Class Decorator

1. Jalankan program berikut, screenshot outputnya dan berikan analisis/penjelasan program!

```
class Person:
    TITLES = ('Dr', 'Mr', 'Mrs', 'Ms')

    def __init__(self, name, surname):
        self.name = name
        self.surname = surname

    def fullname(self): # instance method
        # instance object accessible through self
        return "%s %s" % (self.name, self.surname)

    @property
    def fullname2(self):
        return "%s %s" % (self.name, self.surname)

    @classmethod
    def allowed_titles_starting_with(cls, startswith): # class method
        # class or instance object accessible through cls
        return [t for t in cls.TITLES if t.startswith(startswith)]

    @staticmethod
    def allowed_titles_ending_with(endswith): # static method
        # no parameter for class or instance object
        # we have to use Person directly
        return [t for t in Person.TITLES if t.endswith(endswith)]

jane = Person("Jane", "Smith")

print(jane.fullname())
print(jane.fullname2) # no brackets!

print(jane.allowed_titles_starting_with("M"))
print(Person.allowed_titles_starting_with("M"))

print(jane.allowed_titles_ending_with("s"))
print(Person.allowed_titles_ending_with("s"))
```

2. Buatlah program menggunakan class decorator @classmethod, @staticmethod, dan @property! Boleh dalam 1 program atau dipisah 1 class decorator per program.