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

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:

o nane (type: str]: Represents the name of the performer.

o age (type: int): Represents the age of the performer.

. Constructor:

o _init_(self, name: str, age: int): Initializes the name and age properties.

. Getters

o get_name(self) -> str: Returns the name

o get_age[self) -> int: Returns the age

Subclass - Singer:

. Inherits From: Performer

. Additional Property:

o vocal range (type: str]: Represents the vocal range of the singer.

. Constructor:

o _ init_ (self, names 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:

o get_vocal_range(self) -> str: Returns the vocal range of the singer.

. Method:

o sing(self) -> None: Prints "[name] is singing with a [vocalrange) range."

Subclass - Dancer:

. Inherits From: per-Former

. Additional Property:

- dance_style (type: str): Represents the dance style of the dancer.

. Constructor:

o _Init_(self, mame: str, age: int, dance_style: str): Initlalizes the name and age propertles by calling the parent class's constructor and sets the dance_style property.

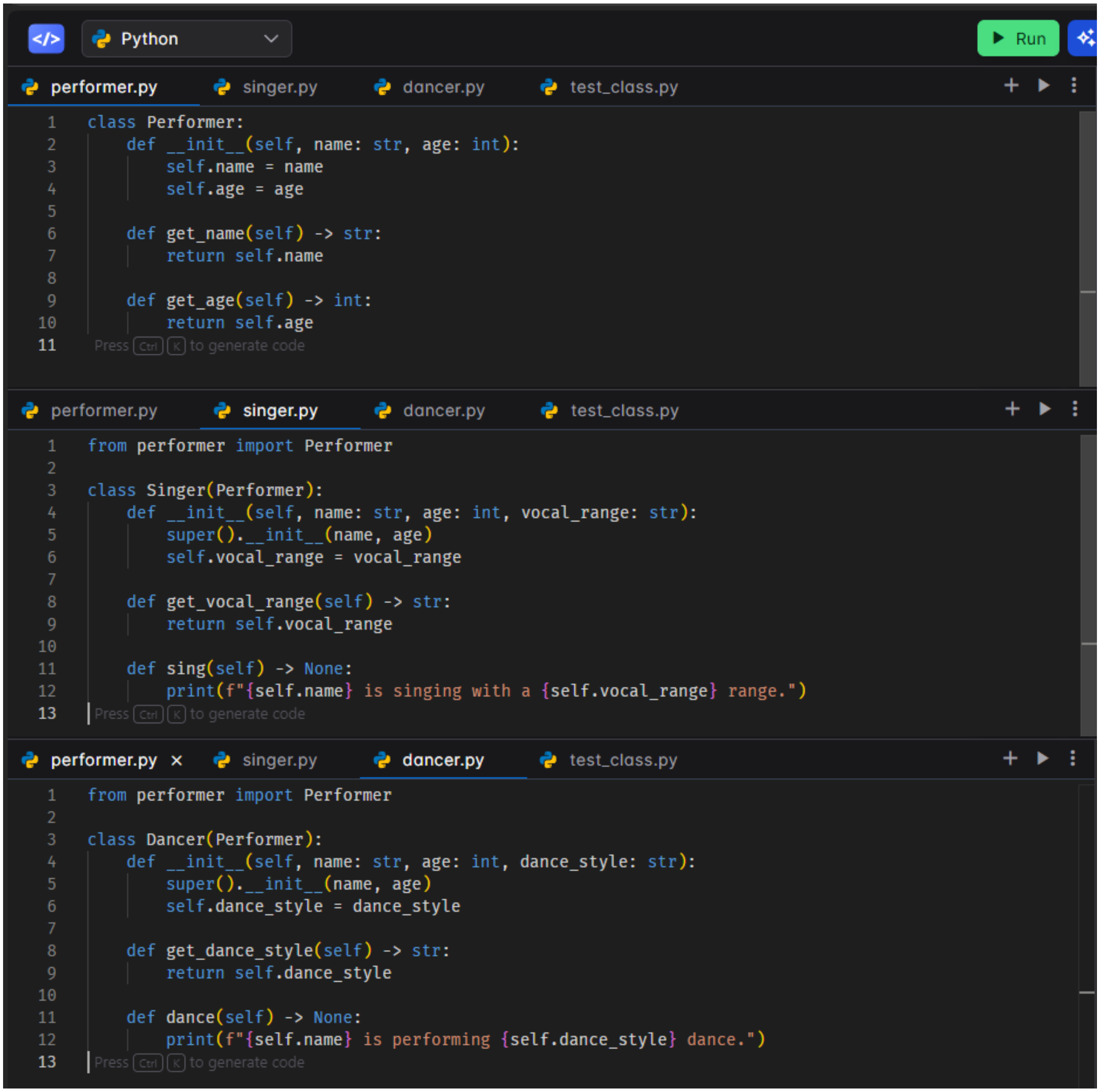
· Getter:

o get_dance_style(self) -> str: Returns the dance style of the dancer.

. Method:

a dance(self) > None: Prints "(name) is performing (dance_style] dance."

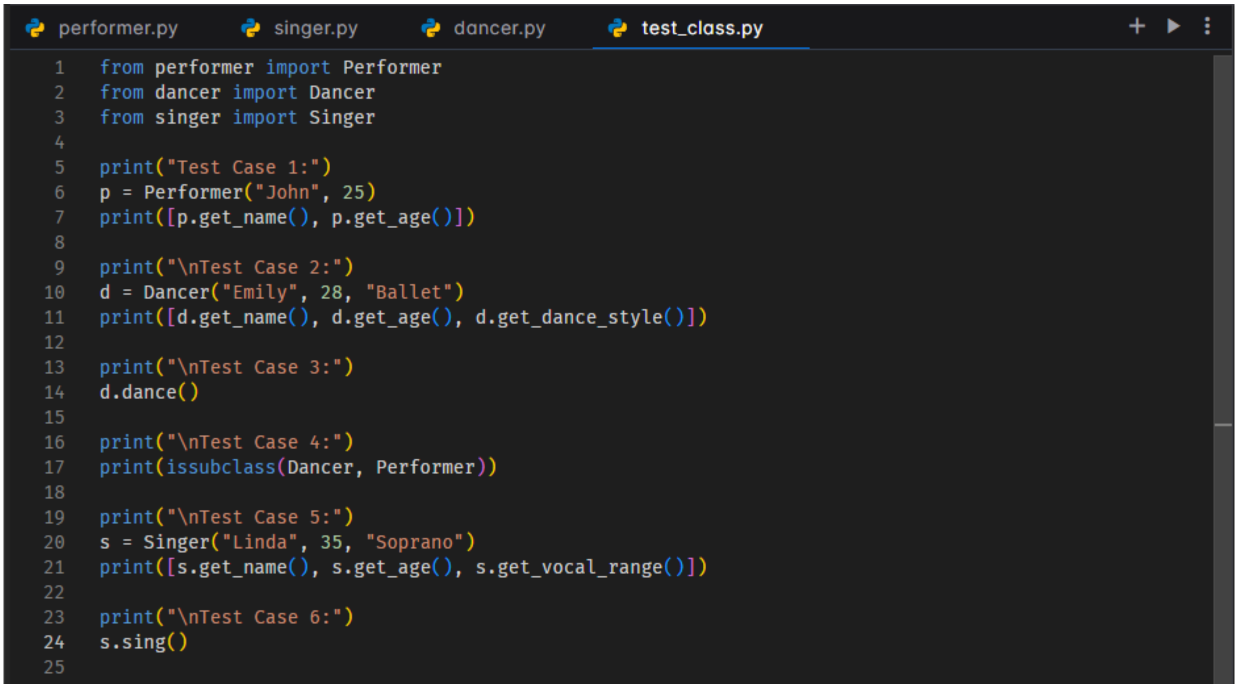
Code:



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

    def get_name(self) -> str:
        return self.name

    def get_age(self) -> int:
        return self.age
```

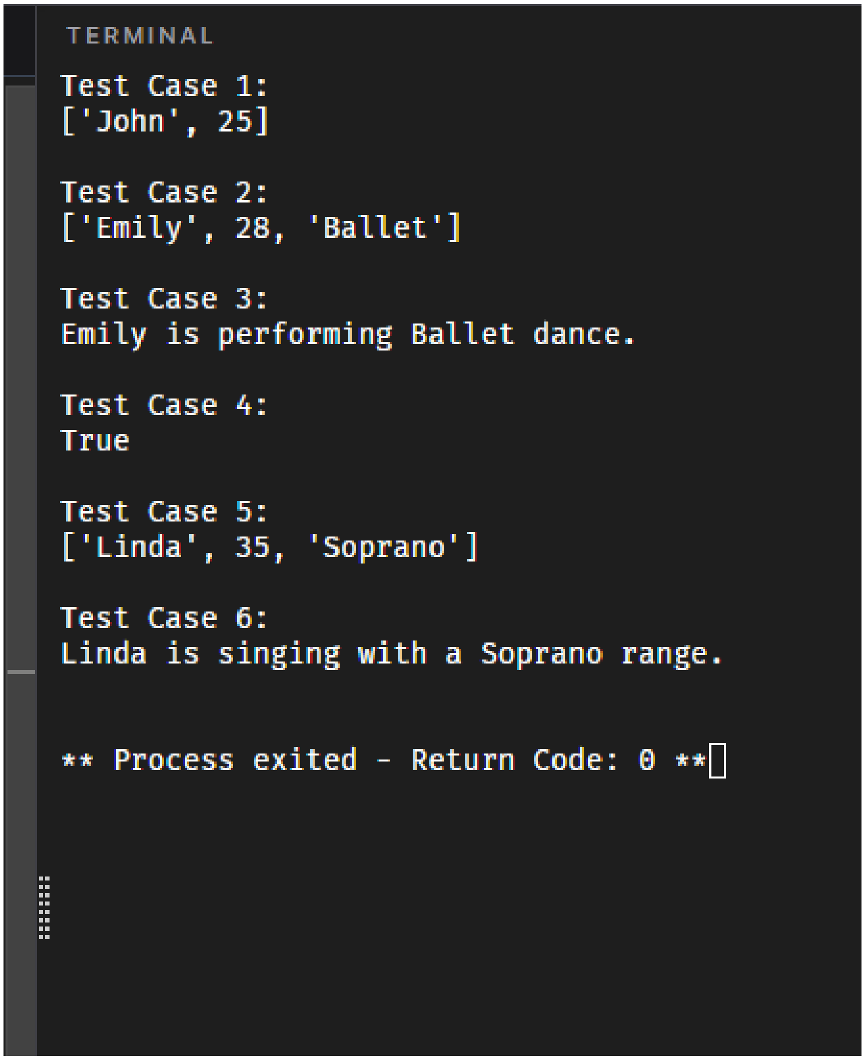
```python
from performer import Performer

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

    def get_vocal_range(self) -> str:
        return self.vocal_range

    def sing(self) -> None:
        print(f"{self.name} is singing with a {self.vocal_range} range.")
```

```python
from performer import Performer

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

    def get_dance_style(self) -> str:
        return self.dance_style

    def dance(self) -> None:
        print(f"{self.name} is performing {self.dance_style} dance.")
```



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

print("Test Case 1:")
p = Performer("John", 25)
print([p.get_name(), p.get_age()])

print("\nTest Case 2:")
d = Dancer("Emily", 28, "Ballet")
print([d.get_name(), d.get_age(), d.get_dance_style()])

print("\nTest Case 3:")
d.dance()

print("\nTest Case 4:")
print(issubclass(Dancer, Performer))

print("\nTest Case 5:")
s = Singer("Linda", 35, "Soprano")
print([s.get_name(), s.get_age(), s.get_vocal_range()])

print("\nTest Case 6:")
s.sing()

```

Output:



```
Test Case 1:
['John', 25]

Test Case 2:
['Emily', 28, 'Ballet']

Test Case 3:
Emily is performing Ballet dance.

Test Case 4:
True

Test Case 5:
['Linda', 35, 'Soprano']

Test Case 6:
Linda is singing with a Soprano range.


** Process exited - Return Code: 0 **
```