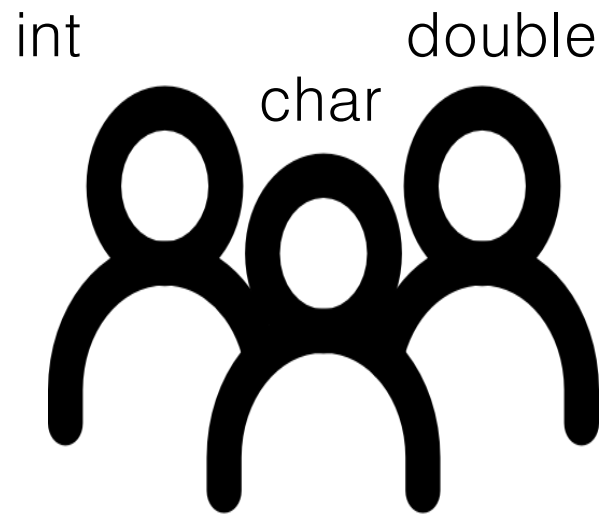


spring 한스푼(1주차)

string/클래스&객체/상속

미리보기(string)



기본 라이브러리

string



string vs char

- char은 좌표로 문자를 저장하지만 string은 문자열 자체로 하나다.

ex) hello,world!

- char은 선언한 순간 제한이 생기지만 string은 제한이 없다.

string 사용법

```
System.out.println(" hello world! ");
```

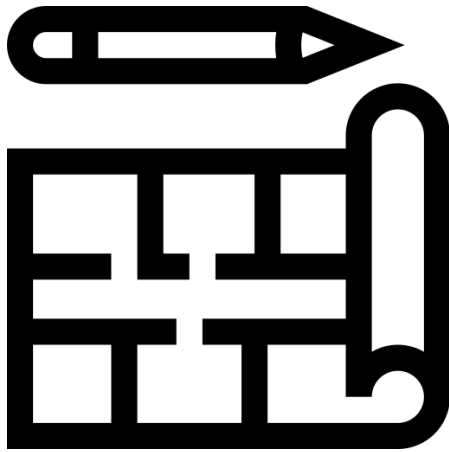
```
String name;
```

```
name = "박성수";
```

```
System.out.println(name);
```

미리보기(클래스&객체)

클래스(설계도)



객체(집)



클래스? 객체?

클래스(class)

- 객체의 속성과 기능을 코드로 구현한 것. 선언한 것
- “클래스를 정의한다”라고 함

• 객체(object)

- 프로그램 실행중에 생성되는 실체. 메모리 공간을 갖는 구체적인 실체
- 인스턴스라고도 부름

클래스(코드) \Rightarrow 인스턴스(힙메모리)

접근 제어자 키워드 클래스 이름

```
public class Car {
```

클래스 필드 → `private String modelName;`
`private int modelYear;` ← 세미 콜론

...

클래스 → public void brake() { }

...

}

클래스의
멤버

직접 해보기!

Tv라는 클래스의 멤버 변수는 음량과 전원 버튼이 있다.

음량 기본 10으로 설정

up() 함수에 음량을 올리고, down() 함수에 음량을 낮추는 코드와

button() 함수에 false면 “TV 끄기”, true면 “TV 켜기”

상태를 모두 볼 수 있는 함수까지 코드를 만들어보자.

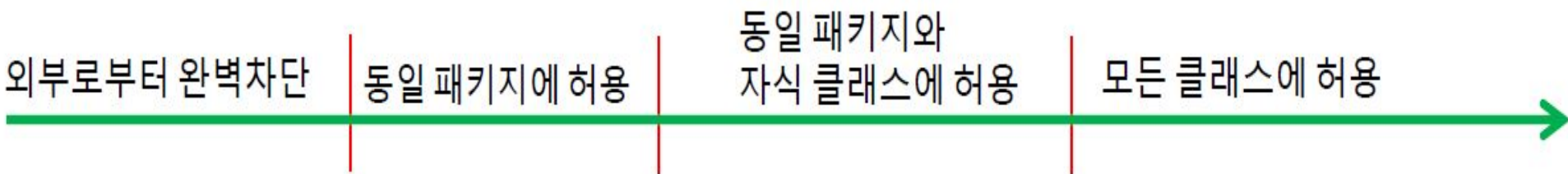
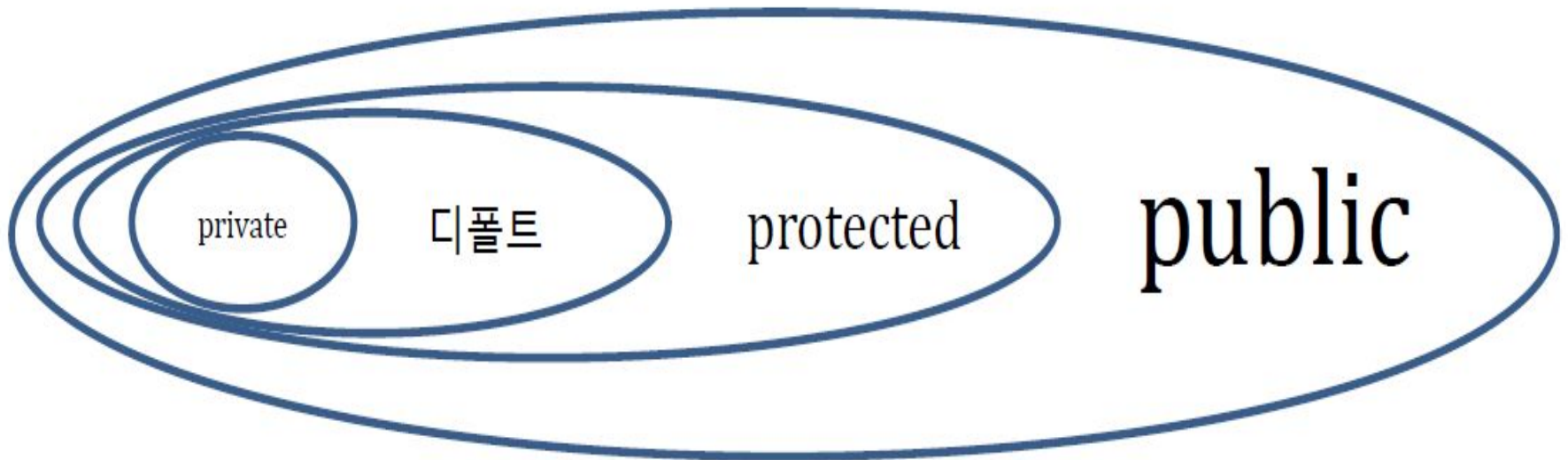
접근 제어자?

디폴트 - 같은 패키지 내에 클래스 접근 가능

public - 프로젝트 내에 클래스 접근 가능

private - 클래스 내에만 접근 가능.

protected - 같은 패키지와 자식 클래스 접근 가능



접근 제어자 왜 필요한데?

- 객체를 캡슐화하여, 그 내부를 보호하고 볼 수 없게 하기 위함.

- 객체 보호, 정보 은닉



접근이 제한된 클래스 정보 변경

- 디폴트 생성자

- +

- this

- getter

- setter

```
package constructor;
```

```
public class Person {
```

```
    String name;
```

```
    float height;
```

```
    float weight;
```

```
    public Person( ) { }
```

디폴트 생성자

```
    public Person(String pname) {
```

```
        name = pname;
```

```
    }
```

이름을 매개변수로 입력받는 생성자

```
    public Person(String pname, float pheight, float pweight) {
```

```
        name = pname;
```

```
        height = pheight;
```

```
        weight = pweight;
```

```
    }
```

이름, 키, 몸무게를
매개변수로 입력
받는 생성자

```
}
```

this

- 클래스 내에 멤버를 가리킬 때

this.멤버변수이름

- 클래스 내에 함수를 가리킬 때

this()

```
package constructor;
```

```
public class Person {  
    String name;  
    float height;  
    float weight;
```

```
public Person(){ this("홍길동");}
```

```
public Person(String name){ this(name, 0, 0);}
```

```
public Person(String name, float height, float weight){  
    this.name = name;  
    this.height = height;  
    this.weight = weight;  
}
```

```
}
```

getter와 setter

- getter()를 통해 멤버 변수를 가져올 수 있다.

```
public String getName(){ return name;}
```

- setter()를 통해 멤버 변수를 수정 할 수 있다.

```
public void setName(String name){  
    this.name= name;}
```


직접 해보기!

Smart_TV라는 클래스가 있다.

클래스 멤버는 brand, price, inch가 있다. 이 멤버들은 숨겨져있다.

한 객체는 매개변수 생성자로 LG, 900000, 40으로 만들고

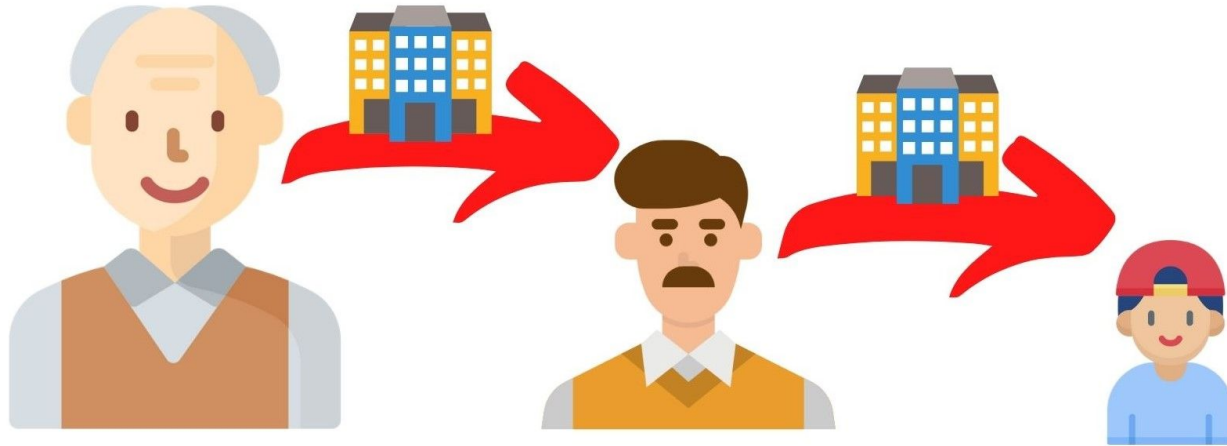
한 객체는 setter로 SAMSUNG, 1100000, 70으로 만들어라.

print()로 각각 나타내보자.

브랜드 : LG | 가격 : 900000 | 인치 : 40

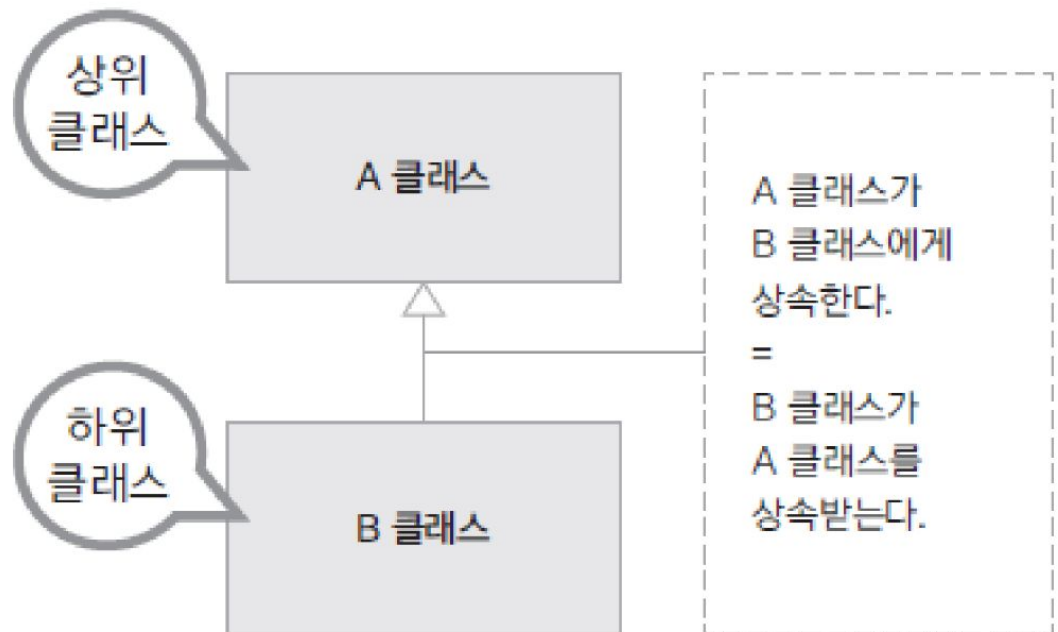
브랜드 : SAMSUNG | 가격 : 1100000 | 인치 : 70

미리보기(상속)



상속이란?

- 자식(클래스)이 상속받고 싶은 부모(클래스)를 선택해서 물려 받는 것.
- 상속 받은 클래스를 **자식 클래스**, 상속을 해주는 클래스를 **부모 클래스**



상속이 필요한 이유!

- 중복된 코드를 줄일 수 있다.
- 유지 보수↑(오류를 쉽게 찾고, 고치기 쉬움)
- 통일성이 있다.
- 다형성을 구현할 수 있다.(멤버십 프로그램)

상속 선언

<pre>class (자식)클래스 extends (부모)클래스, (부모)클래스2 { .. }</pre>	불가
<pre>class (자식)클래스1 extends (부모)클래스 { .. } class (자식)클래스2 extends (부모)클래스 { .. }</pre>	가능
<pre>class (부모)클래스 extends (조부모)클래스 { .. } class (자식)클래스 extends (부모)클래스 { .. }</pre>	가능

직접 해보기!

상속을 활용하여 고객관리 프로그램을 만들어보자!

customerName - 고객 이름

customerGrade - 고객 등급 (생성시 **SILVER**)

bonusPoint - 고객의 보너스 포인트

bonusRatio - 보너스 포인트 적립 비율

VIP는 **salesRatio** 추가(할인 비율)

