

# [Workshop #3]

오늘 배운 내용을 이용하여 명세에 맞는 프로그램을 작성한다.

❖ 프로젝트명

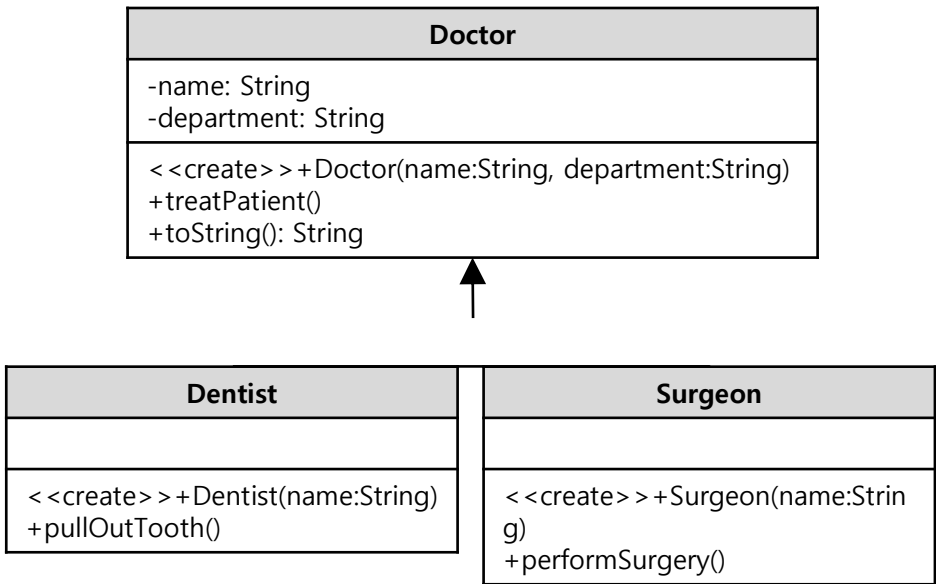
WorkShop3

❖ Workshop #3-1

상속 구조를 이용하여 치과 의사와 외과 의사 정보를 출력하려고 한다. 제공된 코드와 아래의 Class Diagram 을 이용하여 Sample Run과 같이 출력되도록 프로그램을 작성해 보자.

- 작성할 클래스명 : com.test.workshop3.doctor.Doctor / Dentist / Surgeon / DoctorTest

- Class Diagram



- Sample Run

치과 : 홍길동 선생님  
환자를 진료합니다.  
이를 뽑습니다.

치과 : 전우치 선생님  
환자를 진료합니다.  
이를 뽑습니다.

Sample Run

외과 : 임격정 선생님  
환자를 진료합니다.  
수술을 집도하다.

외과 : 홍길순 선생님  
환자를 진료합니다.  
수술을 집도하다.

외과 : 광개토 선생님  
환자를 진료합니다.  
수술을 집도하다.

## ❖ Workshop #3-2

arrayOne, arrayTwo 두 개의 배열이 있다. 이 두 개의 배열은 정수를 저장하고 있으며, 서로 크기가 다르다. 두 개의 배열의 크기를 비교하여 더 큰 배열의 수만큼 for 반복문을 작성해 보자.

반복문을 수행하면서 arrayOne에 있는 숫자와 arrayTwo에 있는 숫자를 나눴셈 하도록 한다. 서로 같은 위치 (같은 인덱스)

에 있는 숫자끼리 나눴셈을 수행하고, 그 결과를 출력한다. 그리고 발생할 수 있는 에러 상황에 대해서 예외 처리를 통해 프로그램이 비정상적으로 종료되지 않도록 해 보자.

- 작성할 클래스명 : com.test.workshop3.ArrayDivideTest

- Sample Run

10 / 25

20 / 54

나누기 연산 시 에러가 발생하였습니다.

25 / 38

인덱스 범위를 벗어나는 값에 접근하였습니다.

## ❖ Workshop #3-3

계좌 정보를 생성하고, 입금/출금하는 프로그램을 작성해 본다. 단, 여러 가지 예외 상황이 발생했을 때 Exception을 강제로

발생시켜 정상적인 상황이 아님을 프로그램에게 알리고, 적절한 처리를 하도록 한다.

이번 프로그램은 메뉴를 입력하여 그 메뉴에 따른 처리를 하도록 한다. 메뉴는 9번을 입력하기 전까지 반복하여 프로그램을 실행시킬 수 있도록 한다.

- 메뉴 1. 계좌 생성 : 사용자로부터 계좌번호, 예금주, 최초 입금액을 입력 받아 계좌를 생성한다. 최초 입금액은 반드시 0원을 초과하는 금액이어야 하며, 0원 이하의 금액이 들어온 경우 `InvalidValueException`을 발생시키고, 적절한 에러 메시지를 넘겨준다.
- 메뉴 2. 계좌 정보 출력 : 계좌의 정보를 출력한다. 만약, 계좌가 아직 생성되지 않은 상태라면, `AccountNotFoundException`을 발생시킨다.
- 메뉴 3. 입금 : 입금할 금액을 입력 받고, 계좌에 입금한다. 계좌가 아직 생성되지 않은 상태라면 메뉴 2와 같은 예외를 발생시키도록 하며, 입금할 금액을 0원 이하로 입력한 경우에는 `InvalidValueException`을 발생시킨다.
- 메뉴 4. 출금 : 입금과 마찬가지로이다. 단, 잔액이 부족한 경우에도 마찬가지로 `InvalidValueException`을 발생시킨다.
- 메뉴 9. 종료 : 프로그램을 종료한다.
- 메뉴 입력 시, 문자 입력 : `NumberFormatException`이 발생하며, 프로그램이 비정상 종료되지 않도록 예외 처리 한다.
- 메뉴 1, 2, 3, 4, 9 외에 다른 번호 입력 : Sample Run을 참고하여 적절한 에러 메시지를 출력한다.

다음 장의 Class Diagram과 Sample Run을 참고하여 프로그램을 완성해 보자.

## ❖ Workshop #3-3 (계속)

- 작성할 클래스명 : com.test.workshop3.account.Account / AccountTest
- 참고 클래스 : com.test.workshop3.exception.AccountNotFoundException, InvalidValueException
- Class Diagram

Account	AccountTest
<div>-account: String -name: String -balance: int</div> <div>&lt;&lt;create&gt;&gt;+Account(account:String, name:String, balance:int) +deposit(money:int) +withdraw(money:int) +toString(): String</div>	<div>+main(args:String[]) -printMenu() +getUserInput(): String</div>

- Sample Run

===== < 메뉴 > =====

1. 계좌 생성
2. 계좌 정보 출력
3. 입금
4. 출금
9. 종료

=====

>> 메뉴 : 2

[에러] 계좌가 없습니다.

===== < 메뉴 > =====

1. 계좌 생성
2. 계좌 정보 출력
3. 입금
4. 출금
9. 종료

=====

>> 메뉴 : 1

계좌번호 : 123-4567-890

예금주 : 홍길동

최초 입금 : 0

[에러] 계좌 생성 시, 최초 입금 금액이 필요합니다.

===== < 메뉴 > =====

...

=====

>> 메뉴 : 2

[에러] 계좌가 없습니다.

## ❖ Workshop #3-3 (계속)

- Sample Run

```
===== < 메뉴 > =====
```

1. 계좌 생성
2. 계좌 정보 출력
3. 입금
4. 출금
9. 종료

```
=====
```

```
>> 메뉴 : 1
```

계좌번호 : 123-4567-890

예금주 : 홍길동

최초 입금 : 10000

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 2
```

[계좌] 123-4567-890      [예금주] 홍길동 [잔액] 10000

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 3
```

입금할 금액 : -5000

[에러] 입금할 금액은 0원 초과인 숫자로 입력하세요.

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 3
```

입금할 금액 : 10000

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 2
```

[계좌] 123-4567-890      [예금주] 홍길동 [잔액] 20000

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 4
```

출금할 금액 : -5000

[에러] 출금할 금액은 0원 초과인 숫자로 입력하세요.

## ❖ Workshop #3-3 (계속)

- Sample Run

```
===== < 메뉴 > =====
```

1. 계좌 생성
2. 계좌 정보 출력
3. 입금
4. 출금
9. 종료

```
=====
```

```
>> 메뉴 : 4
```

```
출금할 금액 : 30000
```

```
[에러] 잔액이 부족합니다.
```

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 4
```

```
출금할 금액 : 15000
```

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 2
```

```
[계좌] 123-4567-890      [예금주] 홍길동 [잔액] 5000
```

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : 5
```

```
[에러] 메뉴를 잘못 입력하였습니다.
```

```
===== < 메뉴 > =====
```

```
...
```

```
=====
```

```
>> 메뉴 : a
```

```
[에러] 메뉴는 숫자로 입력해야 합니다.
```

```
===== < 메뉴 > =====
```

1. 계좌 생성
2. 계좌 정보 출력
3. 입금
4. 출금
9. 종료

```
=====
```

```
>> 메뉴 : 9
```

```
종료합니다.
```