

제6장 연습문제

[문제1]

다음과 같은 <처리조건>에 맞게 Rectangle 클래스를 작성하시오.

<처리조건>

1. 멤버변수 : 가로(width), 세로(height)
2. 생성자 : 가로(width), 세로(height) 멤버 변수 초기화
3. 메서드(area_calc) : 사각형의 넓이를 구하는 함수
사각형 넓이 = 가로 * 세로
4. 메서드(circum_calc) : 사각형의 둘레를 구하는 함수
사각형 둘레 = (가로 + 세로) * 2
5. 기타 세부내용은 <출력 결과 예시> 참조

<출력 결과 예시>

사각형의 넓이와 둘레를 계산합니다.

사각형의 가로 입력 : 10

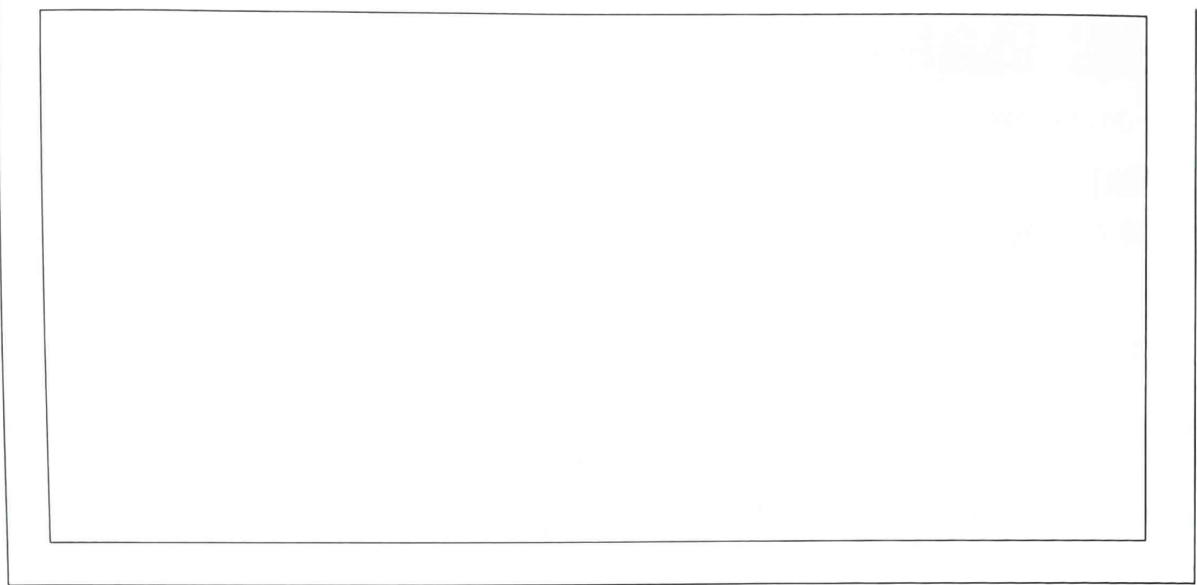
사각형의 세로 입력 : 5

사각형의 넓이 : 50

사각형의 둘레 : 30

<코딩 시작>

```
print("사각형의 넓이와 둘레를 계산합니다.")
w = int(input('사각형의 가로 입력 : '))
h = int(input('사각형의 세로 입력 : '))
```

**[문제2]**

동적 멤버 변수를 생성하여 다음과 같은 산포도를 구하는 클래스를 정의하시오.

< 출력 결과 예시 >

분산 : 7.4666666666666666

표준편차 : 2.7325202042558927

<코딩 시작>

```
from statistics import mean
from math import sqrt
```

```
x = [5, 9, 1, 7, 4, 6]
```

```
# 산포도 클래스
```

```
class Scattering:
```

```
    #생성자
```

```

class Person:
    def __init__(self, name, gender, age):
        self.name = name
        self.gender = gender
        self.age = age
    def display(self):
        print(f"이름: {self.name}, 성별: {self.gender}, 나이: {self.age}")

```

#메서드 : 분산 (var_func)

```

class Person:
    def __init__(self, name, gender, age):
        self.name = name
        self.gender = gender
        self.age = age
    def display(self):
        print(f"이름: {self.name}, 성별: {self.gender}, 나이: {self.age}")

```

#메서드 : 표준편차 (std_func)

```

class Person:
    def __init__(self, name, gender, age):
        self.name = name
        self.gender = gender
        self.age = age
    def display(self):
        print(f"이름: {self.name}, 성별: {self.gender}, 나이: {self.age}")

```

[문제3]

다음과 같은 <처리조건>에 맞게 Person 클래스를 작성하시오.

<처리조건>

1. 멤버 변수 : 이름(name), 성별(gender), 나이(age)
2. 생성자 : 이름, 성별, 나이 초기화
3. 메서드 : display(이름, 성별, 나이 출력 기능)
4. 기타 세부내용은 <출력 결과 예시> 참조

〈출력 결과 예시〉

```

이름 입력 : 유관순
나이 입력 : 35
성별(male/female) 입력 : female
=====
이름 : 유관순, 성별 : 여자
나이 : 35
=====

```

〈코딩 시작〉

```
class Person :
```

```
    #생성자
```

```
    #메서드 (display)
```

```
# 키보드 입력
```

```
name = input('이름 : ')
age = int(input('나이 : '))
gender = input('성별(male/female) : ')

```

```
# 객체 생성
```

```
p = Person(age, name, gender) # 생성자 이용 전역변수 초기화
```

[문제4]

다음과 같은 <처리조건>에 맞게 Employee 클래스를 상속받아서 Permanent와 Temporary 클래스를 구현하시오.

<처리 조건>

1. 키보드로 정규직과 임시직을 구분한다.
2. 정규직인 경우에는 기본급과 상여금을 입력 받아서 급여를 계산한다.
3. 임시직인 경우에는 작업시간과 시급을 입력 받아서 급여를 계산한다.
4. 기타 세부내용은 <출력 결과 예시> 참고

<출력 결과 예시>

```
고용형태 선택(정규직<P>, 임시직<T>) : >? P
```

```
이름 : >? 홍길동
```

```
기본급 : >? 2000000
```

```
상여금 : >? 500000
```

```
=====
```

```
고용형태 : 정규직
```

```
이름 : 홍길동
```

```
급여 : 2,500,000
```

```
고용형태 선택(정규직<P>, 임시직<T>) : >? T
```

```
이름 : >? 김길동
```

```
작업시간 : >? 200
```

```
시급 : >? 12000
```

```
=====
```

```
고용형태 : 임시직
```

```
이름 : 김길동
```

```
급여 : 2,400,000
```

[코딩 시작]

부모클래스

```
class Employee :  
    name = None  
    pay = 0  
  
    def __init__(self, name):  
        self.name = name
```

자식클래스 - 정규직

```
class Permanent(Employee):
```

자식클래스 - 임시직

```
class Temporary(Employee):
```

```
empType = input("고용형태 선택 (정규직<P>, 임시직<T>) : ")
```

```
if empType == 'P' or empType == 'p' :
```

```
elif empType == 'T' or empType == 't' :
```

```

else :
    print('='*30)
    print('입력 오류')

```

[문제5]

다음과 같은 <처리조건>에 맞게 사칙연산 관련 패키지와 모듈을 작성하고, 다른 모듈에서 import하여 결과를 확인하시오.

<처리조건>

1. 패키지명 : myCalcPackage
2. 모듈명 : calcModule.py
3. 함수명 : Add(), Sub(), Mul(), Div()
4. 호출 모듈명 : example.py
5. 호출 모듈에서 사칙연산 함수를 호출한 결과 : <출력 결과 예시> 참고

<출력 결과 예시>

```

x = 10; y = 5 일 때
Add= 15
Sub= 5
Mul= 50
Div= 2.0

```