

2020-2

# 플랫폼기반프로그래밍

## Lab1

Week3

Classes I - Basic

담당교수 : 정은미



부산대학교  
PUSAN NATIONAL UNIVERSITY

## 실습 목표

1. 객체 지향 개념을 학습한다.
2. class, instance variable, method를 구현해 본다.
3. static 멤버와 초기화 과정에 대해 학습한다.

## 참고.

### 객체지향프로그램 개념

#### 1. Object

- 객체 지향 프로그래밍(Objected-Oriented programming)은 객체(Object)개념을 기본으로 하는 프로그래밍 패러다임이다.
- 객체는 특정한 상태(attribute 혹은 property)을 가지고, 특정한 행동(method)을 한다.  
TV를 객체의 관점에서 보았을 때 property와 method에서



TV
속성(상태) 전원상태, 볼륨, 채널
행동 ON/OFF, 채널변경, 볼륨변경

#### 2. Class

- 클래스는 이러한 객체가 어떠한 형태로 정의되는지를 나타내는 템플릿(설계도, 틀)이다.

TV
속성(상태) 전원상태, 볼륨, 채널
행동 ON/OFF, 채널변경, 볼륨변경



Class
- power : boolean - volume : int - channel : int +Power(): void +changChannel(int n) +volumeUp(); +volumeDown();

#### 3. Example

##### 3.1 class

```
public class Tv {  
    private boolean power;  
    private int channel;  
    private int volume;  
    public Tv()  
    {  
        power=false;  
        channel=1;  
        volume=10;  
    }  
    public void Power()  
    {  
        power=!power;  
    }  
    public void changeChannel(int n)  
    {  
        channel=n;  
    }  
    public void volumeUp()  
    {  
    }  
}
```

```

        volume++;
    }
    public void volumeDown()
    {
        volume--;
    }
    public int getVolume()
    {
        return volume;
    }
    public void setVolume(int volume)
    {
        this.volume=volume;
    }
}

```

### - Constructors

생성자는 오브젝트 생성시 호출된다.

생성자의 이름은 클래스명과 동일해야 한다.

생성자는 여러 인자를 가질 수 있다.

생성자를 정의하지 않더라도, 자바 컴파일러가 기본 생성자를 만들어준다.

### - Getter/ Setter

Getter는 get필드명()로 관례적으로 구현하며, 외부에서 필드(private 멤버변수)의 값이 필요할 때 필드값을 리턴해줌

Setter는 set필드명(인수)으로 관례적으로 구현하며, 외부에서 필드(private 멤버변수)의 값을 변경이 필요시 호출

```

public int getVolume()
{
    return volume;
}
public void setVolume(int volume)
{
    this.volume=volume;
}

```

## 3.2 Creating Object

객체 생성 키워드

Tv tv1
= new Tv()
;

객체변수선언
생성자 호출

## 4. UML 표기법

### Class Diagram

클래스명
속성
메소드

- 클래스를 표현
- 객체의 속성(필드)과 행동(메소드)을 포함.
- 접근제어 표현법

Visibility(접근 제어)	
+	public
-	private
#	protection
~	package

- 속성, 행동 공통사항 : static는 밑줄로 표현

## Lab1. Mini 도서관 관리 프로그램.

### Book 클래스

- 클래스 다이어그램과 수행결과, BookTest클래스를 참고하여 완성하시오.

Book
-author : String -title : String -id : int <u>-indetification</u> : int <u>+bookCount</u> : int
+ Book() + Book(title:String) + Book(title:String, author:String) <u>+ decrease():void</u> + setTitle(title:String):void + setAuthor(author:String):void + getTitle():String + getAuthor():String + getId():int + equals(obj:Object):boolean + toString():String

### 조건 :

클래스 초기화 블록을 이용하여 indetification(id로 사용), bookCount(권수)를 초기화

- indetification : 0~10000사이의 난수 값
- bookCount : 0

인스턴스 초기화 블록을 필드값 초기화

- indetification 값 1증가
- bookCount값 1증가
- id의 값을 indetification으로 초기화

equals() : 객체 비교, 책제목과 저자가 같으면 true, 그렇지 않으면 false

toString() : 수행결과를 참고하여 작성

▶ 프로그램 수행결과

Enter Operation String!(ADD, LIST, FIND, QUIT)add

-----  
Type the title of the book:

부활

Type the name of thd author:

톨스토이

Book No: 4067, entitled "부활", written by 톨스토이  
책이 추가 되었습니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)add

-----  
Type the title of the book:

어린왕자

Type the name of thd author:

생텍쥐페리

Book No: 4068, entitled "어린왕자", written by 생텍쥐페리  
책이 추가 되었습니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)add

-----  
Type the title of the book:

노인과 바다

Type the name of thd author:

헤밍웨이

Book No: 4069, entitled "노인과 바다", written by 헤밍웨이  
책이 추가 되었습니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)list

-----  
Book List

-----  
Book No: 4067, entitled "부활", written by 톨스토이  
Book No: 4068, entitled "어린왕자", written by 생텍쥐페리  
Book No: 4069, entitled "노인과 바다", written by 헤밍웨이  
-----

총 3권

Enter Operation String!(ADD, LIST, FIND, QUIT)find

-----  
Type the title of the book:

부활

Type the name of thd author:

톨스토이

해당하는 책이 도서관에 존재합니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)find

-----  
Type the title of the book:

이방인

Type the name of thd author:

알베르 까뮈

찾는 책이 없습니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)list

-----  
Book List

-----  
Book No: 4067, entitled "부활", written by 톨스토이  
Book No: 4068, entitled "어린왕자", written by 생텍쥐페리  
Book No: 4069, entitled "노인과 바다", written by 헤밍웨이  
-----

총 3권

Enter Operation String!(ADD, LIST, FIND, QUIT)quit

-----  
현재 도서관 책 보유량 : 3권

도서관 보유 책 리스트

-----  
Book No: 4067, entitled "부활", written by 톨스토이  
Book No: 4068, entitled "어린왕자", written by 생텍쥐페리  
Book No: 4069, entitled "노인과 바다", written by 헤밍웨이  
-----

## ▶ BookTest 클래스

- 수행 결과를 참고하여 다음 코드를 확장하여 프로그램을 작성하시오.
  - 단, 주어진 코드를 수정/제거하지 마시고, 추가하여 사용
  - 책은 최대 5권까지만 추가할 수 있음.
  - 메뉴 선택은 대소문자를 구분하지 않아도 됨
- ADD : 책제목과 저자를 입력받아 책을 추가  
LIST : 현재까지 추가된 책 목록을 보여줌  
FIND : 제목과 저자를 입력받아 존재여부 출력
- aBook객체를 생성하여 Bookes[]객체 안에 있는지 비교  
aBook객체를 생성하면 id가 1증가하고, 책의 수도 1증가됨.  
decrease() 메소드를 이용해 id와 책의 수가 증가되지 않도록 함.

```
Enter Operation String!(ADD, LIST, FIND, QUIT)Add
```

```
-----  
Type the title of the book:
```

이방인

```
Type the name of thd author:
```

알베르 까뮈

Book No: 1192, entitled "이방인", written by 알베르 까뮈  
책이 추가 되었습니다.

```
Enter Operation String!(ADD, LIST, FIND, QUIT)Add
```

```
-----  
Type the title of the book:
```

변신

```
Type the name of thd author:
```

카프카

Book No: 1193, entitled "변신", written by 카프카  
책이 추가 되었습니다.

```
Enter Operation String!(ADD, LIST, FIND, QUIT)find
```

```
-----  
Type the title of the book:
```

변신

```
Type the name of thd author:
```

카프카

해당하는 책이 도서관에 존재합니다.

```
Enter Operation String!(ADD, LIST, FIND, QUIT)list
```

```
-----  
Book List
```

```
-----  
Book No: 1192, entitled "이방인", written by 알베르 까뮈
```

```
Book No: 1193, entitled "변신", written by 카프카
```

```
-----  
총 3권
```

← 실제 책은 2권이지만 책을 검색하기 위해 만든 객체로 인해 책 수량이 증가함.

```
Enter Operation String!(ADD, LIST, FIND, QUIT)
```

QUIT : 책 추가 루프를 빠져나옴

- 루프를 빠져 나온 후 현재까지 추가된 책 목록과 총 보유 수량을 출력

```
-----  
현재 도서관 책 보유량 : 3권  
도서관 보유 책 리스트
```

```
-----  
Book No: 4067, entitled "부활", written by 톨스토이
```

```
Book No: 4068, entitled "어린왕자", written by 생텍쥐페리
```

```
Book No: 4069, entitled "노인과 바다", written by 헤밍웨이
```

- ▶ Getter/Setter Test
  - BookTest 클래스에 코드 추가

보리스

Book No: 1871, entitled "닥터 지바고", written by 보리스  
책이 추가 되었습니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)add

-----  
Type the title of the book:

바람과 함께 사라지다

Type the name of thd author:

마가렛 미첼

Book No: 1872, entitled "바람과 함께 사라지다", written by 마가렛 미첼  
책이 추가 되었습니다.

Enter Operation String!(ADD, LIST, FIND, QUIT)quit

-----  
현재 도서관 책 보유량 : 2권

도서관 보유 책 리스트

-----  
Book No: 1871, entitled "닥터 지바고", written by 보리스

Book No: 1872, entitled "바람과 함께 사라지다", written by 마가렛 미첼

-----  
정보수정을 원하는 책 id를 입력하세요:

1871

책이름 : 닥터 지바고

저자 : 보리스

수정할 부분을 선택(1: title, 2: author) : 2

저자입력보리스 파스테르나크

Book No: 1871, entitled "닥터 지바고", written by 보리스 파스테르나크 <--수정됨.





```
        System.out.println("해당하는 책이 도서관에 존재합니다.");  
        find=true;  
        break;  
    }  
    if(find==false)  
        System.out.println("찾는 책이 없습니다.");  
}  
} //switch  
} //for
```

/\*책 목록과 책 보유수량을 출력하는 코드를 추가\*/

/\*Getter/ Setter를 이용하여 책 정보를 수정하시오.\*/

```
}
```

```
}
```



