

클래스 생성자

다음 문제를 풀면서 Member 클래스를 완성하여라.

1. Member 클래스는 필드로 이름, 아이디, 비밀번호를 갖는다. 모든 필드를 선언하고, 기본 생성자와 클래스의 모든 데이터를 변경할 수 있는 생성자를 만드시오.
2. 각 필드에 대한 setter와 getter를 만들어라.
3. 모든 필드의 값을 출력하는 displayInfo() 메소드를 만들어라.
4. 실행 클래스를 만들어 이름을 '자바', 아이디는 'java', 비밀번호는 '123'으로 변경하고 출력메소드를 사용하여 결과를 확인하여라.

5.

13. 현실 세계의 회원을 Member 클래스로 모델링하려고 합니다. 회원의 데이터로는 이름, 아이디, 패스워드, 나이가 있습니다. 이 데이터들을 가지는 Member 클래스를 선언해보세요.

데이터 이름	필드 이름	타입
이름	name	문자열
아이디	id	문자열
패스워드	password	문자열
나이	age	정수

【 Member.java 】

```
01 public class Member {  
02     //작성 위치  
03 }
```

14. 위에서 작성한 Member 클래스에 생성자를 추가하려고 합니다. 다음과 같이 Member 객체를 생성할 때 name 필드와 id 필드를 외부에서 받은 값으로 초기화하려면 생성자를 어떻게 선언해야 할까요?

```
Member user1 = new Member("홍길동", "hong");
Member user2 = new Member("강자바", "java");
```

【 Member.java 】

```
01 public class Member {
02     //작성 위치
03 }
```

15. MemberService 클래스에 login() 메소드와 logout() 메소드를 선언하려고 합니다. login() 메소드를 호출할 때에는 매개값으로 id와 password를 제공하고, logout() 메소드는 id만 매개값으로 제공합니다. MemberService 클래스와 login(), logout() 메소드를 선언해보세요.

- ❶ login() 메소드는 매개값 id가 "hong", 매개값 password가 "12345"일 경우에만 true로 리턴하고 그 이외의 값일 경우에는 false를 리턴하도록 하세요.
- ❷ logout() 메소드의 내용은 "로그아웃 되었습니다."가 출력되도록 하세요.

리턴 타입	메소드 이름	매개 변수(타입)
boolean	login	id(String), password(String)
void	logout	id(String)

【 MemberService.java 】

```
01 public class MemberService {
02     //작성 위치
03 }
```

【 MemberServiceExample.java 】

```
01 public class MemberServiceExample {
02     public static void main(String[] args) {
03         MemberService memberService = new MemberService();
04         boolean result = memberService.login("hong", "12345");
05         if(result) {
```

```

06      System.out.println("로그인 되었습니다.");
07      memberService.logout("hong");
08  } else {
09      System.out.println("id 또는 password가 올바르지 않습니다.");
10  }
11  }
12  }

```

실행결과

```

Console
<terminated> MemberSe
로그인 되었습니다.
로그아웃 되었습니다.

```

6.

다음과 같은 멤버를 가지는 직사각형을 표현하는 Rectangle 클래스를 작성하라.

- int 타입의 x1, y1, x2, y2 필드 : 사각형을 구성하는 두 점의 좌표
- 생성자 2개 : 디폴트 생성자와 x1, y1, x2, y2의 값을 설정하는 생성자
- void set(int x1, int y1, int x2, int y2) : x1, y1, x2, y2 좌표 설정
- int square() : 사각형 넓이 리턴
- void show() : 좌표와 넓이 등 직사각형 정보의 화면 출력
- boolean equals(Rectangle r) : 인자로 전달된 객체 r과 현 객체가 동일한 직사각형이면 true 리턴

Rectangle을 이용한 main() 메소드는 다음과 같으며 이 main() 메소드가 잘 작동하도록 하라.

```

public static void main(String args[]) {
    Rectangle r = new Rectangle();
    Rectangle s = new Rectangle(1,1,2,3);

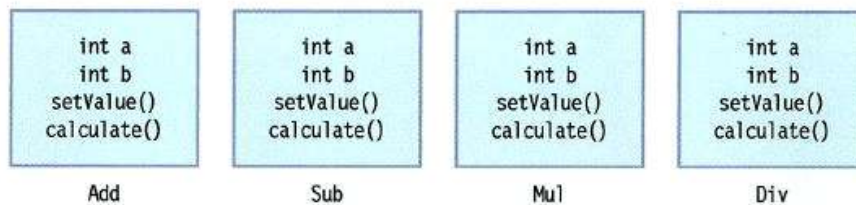
    r.show();
    s.show();
    System.out.println(s.square());
    r.set(-2,2,-1,4);
    r.show();
    System.out.println(r.square());
    if(r.equals(s))
        System.out.println("두 사각형은 같습니다.");
}

```

7.

다수의 클래스를 정의하고 활용하는 연습을 해보자. 더하기(+), 빼기(-), 곱하기(*), 나누기(/)를 수행하는 각 클래스 Add, Sub, Mul, Div를 만들어라. 이들은 모두 다음 필드와 메소드를 가진다.

- int 타입의 a, b 필드 : 연산하고자 하는 피연산자
- void setValue(int a, int b) : 피연산자를 객체 내에 설정한다.
- int calculate() : 해당 클래스의 목적에 맞는 연산을 실행하고 그 결과를 리턴한다.



main() 메소드에서는 다음 실행 사례의 그림과 같이 키보드로부터 두 정수와 계산하고자 하는 연산자를 입력받은 후 Add, Sub, Mul, Div 중에서 이 연산을 실행할 수 있는 객체를 생성하고 setValue()와 calculate()를 호출하여 그 결과 값을 화면에 출력한다.

-실행예시-

첫 번째 수 : 10

두 번째 수 : 20

연산자 : +

10 + 20 = 30