Name: Manaois, Ivan Bryan R. Date:11/19/25
Year/section: 2nd year BSCS C204 Subject:7OOP

Finals Task3. Polymorphism

Problem. Chirp and Tweet

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

Class - Bird:
. Methods: o 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: o 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: o 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: o 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:
Test case 1
Should return | 'Chirp Chirp'] when inoking the method [make_sound()] of Sparrow object retumed when invoking the Sparrow() constructor of the Sparrow class.

Test case 2
Should return ['Tweet Tweet'] when invoking the method [make_sound()] of Parrot object returned when invoking the Parrot()constructor of the Parrot class.

Test case 3
Should return ['Chirp Chirp'] when invoking the method [make_sound()] of Bird object returned when invoking the Sparrow() constructor of the Sparrow class and return ['Tweet Tweet"] when invoking the method [maka_sound()]of Bird object returned when invoking the Parrot() constructor of the Parrot class.

Test case 4
Should make Bird class an abstract.

Test case 5
Should return ['Chirp Chirp', 'Tweet Tweet'] when invoking the method [make_bird_sounds([Sparrow(), Parrot()])] of BirdCage object returned when invoking the BirdCage() constructor of the BirdCage class.

Code:

Python

bird.py

```python
from abc import ABC, abstractmethod

class Bird(ABC):

    @abstractmethod
    def make_sound(self) -> list:
        pass
```

Press Ctrl K to generate code

sparrow.py

```python
from bird import Bird

class Sparrow(Bird):

    def make_sound(self) -> list:
        return ["Chirp Chirp"]
```

Press Ctrl K to generate code

parrot.py

```python
from bird import Bird

class Parrot(Bird):

    def make_sound(self) -> list:
        return ["Tweet Tweet"]
```

Press Ctrl K to generate code

birdCage.py

```python
class BirdCage:

    def make_bird_sounds(self, birds: list) -> list:
        sounds = []
        for bird in birds:
            sounds += bird.make_sound()
        return sounds

```