Finals Task 2. Inheritance

Problem School Performance

Note: You are to create 4 separate python files for this task:

- **performer.py(base class)**
- **singer.py(sub class)**
- **dancer.py(sub class)**
- **test_class.py – following the required test cases**

In a school musical performance, different types of performers participate. For this program, we will be implementing the performers.

Base Class - Performer:

- Properties:
  - `name` (type: str): Represents the name of the performer.
  - `age` (type: int): Represents the age of the performer.
- Constructor:
  - `__init__(self, name: str, age: int)`: Initializes the `name` and `age` properties.
- Getters
  - `get_name(self) -> str`: Returns the name
  - `get_age(self) -> int`: Returns the age

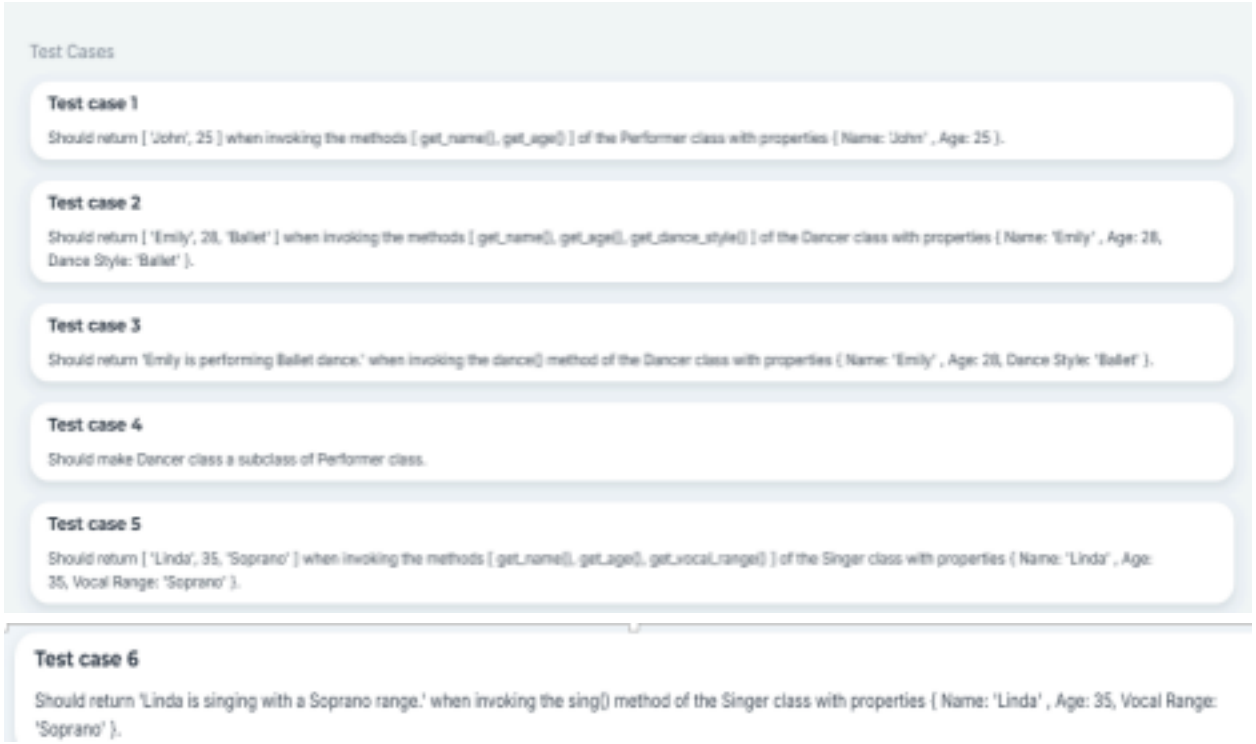
Subclass - Singer:

- Inherits From: `Performer`
- Additional Property:
  - `vocal range` (type: str): Represents the vocal range of the singer.
- Constructor:
  - `__init__(self, name: str, age: int, vocal_range: str)`: Initializes the `name` and `age` properties by calling the parent class's constructor and sets the `vocal_range` property.
- Getter:
  - `get_vocal_range(self) -> str`: Returns the vocal range of the singer.
- Method:
  - `sing(self) -> None`: Prints "{name} is singing with a {vocal_range} range."

Subclass - Dancer:

- Inherits From: `Performer`
- Additional Property:
  - `dance_style` (type: str): Represents the dance style of the dancer.
- Constructor:
  - `__init__(self, name: str, age: int, dance_style: str)`: Initializes the `name` and `age` properties by calling the parent class's constructor and sets the `dance_style` property.
- Getter:
  - `get_dance_style(self) -> str`: Returns the dance style of the dancer.
- Method:
  - `dance(self) -> None`: Prints "{name} is performing {dance_style} dance."

Sample output for the Test Class



CODE:

performer.py

```python
class Performer:
    def __init__(self, name: str, age: int):
        self._name = name
        self._age = age

    5 usages
    def get_name(self) -> str:
        return self._name

    3 usages
    def get_age(self) -> int:
        return self._age
```

singer.py

```python
from performer import Performer

2 usages
class Singer(Performer):
    def __init__(self, name: str, age: int, vocal_range: str):
        Performer.__init__(self, name, age)
        self._vocal_range = vocal_range

    1 usage
    def get_vocal_range(self) -> str:
        return self._vocal_range

    1 usage
    def sing(self) -> None:
        print(f"{self.get_name()} is singing with a {self._vocal_range} range.")
```

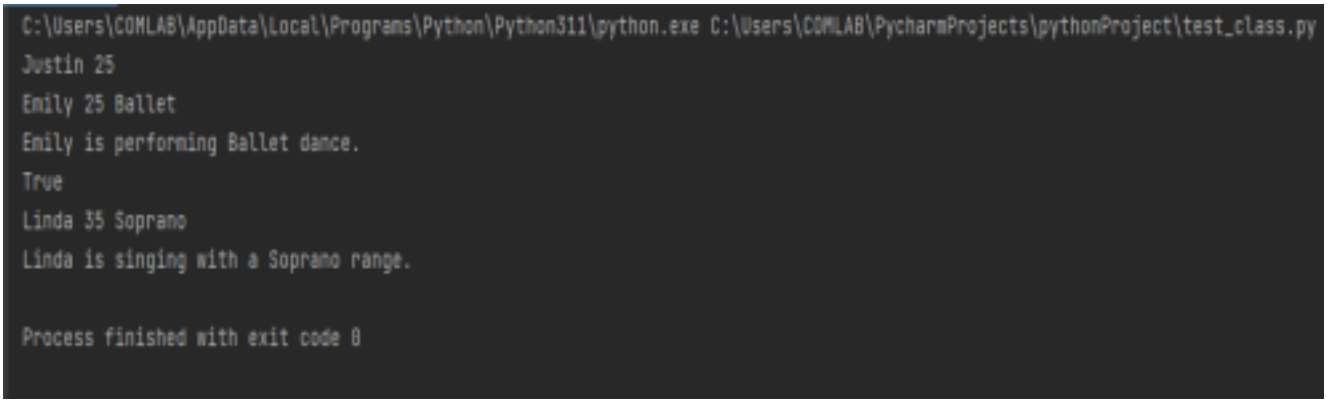
dancer.py

```python
from performer import Performer


3 usages
class Dancer(Performer):
    def __init__(self, name: str, age: int, dance_style: str):
        Performer.__init__(self, name, age)
        self._dance_style = dance_style

    1 usage
    def get_dance_style(self) -> str:
        return self._dance_style

    1 usage
    def dance(self) -> None:
        print(f"{self.get_name()} is performing {self._dance_style} dance.")
```

test_class.py

```python
from performer import Performer
from singer import Singer
from dancer import Dancer

performer = Performer( name: "Justin",  age: 25)
print(performer.get_name(),
      performer.get_age())

dancer = Dancer( name: "Emily",  age: 25,  dance_style: "Ballet")
print(dancer.get_name(),
      dancer.get_age(),
      dancer.get_dance_style())

dancer.dance()

print(issubclass(Dancer, Performer))

singer = Singer( name: "Linda",  age: 35,  vocal_range: "Soprano")
print(singer.get_name(),
      singer.get_age(),
      singer.get_vocal_range())

singer.sing()
```

OUTPUT:


C:\Users\COMLAB\AppData\Local\Programs\Python\Python311\python.exe C:\Users\COMLAB\PycharmProjects\pythonProject\test_class.py
Justin 25
Emily 25 Ballet
Emily is performing Ballet dance.
True
Linda 35 Soprano
Linda is singing with a Soprano range.
Process finished with exit code 0