15 Python基礎

クラスまとめ

- クラス定義が正しくできるか？
    - コンストラクタの働き
    - クラス変数・インスタンス変数の区別と使い方
    - メソッドの作成方法
        - セッター・ゲッターの役割と作り方
        - プロパティを用いた属性の読み出しと設定
- 継承を正しく定義できるか？
    - メソッドのOverrideができるか？
    - 親クラスのメソッド呼び出しができるか？
    - 多重継承できるか？

課題1

- 以下のプログラムを順を追って作成してください。
    - 正しく動作するかを確認するmainを適当に作成してください。

1. クラスPersonを作成する【class_person1.py】
    - first_name(名)とlast_name(姓)をプロパティとして持つ
    - メソッド `full_name()` を持つ(プロパティでも良い。利用時に変更してOK)
        - 姓名を返す
    - インスタンスを生成する。
    - `full_name()` の結果を出力し、正しく動作するのを確認する。

【実行結果】（例）

```
名無しの権兵衛
名無しの花子
山田太郎
```

【コード例】

```python
class Person:
  pass

# main
man = Person()
print(man.full_name())
# プロパティの場合
# print(man.full_name)

man = Person('花子')
print(man.full_name())
# プロパティの場合
# print(man.full_name)

man = Person('太郎', '山田')
print(man.full_name())
# プロパティの場合
# print(man.full_name)
```

2. 上記、Personクラスを継承し、以下のクラスを作成してください。【class_person2.py】
    - クラスCustomer　　（Customerは顧客を意味する）
        - company_idとcompany_nameをプロパティで持つ
    - cus = Customer()でインスタンスを生成する。
        - 姓・名を設定する。
        - full_nameの結果を表示する。
        - company_idと会社名を表示する。

【実行結果】（例）

```
山田太郎
鈴木花子 id:10 会社名:橋本商会
```

【コード例】

```python
class Person:
```

```
  pass


class Customer(Person):
  pass


# main
man = Person('太郎', '山田')
print(man.full_name())
# プロパティの場合
# print(man.full_name)

cust = Customer(company_id=10, company_name='橋本商会')
cust.first_name = '花子'
cust.last_name = '鈴木'
print(f'{cust.full_name()} id:{cust.company_id} 会社名:{cust.company_name}')
# full_nameがプロパティの場合
# print(f'{cust.full_name} id:{cust.company_id} 会社名:{cust.company_name}')
```

3. 上記、Personクラスにshowメソッドを作成してください。【class_person3.py】
    - Person.showメソッド
        - 姓名を出力する

【実行結果】（例）

```
氏名：山田太郎
氏名：鈴木花子
```

【コード例】

```
class Person:
  pass


class Customer(Person):
  pass


# main
man = Person('太郎', '山田')
man.show()

cust = Customer(company_id=10, company_name='橋本商会')
```

```
cust.first_name = '花子'
cust.last_name = '鈴木'
cust.show()
```

4. Customerクラスにshowメソッドをオーバーライドしてください。【class_person4.py】
    - Costomer.showメソッド
        - 姓名を出力し、会社IDと会社名を出力する

【実行結果】（例）

```
氏名：山田太郎
氏名：鈴木花子
  id:10 会社名:橋本商会
```

【コード例】

```
class Person:
  pass


class Customer(Person):
  pass


# main
man = Person('太郎', '山田')
man.show()

cust = Customer(company_id=10, company_name='橋本商会')
cust.first_name = '花子'
cust.last_name = '鈴木'
cust.show()
```

課題2

- 以下のプログラムを作成してください。【class_inheritance.py】
    - クラス Shapeを作成する。

        - クラス変数としてmember_countを持っている。
            - member_countは、オブジェクトの個数覚えている
        - メソッド
            - コンストラクタでオブジェクト数を数える（カウントアップする）
            - 面積を計算するarea()を持っている
            - 面積を表示するshow_area()を持っている
    - Shapeクラスを継承し、三角形を表すTriangleクラスを作成する。
        - area( ) をoverride（上書き）し、自分自身の面積を求める。
    - Shapeクラスを継承し、長方形を表すRectangleクラスを作成する。
        - area( ) をoverrideし、自分自身の面積を求める。
    - Triangle、Rectangleはインスタンス変数として以下を持つ
        - 底辺の長さ
        - 高さ（長さ）
    - Triangle、Rectangleはコンストラクタを持つ
        - 必ず `super().__init__()` を利用する
    - 適当なメイン関数を作成し、正しく計算していることを確認する。
    - 最後に、何個のオブジェクトを作成したのかを表示する。

【class_inheritance.py】

```
class Shape:
    pass


class Triangle(Shape):
    pass


class Rectangle(Shape):
    pass


# Main:
print('Shapeを1個作成')
obj0 = Shape()
```

```
obj0.show_area()
print('Triangleを2個作成')
obj1 = Triangle()
obj2 = Triangle(10, 5)
obj1.show_area()
obj2.show_area()
print('Rectangleを2個作成')
obj3 = Rectangle()
obj4 = Rectangle(10, 5)
obj3.show_area()
obj4.show_area()
print(f'オブジェクトの個数：{Shape.member_count}')
```

【実行結果】

```
Shapeを1個作成
面積は0です
Triangleを2個作成
面積は0.5です
面積は25.0です
Rectangleを2個作成
面積は1です
面積は50です
オブジェクトの個数：5
```