Python -pytest

Prompt

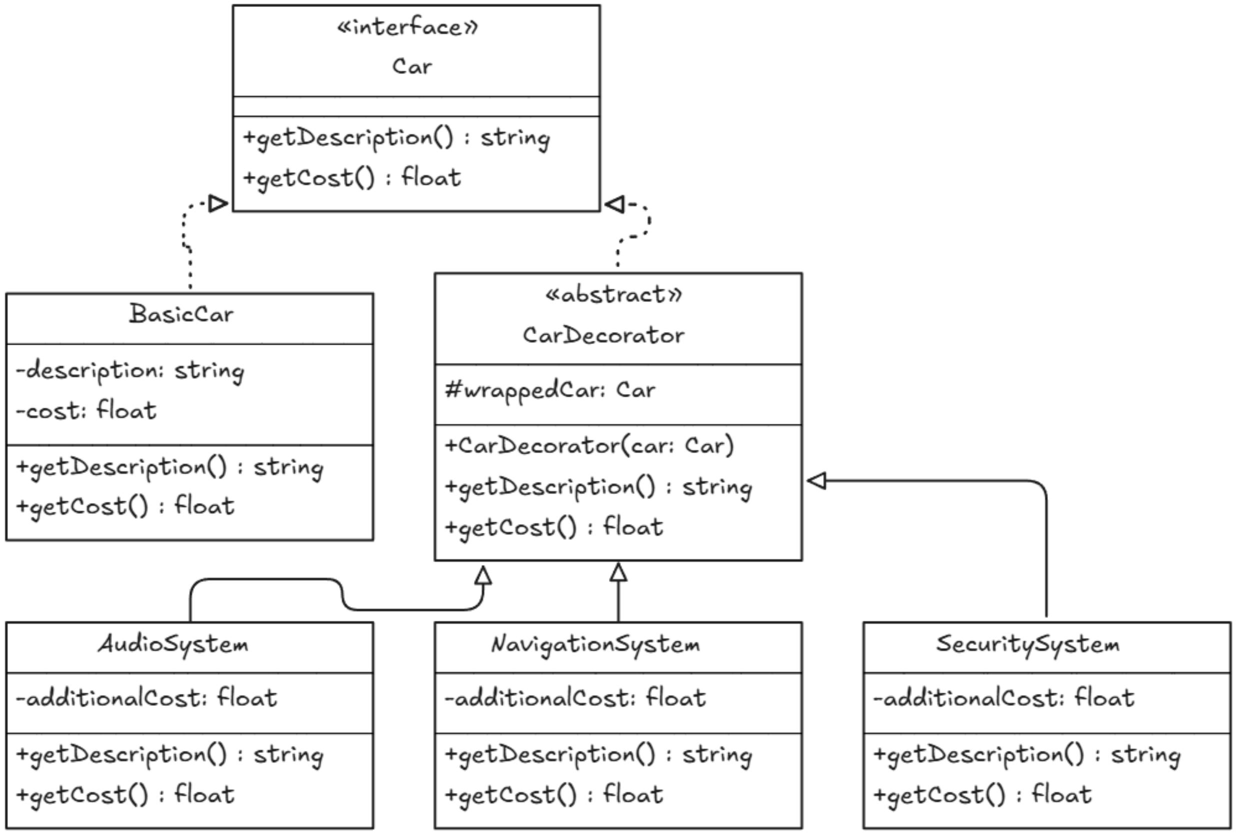
-Decorator

Use the Decorator pattern to write code in Python programming language and write pytest to test the given code that have 100% statement coverage from the requirements below.

Use Decorator pattern to design automotive system development by adding features to cars, such as audio system, navigation system or security systems without changing the main structure of the car.

Diagram

Decorator



Gemini 1.5 Flash

Decorator

รอบที่ 3

Code.py

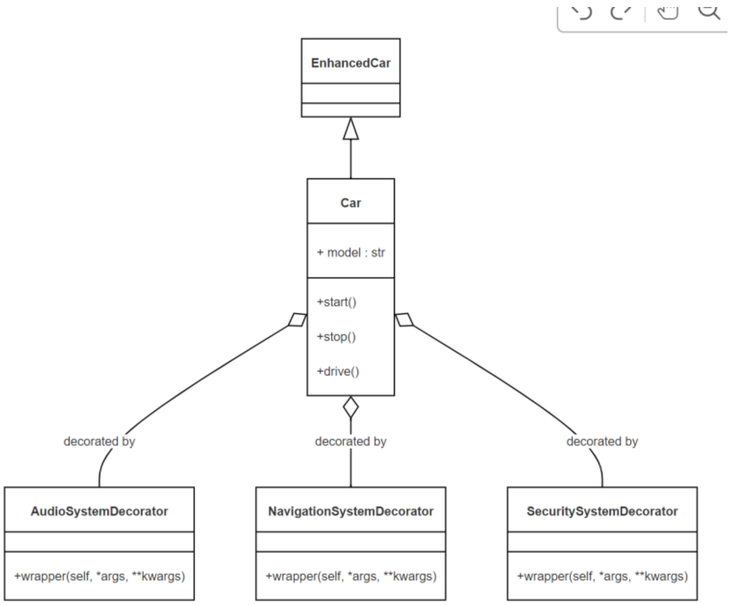
```
class Car:
    def __init__(self, model):
        self.model = model

    def start(self):
        print(f"Starting {self.model} car")

    def stop(self):
        print(f"Stopping {self.model} car")

    def drive(self):
        print(f"Driving {self.model} car")


def audio_system(func):
    def wrapper(self, *args, **kwargs):
        print("Turning on the audio system")
        func(self, *args, **kwargs)
        print("Turning off the audio system")
    return wrapper


def navigation_system(func):
    def wrapper(self, *args, **kwargs):
        print("Navigating to destination")
        func(self, *args, **kwargs)
        print("Navigation completed")
    return wrapper
```

```
def security_system(func):
    def wrapper(self, *args, **kwargs):
        print("Activating security system")
        func(self, *args, **kwargs)
        print("Deactivating security system")
    return wrapper



@audio_system
@navigation_system
@security_system
class EnhancedCar(Car):
    pass
```

Test.py

```
import pytest
from code import EnhancedCar,capsys

class TestCar:

    def test_start(self):
        car = EnhancedCar("Tesla")
        car.start()
        assert "Starting Tesla car" in capsys.readouterr().out

    def test_stop(self):
        car = EnhancedCar("Tesla")
        car.stop()
        assert "Stopping Tesla car" in capsys.readouterr().out

    def test_drive(self):
        car = EnhancedCar("Tesla")
        car.drive()
        assert "Driving Tesla car" in capsys.readouterr().out

    def test_audio_system(self):
        car = EnhancedCar("Tesla")
        car.start()
        assert "Turning on the audio system" in capsys.readouterr().out
        assert "Turning off the audio system" in capsys.readouterr().out

    def test_navigation_system(self):
        car = EnhancedCar("Tesla")
        car.drive()
        assert "Navigating to destination" in capsys.readouterr().out
        assert "Navigation completed" in capsys.readouterr().out

    def test_security_system(self):
        car = EnhancedCar("Tesla")
        car.start()
        assert "Activating security system" in capsys.readouterr().out
        assert "Deactivating security system" in capsys.readouterr().out
```

Diagram From Code:



Result:

split class in code.py

TestCar.py(Edit)

```
from enhancedCar import EnhancedCar
from capsys import capsys
```

6 Fail