.makeSound()

    birb2 = Parrot()
    birb2.makeSound()

    birbs = [birb1, birb2]
    cage = BirdCage()
    cage.makeBirdSounds(birbs)



if __name__ == '__main__':
    test()
```

Test Cases.



Output.

```
CHIRP CHIRP
TWEET TWEET
CHIRP CHIRP
TWEET TWEET
```