

Final Task 3. Polymorphism

Mangalino, Chriselle C.

BSCS C203

Bird

```
bird.py × sparrow.py × parrot.py × birdCage.py × main.py ×
1 from abc import ABC, abstractmethod
2
3     2 usages
4     ⚡ class Bird(ABC):
5         1 usage
6             @abstractmethod
7                 def make_sound(self) -> None:
8                     pass
```

Sparrow

```
bird.py × sparrow.py × parrot.py × birdCage.py × main.py ×
1 from bird import Bird
2
3     ⚡ class Sparrow(Bird):
4         ⚡     def make_sound(self) -> None:
5             print("Chirp Chirp")
```

Parrot

```
bird.py × sparrow.py × parrot.py × birdCage.py × main.py ×
1 from bird import Bird
2
3     ⚡ class Parrot(Bird):
4         ⚡     def make_sound(self) -> None:
5             print("Tweet Tweet")
```

Birdcage

A screenshot of a code editor showing the `birdCage.py` file. The file contains the following Python code:

```
from typing import List
from bird import Bird

class BirdCage:
    def make_bird_sounds(self, birds: List[Bird]) -> None:
        for bird in birds:
            bird.make_sound()
```

Main

A screenshot of a code editor showing the `main.py` file. The file contains the following Python code:

```
from sparrow import Sparrow
from parrot import Parrot
from birdCage import BirdCage

if __name__ == "__main__":
    sparrow = Sparrow()
    parrot = Parrot()

    print("Test Case 1:")
    sparrow.make_sound()

    print("\nTest Case 2:")
    parrot.make_sound()

    print("\nTest Case 3:")
    birds = [sparrow, parrot]
    for b in birds:
        b.make_sound()

    print("\nTest Case 5:")
    cage = BirdCage()
    cage.make_bird_sounds(birds)]
```

Test Case 1

```
bird.py × sparrow.py × parrot.py × birdCage.py × main.py ×
1     from sparrow import Sparrow
2     from parrot import Parrot
3     from birdCage import BirdCage
4
5 ➤ 6     if __name__ == "__main__":
6         sparrow = Sparrow()
7         parrot = Parrot()
```

Test Case 2

```
8
9     print("Test Case 1:")
10    sparrow.make_sound()
11
```

Test Case 3

```
11
12    print("\nTest Case 2:")
13    parrot.make_sound()
14
```

Test Case 4

```
14  
15     print("\nTest Case 3:")  
16     birds = [sparrow, parrot]  
17     for b in birds:  
18         b.make_sound()  
19
```

Test Case 5

```
L 19  
S 20     print("\nTest Case 5:")  
E 21     cage = BirdCage()  
P 22     cage.make_bird_sounds(birds)  
P  
S
```

Sample Output

The screenshot shows the PyCharm interface with the 'Run' tab selected. The run configuration is set to 'main'. The output window displays the results of five test cases:

- Test Case 1:
Chirp Chirp
- Test Case 2:
Tweet Tweet
- Test Case 3:
Chirp Chirp
Tweet Tweet
- Test Case 4:
Chirp Chirp
Tweet Tweet
- Test Case 5:
Chirp Chirp
Tweet Tweet

At the bottom of the output window, it says "Process finished with exit code 0".