

2024년 상반기 K-디지털 트레이닝

# 인터페이스

[KB] IT's Your Life



- ☑ RemoteControl 인터페이스를 완성하세요.
  - o 메서드는 리턴값이 없는 turnOn() 메서드

```
package ch08.sec02;
public _____ RemoteControl {
}
```

☑ RemoteControl 인터페이스를 구현하는 Television, Audiot 클래스를 완성하세요.

```
package ch08.sec02;
public class Television {
}
package ch08.sec02;
public class Audio {
}
```

# **☑** RemoteControl.java

```
package ch08.sec02;

public interface RemoteControl {
   //public 추상 메소드
   public void turnOn();
}
```

# Television.java

```
public class Television implements RemoteControl {
    @Override
    public void turnOn() {
        System.out.println("TV를 켭니다.");
    }
}
```

# Television.java

```
public class Audio implements RemoteControl {
    @Override
    public void turnOn() {
        System.out.println("Audio를 켭니다.");
    }
}
```

앞에서 정의한 인터페이스와 클래스를 이용하여 다음과 같이 출력되도록 완성하세요.

```
package ch08.sec02;

public class RemoteControlExample {
    public static void main(String[] args) {
        _____r;
        rc.turnOn();
        ____;
        rc.turnOn();
    }
}
```

TV를 켭니다. Audio를 켭니다.

# RemoteControlExample.java

```
package ch08.sec02;
public class RemoteControlExample {
 public static void main(String[] args) {
   RemoteControl rc;
   //rc 변수에 Television 객체를 대입
   rc = new Television();
   rc.turnOn();
   //rc 변수에 Audio 객체를 대입(교체시킴)
   rc = new Audio();
   rc.turnOn();
```

```
TV를 켭니다.
Audio를 켭니다.
```

#### ♥ 다음과 같이 인터페이스를 정의하세요.

```
package ch08.sec04;

public interface RemoteControl {
  int MAX_VOLUME = 10;
  int MIN_VOLUME = 0;

  void turnOn();
  void turnOff();
  void setVolume(int volume);
}
```

- ♥ 앞에서 정의한 RemoteControl 인터페이스를 구현하는 Television 클래스를 정의하세요.
  - o setVolume() 메서드
    - 매개변수 volume 값으로 필드 volume 값으로 설정
    - 매개변수 volume 값이 최대 볼륨보다 큰 값이면 최대 볼륨으로 설정
    - 최소 볼륨보다 작은 값이면 최소 볼륨으로 설정
    - "현재 TV 볼륨: 값" 출력

```
package ch08.sec04;

public class Television {
   private int volume;
}
```

### Television.java

```
package ch08.sec04;
                                                    //setVolume() 추상 메소드 오버라이딩
public class Television implements RemoteControl {
                                                    @Override
 //필드
                                                    public void setVolume(int volume) {
 private int volume;
                                                      if(volume>RemoteControl.MAX_VOLUME) {
                                                       this.volume = RemoteControl.MAX VOLUME;
 //turn0n() 추상 메소드 오버라이딩
                                                      } else if(volume<RemoteControl.MIN_VOLUME) {</pre>
 @Override
                                                       this.volume = RemoteControl.MIN VOLUME;
 public void turnOn() {
                                                      } else {
   System.out.println("TV를 켭니다.");
                                                       this.volume = volume;
                                                      System.out.println(
 //turn0ff() 추상 메소드 오버라이딩
                                                                 "현재 TV 볼륨: " + this.volume);
 @Override
 public void turnOff() {
   System.out.println("TV를 끕니다.");
```

#### ♡ 앞에서 정의한 RemoteControl 인터페이스를 구현하는 Audio 클래스를 정의하세요.

- o setVolume() 메서드
  - 매개변수 volume 값으로 필드 volume 값으로 설정
  - 매개변수 volume 값이 최대 볼륨보다 큰 값이면 최대 볼륨으로 설정
  - 최소 볼륨보다 작은 값이면 최소 볼륨으로 설정
  - "현재 Audio 볼륨: 값" 출력

```
package ch08.sec04;

public class Audio {
  private int volume;
}
```

### Audio.java

```
package ch08.sec04;
                                                    //setVolume() 추상 메소드 오버라이딩
public class Audio implements RemoteControl {
                                                    @Override
 //필드
                                                    public void setVolume(int volume) {
 private int volume;
                                                      if(volume>RemoteControl.MAX_VOLUME) {
                                                       this.volume = RemoteControl.MAX VOLUME;
 //turn0n() 추상 메소드 오버라이딩
                                                      } else if(volume<RemoteControl.MIN_VOLUME) {</pre>
 @Override
                                                       this.volume = RemoteControl.MIN VOLUME;
 public void turnOn() {
                                                      } else {
   System.out.println("Audio를 켭니다.");
                                                       this.volume = volume;
                                                      System.out.println("현재 Audio 볼륨: " + volume);
 //turn0ff() 추상 메소드 오버라이딩
 @Override
 public void turnOff() {
   System.out.println("Audio를 끕니다.");
```

○ 앞에서 정의한 인터페이스와 클래스를 이용하여 다음과 같이 출력되도록 RemoteControlExample 클래스를 완성하세요

```
package ch08.sec04;
public class RemoteControlExample {
 public static void main(String[] args) {
       rc;
  rc = ;
  rc =
```

TV를 켭니다. 현재 TV 볼륨: 5 TV를 끕니다. Audio를 켭니다. 현재 Audio 볼륨: 5 Audio를 끕니다.

## RemoteControlExample.java

```
package ch08.sec04;
public class RemoteControlExample {
 public static void main(String[] args) {
   //인터페이스 변수 선언
   RemoteControl rc;
   //Television 객체를 생성하고 인터페이스 변수에 대입
   rc = new Television();
   rc.turnOn();
   rc.setVolume(5);
   rc.turnOff();
   //Audio 객체를 생성하고 인터페이스 변수에 대입
   rc = new Audio();
   rc.turnOn();
                                     TV를 켭니다.
   rc.setVolume(5);
                                     현재 TV 볼륨: 5
   rc.turnOff();
                                     TV를 끕니다.
                                    Audio를 켭니다.
                                     현재 Audio 볼륨: 5
                                     Audio를 끕니다.
```

#### ♥ 다음과 같이 인터페이스를 정의하세요.

```
package ch08.sec08;

public interface RemoteControl {
   void turnOn();
   void turnOff();
}

package ch08.sec08;

public interface Searchable {
   void search(String url);
}
```

♡ 앞에서 정의한 RemoteControl, Searchable 인터페이스를 모두 구현한 SmartTelevision 클래스를 정의하세요.

```
package ch08.sec08;
public class SmartTelevision {
}
```

# SmartTelevision.java

```
package ch08.sec08;
public class SmartTelevision implements RemoteControl, Searchable {
 //turn0n() 추상 메소드 오버라이딩
 @Override
 public void turnOn() {
   System.out.println("TV를 켭니다.");
                                       RemoteControl 인터페이스구현
 //turnoff() 추상 메소드 오버라이딩
 @Override
 public void turnOff() {
   System.out.println("TV를 끕니다.");
 //search() 추상 메소드 오버라이딩
 @Override
 public void search(String url) {
                                            Searchable 인터페이스구현
   System.out.println(url + "을 검색합니다.");
```

앞에서 만든 인터페이스와 클래스를 이용하여 다음 코드를 완성하세요.

TV를 켭니다. TV를 끕니다. https://www.youtube.com을 검색합니다.

# MultiInterfaceImplExample.java

```
package ch08.sec08;
public class MultiInterfaceImplExample {
 public static void main(String[] args) {
   //RemoteControl 인터페이스 변수 선언 및 구현 객체 대입
   RemoteControl rc = new SmartTelevision();
   //RemoteControl 인터페이스에 선언된 추상 메소드만 호출 가능
   rc.turnOn();
   rc.turnOff();
   //Searchable 인터페이스 변수 선언 및 구현 객체 대입
   Searchable searchable = new SmartTelevision();
   //Searchable 인터페이스에 선언된 추상 메소드만 호출 가능
   searchable.search("https://www.youtube.com");
```

```
TV를 켭니다.
TV를 끕니다.
https://www.youtube.com을 검색합니다.
```

☑ 다음 인터페이스를 정의하세요.

```
package ch08.sec09;

public interface InterfaceA {
   void methodA();
}

package ch08.sec09;

public interface InterfaceB {
   void methodB();
}
```

☑ InterfaceA, InterfaceB를 상속하는 InterfaceC를 정의하세요.

```
package ch08.sec09;

public interface InterfaceC ______{
   void methodC();
}
```

# **☑** InterfaceC.java

```
package ch08.sec09;

public interface InterfaceC extends InterfaceA, InterfaceB {
    //추상 메소드
    void methodC();
}
```

- ☑ interfaceC를 구현하는 InterfaceCImpl 클래스를 정의하세요.
  - ㅇ 어떤 메서드가 호출되는지 출력하는 코드로 각 메서드를 정의
    - InterfaceCImpl-methodA() 실행
    - InterfaceCImpl-methodB() 실행
    - InterfaceCImpl-methodC() 실행

```
package ch08.sec09;
public class InterfaceCImpl {
}
```

# **☑** InterfaceCImpl.java

```
package ch08.sec09;
public class InterfaceCImpl implements InterfaceC {
 public void methodA() {
   System.out.println("InterfaceCImpl-methodA() 실행");
 public void methodB() {
   System.out.println("InterfaceCImpl-methodB() 실행");
 public void methodC() {
   System.out.println("InterfaceCImpl-methodC() 실행");
```

#### 앞에서 정의한 인터페이스와 클래스를 이용하여 다음 코드를 완성하세요.

```
package ch08.sec09;
public class ExtendsExample {
 public static void main(String[] args) {
   InterfaceCImpl impl = new InterfaceCImpl();
                    ia = impl;
   ia.methodA();
   System.out.println();
                    ib = impl;
   ib.methodB();
   System.out.println();
                                              InterfaceCImpl-methodA() 실행
                    ic = impl;
                                              InterfaceCImpl-methodB() 실행
   ic.methodA();
   ic.methodB();
                                              InterfaceCImpl-methodA() 실행
   ic.methodC();
                                              InterfaceCImpl-methodB() 실행
                                              InterfaceCImpl-methodC() 실행
```

# **Extends**Example.java

```
package ch08.sec09;
public class ExtendsExample {
 public static void main(String[] args) {
   InterfaceCImpl impl = new InterfaceCImpl();
   InterfaceA ia = impl;
   ia.methodA();
   //ia.methodB();
   System.out.println();
   InterfaceB ib = impl;
   //ib.methodA();
   ib.methodB();
                                             InterfaceCImpl-methodA() 실행
   System.out.println();
                                             InterfaceCImpl-methodB() 실행
   InterfaceC ic = impl;
   ic.methodA();
                                             InterfaceCImpl-methodA() 실행
   ic.methodB();
                                             InterfaceCImpl-methodB() 실행
   ic.methodC();
                                             InterfaceCImpl-methodC() 실행
```

#### 다음 인터페이스와 클래스를 정의하세요.

```
package ch08.sec10.exam02;
public interface Vehicle {
 void run();
package ch08.sec10.exam02;
public class Bus implements Vehicle {
 @Override
 public void run() {
   System.out.println("버스가 달립니다.");
 public void checkFare() {
   System.out.println("승차요금을 체크합니다.");
```

앞의 인터페이스와 클래스를 이용하여 다음 코드에서 잘못된 부분을 찾아 수정하고, 코드를 완성하세요.

```
package ch08.sec10.exam02;
public class CastingExample {
 public static void main(String[] args) {
   Vehicle vehicle = new Bus();
   vehicle.run();
   vehicle.checkFare();
   Bus bus =
                               vehicle;
   bus.run();
   bus.checkFare();
```

```
버스가 달립니다.
버스가 달립니다.
승차요금을 체크합니다.
```

# **☑** CastingExample.java

```
package ch08.sec10.exam02;
public class CastingExample {
 public static void main(String[] args) {
   //인터페이스 변수 선언과 구현 객체 대입
   Vehicle vehicle = new Bus();
   //인터페이스를 통해서 호출
   vehicle.run();
   //vehicle.checkFare(); (x)
   //강제 타입 변환후 호출
   Bus bus = (Bus) vehicle;
   bus.run();
   bus.checkFare();
```

```
버스가 달립니다.
버스가 달립니다.
승차요금을 체크합니다.
```