

제목 : 제 11 장 연습문제 팀원 소스코드 비교
분석 리포트

학번 : C235418

이름 : 최민혁

팀 번호: 11

팀 선정프로그램 작성자 학번

문제 1 - c235418

문제 2 - c235418

문제 3 - c235357

문제 4 - c235418

문제 5 - c235357

문제 6 - c235357

1번 문제에 대한 "팀선정 프로그램 소스코드"

#1번 C235418 최민혁

#1. 2차원 도형을 나타내는 Shape 클래스를 완성하기.

#2. Shape를 상속받아서 Circle 클래스를 정의하기.

##Shape 클래스는 color와 filled 속성을 가짐.

##Circle 클래스는 반지름을 나타내는 radius 변수를 추가로 정의함.

```
class Shape:
```

```
    def __init__(self, color = 'yellow', filled = True):
```

```
        self.__color = color
```

```
        self.__filled = filled
```

```
    def __str__(self):
```

```
        return f'({self.__color},{self.__filled})'
```

```
class Circle(Shape):
```

```
    def __init__(self, color, filled, radius):
```

```
        super().__init__(color,filled)
```

```
        self.radius = radius
```

```
    def __str__(self):
```

```
        return super().__str__() + f'(radius = {self.radius})'
```

```
    def area(self):
```

```
        return f'{self.radius*self.radius*3.14}'
```

```
def main():  
  
    a = Shape()  
  
    b = Shape("red")  
  
    print(a,b)  
  
    c = Circle("blue", False, 10)  
  
    print(c)  
  
    print(c.area())
```

```
main()
```

비교 프로그램 소스코드

#1 C235357 전채운

```
class Shape:  
  
    def __init__(self, color="yellow", filled=True):  
  
        self.__color = color  
  
        self.__filled = filled  
  
  
    def __str__(self):  
  
        return f'({self.__color},{self.__filled})'
```

```
class Circle(Shape):  
  
    def __init__(self, color, filled, radius):  
  
        super().__init__(color, filled)  
  
        self.__radius = radius  
  
  
    def area(self):
```

```
    return 3.14 * self.__radius * self.__radius
```

```
def __str__(self):
```

```
    return f'{super().__str__}(radius = {self.__radius})'
```

```
def main():
```

```
    a = Shape()
```

```
    b = Shape("red")
```

```
    print(a,b)
```

```
    c = Circle("blue",False,10)
```

```
    print(c)
```

```
    print(c.area())
```

```
main()
```

비교분석: 선정된 프로그램은 calArea메소드에서 넓이의 값을 result에 저장한 후에 result값을 출력했고 비교 프로그램은 넓이의 값을 바로 출력했다.

2번 문제에 대한 "팀선정 프로그램 소스코드"

#2번 C235418 최민혁

```
class Shape:
```

```
    def __init__(self, color = 'yellow', filled = True ):
```

```
        self.__color = color
```

```
        self.__filled = filled
```

```
    def __str__(self):
```

```
        return f'({self.__color},{self.__filled})'
```

```
class Rectangle(Shape):
```

```
    def __init__(self,color, filled, width, height):
```

```
        super().__init__(color, filled)
```

```
        self.width = width
```

```
        self.height = height
```

```
    def __str__(self):
```

```
        return super().__str__() + f'({self.width},{self.height})'
```

```
    def area(self):
```

```
        return f'{self.width*self.height}'
```

```
def main():
```

```
    c = Rectangle("blue",False,10,20)
```

```
    print(c)
```

```
    print(c.area())
```

main()

비교 프로그램 소스코드

#2 C235357 전체운

class Shape:

def __init__(self, color, filled):

self.__color = color

self.__filled = filled

def __str__(self):

return f'({self.__color},{self.__filled})'

class Rectangle(Shape):

def __init__(self, color, filled, width, height):

super().__init__(color, filled)

self.__width = width

self.__height = height

def area(self):

return self.__width * self.__height

def __str__(self):

return f'({super().__str__()}({self.__width},{self.__height})'

def main():

```
c = Rectangle("blue",False,10,20)
```

```
print(c)
```

```
print(c.area())
```

```
main()
```

비교분석: 선정된 프로그램은 Point 클래스의 move 메소드에서 += 연산자를 통해서 값을 업데이트 해주고 있고 비교프로그램은 = + 연산자를 사용해서 값을 업데이트하고 있다.

3번 문제에 대한 "팀선정 프로그램 소스코드"

#3 C235357 전채운

```
class CPoint:

    def __init__(self, x=0, y=0):

        self.__x = x

        self.__y = y


    def move(self, a, b):

        self.__x += a

        self.__y += b

        return self


    def __str__(self):

        return f'pos({self.__x},{self.__y})'
```

```
def main():

    p = CPoint()

    q = CPoint(3,4)

    print(p,q)

    print(q.move(-4,5))

    print(p.move(5,6).move(2,3))
```

```
main()
```

비교 프로그램 소스코드

#3번 C235418 최민혁


```
class CPoint:

    def __init__(self, x = 0, y = 0):

        self.x = x

        self.y = y

    def __str__(self):

        return f'pos({self.x},{self.y})'

    def move(self,a,b):

        self.x += a

        self.y += b

        return self

def main():

    p = CPoint()

    q = CPoint(3,4)

    print(p,q)

    print(q.move(-4,5))

    print(p.move(5,6).move(2,3))

main()
```

비교분석: 선정된 프로그램은 리스트에 trick 요소들을 .append를 통해 더한 후 ".join()을 통해 trick들을 출력했고 비교프로그램은 문자열 +연산자를 사용해서 trick들을 더해서 출력했다.

4번 문제에 대한 "팀선정 프로그램 소스코드"

#4번 C235418 최민혁

```
class CPoint:
```

```
    def __init__(self, x = 0, y = 0):
```

```
        self.x = x
```

```
        self.y = y
```

```
    def __str__(self):
```

```
        return f'pos({self.x},{self.y})'
```

```
    def move(self,a,b):
```

```
        self.x += a
```

```
        self.y += b
```

```
        return self
```

```
class Shape(CPoint):
```

```
    def __init__(self, color = 'yellow', filled = True, x = 0, y = 0):
```

```
        super().__init__(x,y)
```

```
        self.color = color
```

```
        self.filled = filled
```

```
    def __str__(self):
```

```
        return super().__str__()+f'({self.color},{self.filled})'
```

```
class Circle(Shape):
```

```
    def __init__(self, color, filled, radius):
```

```
        super().__init__(color,filled)
```

```
        self.radius = radius
```

```
def __str__(self):  
    return super().__str__() + f'(radius = {self.radius})'
```

```
def area(self):  
    return f'{self.radius*self.radius*3.14}'
```

```
class Rectangle(Shape):  
  
    def __init__(self,color, filled, width, height):  
        super().__init__(color, filled)  
        self.width = width  
        self.height = height  
  
    def __str__(self):  
        return super().__str__() + f'({self.width},{self.height})'  
  
    def area(self):  
        return f'{self.width*self.height}'
```

```
def main() :  
  
    a = Shape()  
    b = Shape("red")  
    c = Shape("black",False,1,2)  
    print(a)  
    print(b)  
    print(c)  
    a.move(2,3)  
    print(a)  
    print(b.move(4,5))  
    d = Circle("blue",False,10).move(3,4)
```

```
print(d)

e = Rectangle("blue",False,10,20)

print(e.move(7,8))

main()
```

비교 프로그램 소스코드

#4 C235357 전체운

```
class CPoint:

    def __init__(self, x=0, y=0):

        self.__x = x

        self.__y = y

    def move(self, a, b):

        self.__x += a

        self.__y += b

        return self

    def __str__(self):

        return f'pos({self.__x},{self.__y})'

class Shape(CPoint):

    def __init__(self, color="yellow", filled=True, x=0, y=0):

        super().__init__(x,y)

        self.__color = color

        self.__filled = filled
```

```
def __str__(self):  
    return f'{super().__str__()}({self.__color},{self.__filled})'
```

```
class Circle(Shape):
```

```
    def __init__(self, color, filled, radius):  
        super().__init__(color, filled)  
        self.__radius = radius
```

```
    def __str__(self):  
        return f'{super().__str__()}(radius = {self.__radius})'
```

```
class Rectangle(Shape):
```

```
    def __init__(self, color, filled, width, height):  
        super().__init__(color, filled)  
        self.__width = width  
        self.__height = height
```

```
    def __str__(self):  
        return f'{super().__str__()}({self.__width},{self.__height})'
```

```
def main():
```

```
    a = Shape()  
    b = Shape("red")  
    c = Shape("black",False,1,2)
```

```
print(a)

print(b)

print(c)

a.move(2,3)

print(a)

print(b.move(4,5))

d = Circle("blue",False,10).move(3,4)

print(d)

e = Rectangle("blue",False,10,20)

print(e.move(7,8))
```

```
main()
```

비교분석: 선정된 프로그램은 `init` 메소드에서 `serialNumber`를 업데이트했고 비교 프로그램은 `serialNumber`를 `__str__` 메소드에서 업데이트했다. 그래서 선정된 프로그램은 `__init__`가 불릴때마다 `serialNumber`가 할당되고, 비교 프로그램은 `print()`문에서 `__str__`이 불릴 때 `serialNumber`가 할당되게 된다. 그래서 비교프로그램은 출력 순서가 달라지면 다른 `serialNumber`가 할당되게 된다.

5번 문제에 대한 "팀선정 프로그램 소스코드"

#5 C235357 전채운

class Shape:

```
def __init__(self,color="yellow",filled=True,cpoint="pos(0,0)":
```

```
    self.__color = color
```

```
    self.__filled = filled
```

```
    self.__cpoint = cpoint
```

```
def move(self, a, b):
```

```
    self.__cpoint = f'pos({a},{b})'
```

```
    return self
```

```
def __str__(self):
```

```
    return f'{self.__cpoint}({self.__color},{self.__filled})'
```

class CPoint:

```
def __init__(self, x=0, y=0):
```

```
    self.__x = x
```

```
    self.__y = y
```

```
def move(self, a, b):
```

```
    self.__x += a
```

```
    self.__y += b
```

```
    return self
```

```
def __str__(self):
```

```
return f'pos({self.__x},{self.__y})'
```

```
class Circle(Shape):
```

```
    def __init__(self, color, filled, radius):
```

```
        super().__init__(color, filled)
```

```
        self.__radius = radius
```

```
    def area(self):
```

```
        return 3.14 * self.__radius * self.__radius
```

```
    def __str__(self):
```

```
        return f'{super().__str__()}(radius = {self.__radius})'
```

```
class Rectangle(Shape):
```

```
    def __init__(self, color, filled, width, height):
```

```
        super().__init__(color, filled)
```

```
        self.__width = width
```

```
        self.__height = height
```

```
    def __str__(self):
```

```
        return f'{super().__str__()}({self.__width},{self.__height})'
```

```
def main():
```

```
    a = Shape()
```

```
    b = Shape("red")
```

```
    cpoint = CPoint(4,7)
```



```
c = Shape("black",False,cpoint)

print(a)

print(b)

print(c)

a.move(2,3)

print(a)

print(b.move(4,5))

d = Circle("blue",False,10).move(3,4)

print(d)

e = Rectangle("blue",False,10,20)

print(e.move(7,8))
```

```
main()
```

비교 프로그램 소스코드

#5번 C235418 최민혁

```
class Shape:
```

```
    def __init__(self, color = 'yellow', filled = True, cpoint = None):

        if cpoint == None:

            self.cpoint = CPoint()

        else:

            self.cpoint = cpoint #cpoint에 좌표를 나타내는 객체를 할당받았을 때

        self.color = color

        self.filled = filled
```

```
def __str__(self):  
    return f'pos({self.cpoint})({self.color},{self.filled})'
```

```
def move(self,a,b):  
    CPoint.move(self.cpoint,a,b)  
    return self
```

```
class CPoint:
```

```
    def __init__(self, x = 0, y = 0):  
        self.x = x  
        self.y = y
```

```
    def __str__(self):  
        return f'pos({self.x},{self.y})'
```

```
    def move(self,a,b):  
        self.x += a  
        self.y += b  
        return self
```

```
class Circle(Shape):
```

```
    def __init__(self, color, filled, radius):  
        super().__init__(color,filled)  
        self.radius = radius
```

```
    def __str__(self):  
        return super().__str__() + f'(radius = {self.radius})'
```

```
    def area(self):  
        return f'{self.radius*self.radius*3.14}'
```

```
class Rectangle(Shape):

    def __init__(self,color, filled, width, height):

        super().__init__(color, filled)

        self.width = width

        self.height = height

    def __str__(self):

        return super().__str__() + f'({self.width},{self.height})'

    def area(self):

        return f'{self.width*self.height}'
```

```
def main() :

    a = Shape()

    b = Shape("red")

    cpoint = CPoint(4,7)

    c = Shape("black",False,cpoint)

    print(a)

    print(b)

    print(c)

    a.move(2,3)

    print(a)

    print(b.move(4,5))

    d = Circle("blue",False,10).move(3,4)

    print(d)

    e = Rectangle("blue",False,10,20)

    print(e.move(7,8))

main()
```

비교분석: 선정된 프로그램은 `init` 메소드에서 `serialNumber`를 업데이트했고 비교 프로그램은 `serialNumber`를 `__str__` 메소드에서 업데이트했다. 그래서 선정된 프로그램은 `__init__`가 불릴때마다 `serialNumber`가 할당되고, 비교 프로그램은 `print()`문에서 `__str__`이 불릴 때 `serialNumber`가 할당되게 된다. 그래서 비교프로그램은 출력 순서가 달라지면 다른 `serialNumber`가 할당되게 된다. 이외에는 거의 동일한 코드이다.

6번 문제에 대한 "팀선정 프로그램 소스코드"

#6 C235357 전채운

Defining parent class 1

class Parent1 :

def __init__(self,a):

super().__init__()

self.p1 = a

Parent's show method

def show(self):

print("Inside Parent1",self.p1)

def display(self) :

print("Inside Parent1",self.p1)

Defining Parent class 2

class Parent2 :

def __init__(self,a,b):

super().__init__(a)

self.p2 = b

Parent's show method

def display(self):

print("Inside Parent2",self.p2)

Defining child class

class Child(Parent2, Parent1):

def __init__(self,a,b,c) :

```
super().__init__(a,b)
```

```
self.c = c
```

```
# Child's show method
```

```
def show(self):
```

```
    print("Inside Child",self.c)
```

```
# Driver's code
```

```
obj = Child(1,2,3)
```

```
obj.show()
```

```
obj.display()
```

비교 프로그램 소스코드

#6번 C235418 최민혁

```
# Defining parent class 1
```

```
class Parent1 :
```

```
    def __init__(self,a) :
```

```
        super().__init__()
```

```
        self.p1 = a
```

```
# Parent's show method
```

```
def show(self):
```

```
    print("Inside Parent1",self.p1)
```

```
def display(self) :
```

```
print("Inside Parent1",self.p1)
```

```
# Defining Parent class 2
```

```
class Parent2 :
```

```
    def __init__(self,a,b) :
```

```
        super().__init__(a)
```

```
        self.p2 = b
```

```
# Parent's show method
```

```
    def display(self):
```

```
        print("Inside Parent2",self.p2)
```

```
# Defining child class
```

```
class Child(Parent2, Parent1):
```

```
    def __init__(self,a,b,c) :
```

```
        super().__init__(a,b)
```

```
        self.c = c
```

```
# Child's show method
```

```
    def show(self):
```

```
        print("Inside Child",self.c)
```

```
# Driver's code
```

```
obj = Child(1,2,3)
```

```
obj.show()
```

```
obj.display()
```

비교분석: maturityAmount 메소드에서 선정된 프로그램은 값을 변수에 저장한 후에 변수를 리턴했고 비교 프로그램은 바로 값을 리턴했다. 이외에는 거의 유사한 코드이다.