



객체지향프로그래밍 7주차

제출일	2023 년 10 월 23 일	학과	컴퓨터공학과
과목	객체지향프로그래밍 01	학번	2022112701
담당교수	한인 교수님	이름	안소희

□ 실습문제 1

□ 문제분석

상속을 이용하여 문제를 해결해야한다. TV에는 size라는 멤버 변수가 있고 이러한 클래스를 ColorTV에 상속시킨다. 하지만 멤버 변수는 private로 접근 지정되어있으므로 public 함수를 통해 접근해야한다. 이런 방법으로 변수 값을 할당하고 출력해야한다.

□ 프로그램 설계 및 알고리즘

TV 클래스 생성 후 상속을 이용해 ColorTV 클래스를 만든 후 메인 함수가 있는 ColorEx 클래스에서 실행한다. 이때 필드에 직접 접근할 수는 없으며 Set 함수를 이용해 접근하여 값을 바꾸어야한다. ColorTV의 생성자는 TV의 생성자에 접근하기 위해 super()메소드를 사용한다. 그리고 직접 size에 접근할 수 없으므로 ColorTV 클래스에서 TV 클래스의 getSize()메서드를 이용해 size를 받은 뒤 자신의 필드인 Color와 출력한다.

□ 소스코드 및 주석

(week7_1)

```
package week7_1;

public class TV {
    private int size;
    public TV() { //기본 생성자
        size = 0;
    }

    public void setSize(int size) { //size를 입력받는 메소드
        this.size = size;
    }

    protected int getSize() { //size를 반환하는 메소드
        return size;
    }
}
```

```

package week7_1;

public class ColorTV extends TV{
    private int color;

    public ColorTV() { //기본생성자
        super();
        color = 0;
    }

    public void setColor(int color) { //color 이 private 이므로
        //다른 클래스에서 이 함수를 이용해 값을 변경시킨다.
        this.color = color;
    }

    public void printProperty() { //정보를 출력하는 함수
        int size = getSize(); //size 가 private 이므로 슈퍼
        //클래스이더라도 접근할 수 없으므로 getSize() 함수 이용해 값을 받는다.
        System.out.println(size+"인치 "+color+"컬러 "); //출력
    }
}

package week7_1;

public class ColorTVEx {

    public static void main(String[] args) {

        ColorTV myTV = new ColorTV(); //객체에 대한 레퍼런스 선언
        //후 할당
        myTV.setSize(32); //size 의 값 32 로 할당
        myTV.setColor(1024); //color 의 값 1024 로 할당
        myTV.printProperty(); //결과 출력

    }

}

```

□ 결과 및 결과 분석

결과 화면	
	32인치 1024컬러
결과 분석	
<p>ColorTV는 TV를 상속받은 클래스이므로 setSize()메소드를 따로 선언하지 않고 사용이 가능하다. size는 TV 클래스에서 private로 접근 지정자가 설정되어 있으므로 ColorTV.size로 접근을 못하기 때문에 public 함수인 setSize()함수를 이용해 접근하여 32을 할당한다. 이와 같이 setColor()함수를 이용해 private인 color를 간접적으로 접근하여 1024을 할당한다. printProperty()를 통해 정보를 출력하는데 private인 size는 서브클래스조차도 접근을 못하기 때문에 getSize()함수를 이용해 size값을 반환한 뒤 이를 color와 함께 출력하여 32인치 1024컬러가 출력되었다.</p>	

□ 소감

상속에 대해 배워 간단한 예제를 통해 작성해보았는데 super()라는 메소드가 아직 어색하였다. 그리고 클래스의 필드는 private로 접근지정을 하는 게 좋다는 것을 알게 되었고 이에 접근하기 위해 public 함수를 이용해야 한다는 것을 배웠다.

□ 실습문제 2

□ 문제분석

Member(일반 동아리원)클래스를 이용하여 학번, 이름을 선언한 뒤 Crew(집행부원)은 Member 의 상속을 받아 학번, 이름을 따로 선언하지 않고 추가 기능인 부서를 선언한다. 일반 동아리원들과 크루원들 수는 여러 명이 가능하기 때문에 객체의 배열을 이용한다. 그 후 원하는 수만큼 각각을 입력 받고 정보를 출력한다.

□ 프로그램 설계 및 알고리즘

Member 클래스에 학번과 이름을 선언해주고 출력함수와 학번을 반환하는 함수를 선언한다. Crew 클래스는 Member 클래스를 상속받아 따로 학번과 이름을 선언하지 않고 super()함수를 통해 서브클래스에서 슈퍼클래스의 생성자를 호출한다. 그리고 추가기능으로 부서를 선언하고 생성할 때 이를 같이 생성하도록 한다. 또한 출력함수를 선언한다. 메인 함수에서 일반 부원의 수와 집행부원의 수를 입력 받은 후 그 수만큼 객체 배열을 생성한다. 그 후 먼저 일반 부원의 학번을 입력 받는다. 만약 이전에 입력 받은 학번과 같다면 이미 존재한다는 에러 메시지가 출력된다. 그 후 다시 입력 받는다. (이름은 동명이인일 수 있으므로 생략) 그 후 입력 받은 학번과 이름을 Member 클래스의 객체를 배열의 원소이자 객체의 레퍼런스인 Members[i]에 할당한다. 그 후 집행부원의 정보를 입력 받는다. 동아리에는 일반 동아리원, 집행부원 두 분류로 나누어져 있으므로 일반 동아리원과 집행부원이 겹치지 않도록 이전과 같이 학번으로 중복을 확인한다. 그렇게 학번과 이름을 입력 받은 뒤 서브 클래스인 Crew 에서 추가한 기능인 부서를 입력 받는다. 그 후 Crew 클래스에 선언된 출력함수로 출력한다.(Crew 의 출력함수 호출-> Member 의 출력함수 호출-> Member 의 출력함수 실행-> Crew 의 출력함수 실행)

□ 소스코드 및 주석

(week7_2)

```
package week7_2;

public class Member {
    private int iID; //학번과 이름을 필드로 가지는 Member 클래스
    private String sName;

    Member(){//기본 생성자
        iID = 0;
        sName = "";
    }
    Member(int id, String name){//학번과 이름을 매개변수로 가지는
        생성자
        iID = id;
        sName = name;
    }

    public int getID(){//ID 를 반환, 메인함수에서 ID 중복을 확인할
        때 사용한다.
        return iID;
    }

    public void PrintMember() { //학번과 이름을 출력하는 메소드
        System.out.println("Student ID: "+iID);
        System.out.println("Name: "+sName);
    }
}

package week7_2;

public class Crew extends Member{//Member 클래스 상속받음

    private String sDepartment; //부서 기능 추가

    Crew(){//기본 생성자 생성
        super();//super()를 통해 명시적으로 슈퍼 클래스의 생성자를
        호출한다.
        sDepartment = "";
```

```

    }

    Crew(int id, String name, String Department) { //학번과 이름
부서를 매개변수로 한 생성자
        super(id,name); //super()를 통해 명시적으로 슈퍼 클래스의
생성자를 호출한다. 이때 매개변수를 기준으로 호출함
        sDepartment = Department;
    }

    public void PrintCrew(){
        PrintMember(); //Member 클래스의 출력 메소드를 통해 학번과
이름을 출력
        System.out.println("Department: "+sDepartment); //추가
기능으로 부서 출력
    }
}

package week7_2;
import java.util.Scanner;
public class MemberEx {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int iMemberNum, iCrewNum;
        int iID;
        int i, j, k;
        String sName, sDp;
        Scanner oInDev;
        oInDev = new Scanner(System.in); //Scanner 객체
생성하여 입력받도록 함

        System.out.println("일반 동아리원 수를 입력하시오: ");
        iMemberNum = oInDev.nextInt();
        System.out.println("집행부 수를 입력하시오: ");
        iCrewNum = oInDev.nextInt();

        Member Members[];
        Members = new Member[iMemberNum]; //입력 받은 일반 부원
수만큼 객체의 배열 원소 수 지정

        Crew Crews[];

```

```

        Crews = new Crew[iCrewNum]; //입력 받은 집행부원 수만큼
객체의 배열 원소 지정

        System.out.println("---Input Members Information---");
        //일반 부원 정보 입력
        for( i = 0 ; i < iMemberNum; i++) {

            System.out.print("student ID:"); //학번
            iID = oInDev.nextInt();
            for( j = 0 ; j < i ; j++) {
                if(Members[j].getID() == iID) { //만약
같은 반복문 빠져나감
                    break;
                }
            }
            if( i == j ) { //두 개의 값이 같다면 이전의
반복문에서 if 문이 실행되지 않은 것이므로 같은 학번이 존재하지 않다고 판단
                System.out.print("Name: "); //이름
                sName = oInDev.next();

                Members[i] = new Member(iID,sName); //입력받은
학번과 이름을 매개변수로 하는 생성자 실행
                System.out.println("");
            }
            else {
                System.out.println("Error: ID already
exists. Please enter a different ID."); //만약 반복문에서의 if 문이
실행되었다면 같은 학번이 있던 것이므로 중복 메시지 출력
                i--; //다시 입력받도록 함
            }

        }
        System.out.println("---Input Crews Information---");
        //집행 부원 정보 입력
        for(i = 0 ; i < iCrewNum; i++) {
            System.out.print("student ID:"); //학번
            iID = oInDev.nextInt();

```



```

        for( k = 0 ; k < iMemberNum ; k++) { //일반
부원의 학번과의 중복 발생했는지 확인
            if(Members[k].getID() == iID) {
                break;
            }
        }
        for( j = 0 ; j < i ; j++) { //집행 부원끼리 중복이
발생했는지 확인
            if(Crews[j].getID() == iID) {
                break;
            }
        }
        if(i == j&& k==iMemberNum){ //두 곳에서 전부 중복이
발생하지 않았다면

            System.out.print("Name: "); //이름 입력받음
            sName = oInDev.next();
            System.out.print("Department: "); //부서 입력받음
            sDp = oInDev.next();

            Crews[i] = new Crew(iID, sName, sDp); //학번, 이름,
부서를 매개변수로 하는 집행부원 객체 생성
            System.out.println("");
        }
        else {
            System.out.println("Error: ID already
exists. Please enter a different ID."); //중복 메시지 출력

            i--; //다시 입력 받음
        }
    }
    System.out.println("----Input Members----"); //입력받은
일반 부원 전체 정보 출력
    for(i = 0 ; i < iMemberNum; i++) {
        Members[i].PrintMember(); //Member 클래스의 출력
메소드 실행
        System.out.println("");
    }
    System.out.println("----Input Crews----"); //입력받은
집행부원 전체 정보 출력
    for(i = 0 ; i < iCrewNum; i++) {

```

```

        Crews[i].PrintCrew();//추가 기능인 부서까지
출력하기 위해 Crew 클래스의 출력 메소드 실행
        System.out.println("");
    }

    oInDev.close();

}
}

```

□ 결과 및 결과 분석

결과 화면

일반 동아리원 수를 입력하시오:

3

집행부 수를 입력하시오:

2

---Input Members Information---

student ID:2020

Name: Sam

student ID:2020

Error: ID already exists. Please enter a different ID.

student ID:2021

Name: jennie

student ID:2023

Name: Jun

---Input Crews Information---

student ID:2023

Error: ID already exists. Please enter a different ID.

student ID:2022

Name: john

Department: president

student ID:2025

Name: mary

Department: secretary

|

```
---Input Members---
Student ID: 2020
Name: Sam
|
Student ID: 2021
Name: jennie

Student ID: 2023
Name: Jun

---Input Crews---
Student ID: 2022
Name: john
Department: president

Student ID: 2025
Name: mary
Department: secretary
```

결과 분석

일반 부원 입력으로 인해 학번과 이름을 매개 변수로 하는 Member 생성자가 실행되어 해당 필드의 학번, 이름 값에 적절한 값(사용자가 입력한 값)이 할당되어 출력되는 것을 볼 수 있다. 이때 학번이 같을 때 코드 작성 목적과 알맞게 객체가 생성되지 않는 것을 알 수 있다. 그 다음으로 집행 부원 입력으로 인해 학번, 이름 그리고 부서를 매개변수로 하는 Crew 생성자가 실행되어 해당 필드에 적절한 값이 들어가 알맞게 출력된 것을 볼 수 있고 이전처럼 학번의 중복을 막은 것을 알 수 있다.

□ 소감

상속받은 서브클래스에서 기능을 추가하며 구현해보면서 상속에 대해 더 이해할 수 있게 되었다. 또한 오류와는 관련 없지만 추가하면 좋을 것 같은 예외처리를 하며 코드의 완성도를 높여보는 기회를 얻어 좋았다.