```
////////
//////// 인터페이스, default 메소드, static 메소드 설명 소스
////////

// RemoteControl.java
//
public interface RemoteControl {
	//상수
	int MAX_VOLUME = 10;
	int MIN_VOLUME = 0;

	//추상 메소드
	void turnOn();
	void turnOff();
	void setVolume(int volume);

	//디폴트 메소드
	default void setMute(boolean mute) {
		if(mute)
			System.out.println("무음 처리합니다.");
		else
			System.out.println("무음 해제합니다.");
	}

	//정적 메소드
	static void changeBattery() {
		System.out.println("건전지를 교환합니다.");
	}
}

//Audio.java
//
public class Audio implements RemoteControl {
	//필드
	private int volume;
	private boolean mute;

	//turnOn() 추상 메소드의 실체 메소드
	public void turnOn() {
		System.out.println("Audio를 켭니다.");
	}
	//turnOff() 추상 메소드의 실체 메소드
	public void turnOff() {
		System.out.println("Audio를 끕니다.");
	}
	//setVolume() 추상 메소드의 실체 메소드
	public void setVolume(int volume) {
		if(volume>RemoteControl.MAX_VOLUME) {
			this.volume = RemoteControl.MAX_VOLUME;
		} else if(volume<RemoteControl.MIN_VOLUME) {
			this.volume = RemoteControl.MIN_VOLUME;
		} else {
			this.volume = volume;
		}
		System.out.println("현재 Audio 볼륨: " + volume);
	}

	@Override
	public void setMute(boolean mute) {
		this.mute = mute;
		if(mute) {
			System.out.println("Audio 무음 처리합니다.");
```

```
                } else {
                        System.out.println("Audio 무음 해제합니다.");
                }
        }
}

//Television.java
//
public class Television implements RemoteControl {
        //필드
        private int volume;

        //turnOn() 추상 메소드의 실체 메소드
        public void turnOn() {
                System.out.println("TV를 켭니다.");
        }
        //turnOff() 추상 메소드의 실체 메소드
        public void turnOff() {
                System.out.println("TV를 끕니다.");
        }
        //setVolume() 추상 메소드의 실체 메소드
        public void setVolume(int volume) {
                if(volume>RemoteControl.MAX_VOLUME) {
                        this.volume = RemoteControl.MAX_VOLUME;
                } else if(volume<RemoteControl.MIN_VOLUME) {
                        this.volume = RemoteControl.MIN_VOLUME;
                } else {
                        this.volume = volume;
                }
                System.out.println("현재 TV 볼륨: " + volume);
        }
}

//RemoteControlExample.java
//
public class RemoteControlExample {
        public static void main(String[] args) {
                RemoteControl rc = null;

                rc = new Television();
                rc.turnOn();
                rc.setMute(true);

                rc = new Audio();
                rc.turnOn();
                rc.setMute(true);

                RemoteControl.changeBattery();
        }
}

////////
//////// 다중 인터페이스 구현
////////

// Searchable.java
//
public interface Searchable {
        void search(String url);
}

// SmartTelevision.java
```

```
//
public class SmartTelevision implements RemoteControl, Searchable {
	private int volume;

	public void turnOn() {
		System.out.println("TV를 켭니다.");
	}
	public void turnOff() {
		System.out.println("TV를 끕니다.");
	}
	public void setVolume(int volume) {
		if(volume>RemoteControl.MAX_VOLUME) {
			this.volume = RemoteControl.MAX_VOLUME;
		} else if(volume<RemoteControl.MIN_VOLUME) {
			this.volume = RemoteControl.MIN_VOLUME;
		} else {
			this.volume = volume;
		}
		System.out.println("현재 TV 볼륨: " + volume);
	}

	public void search(String url) {
		System.out.println(url + "을 검색합니다.");
	}
}

// RemoteControlExample.java
//
public class RemoteControlExample {
	public static void main(String[] args) {
		SmartTelevision stv = new SmartTelevision();

		Searchable searchable = stv;
		searchable.search("TvN");

		RemoteControl rc = stv;
		rc.turnOn();
		rc.setMute(true);

		rc = new Audio();
		rc.turnOn();
		rc.setMute(true);

		RemoteControl.changeBattery();
	}
}

////////
//////// 인터페이스, polymorphism, feild, array 설명 소스
////////

// Tire.java
//
public interface Tire {
	public void roll();
}

// KumhoTire.java
//
public class KumhoTire implements Tire {
	@Override
	public void roll() {
```

```
                System.out.println("금호 타이어가 굴러갑니다.");
        }
}

// HankookTire.java
//
public class HankookTire implements Tire {
        @Override
        public void roll() {
                System.out.println("한국 타이어가 굴러갑니다.");
        }
}

// Car.java
//
public class Car {
        Tire[] tires = {
                new HankookTire(),
                new HankookTire(),
                new HankookTire(),
                new HankookTire()
        };
        void run() {
                for(Tire tire : tires) {
                        tire.roll();
                }
        }
}

//CarExample.java
//
public class CarExample {
        public static void main(String[] args) {
                Car myCar = new Car();

                myCar.run();

                myCar.tires[0] = new KumhoTire();
                myCar.tires[1] = new KumhoTire();

                myCar.run();
        }
}

////////
//////// 인터페이스, polymorphism 설명 소스
////////

// Vehicle.java
//
public interface Vehicle {
        public void run();
}

// Bus.java
//
public class Bus implements Vehicle {
        @Override
        public void run() {
                System.out.println("버스가 달립니다.");
        }
```

```
        public void checkFare() {
                System.out.println("승차요금을 체크합니다.");
        }
}

// Taxi.java
//
public class Taxi implements Vehicle {
        @Override
        public void run() {
                System.out.println("택시가 달립니다.");
        }
}

// Driver.java
//
public class Driver {
        public void drive(Vehicle vehicle) {
                vehicle.run();
        }
}

//Driver.java
//
public class Driver {
        public void drive(Vehicle vehicle) {
                if(vehicle instanceof Bus) {
                        Bus bus = (Bus) vehicle;
                        bus.checkFare();
                }
                vehicle.run();
        }
}

// DriverExample.java
//
public class DriverExample {
        public static void main(String[] args) {
                Driver driver = new Driver();

                Bus bus = new Bus();
                Taxi taxi = new Taxi();

                driver.drive(bus);
                driver.drive(taxi);
        }
}

//VehicleExample.java
//
public class VehicleExample {
        public static void main(String[] args) {
                Vehicle vehicle = new Bus();

                vehicle.run();
                //vehicle.checkFare(); (x)

                Bus bus = (Bus) vehicle;  //강제타입변환

                bus.run();
                bus.checkFare();
        }
```

```
}

////////
//////// 인터페이스 상속 설명 소스
////////

//InterfaceA.java
//
public interface InterfaceA {
	public void methodA();
}

//InterfaceB.java
//
public interface InterfaceB {
	public void methodB();
}

//InterfaceC.java
//
public interface InterfaceC extends InterfaceA, InterfaceB {
	public void methodC();
}

//ImplementationC.java
//
public class ImplementationC implements InterfaceC {
	public void methodA() {
		System.out.println("ImplementationC-methodA() 실행");
	}

	public void methodB() {
		System.out.println("ImplementationC-methodB() 실행");
	}

	public void methodC() {
		System.out.println("ImplementationC-methodC() 실행");
	}
}

//Example.java
//
public class Example {
	public static void main(String[] args) {
		ImplementationC impl = new ImplementationC();

		InterfaceA ia = impl;
		ia.methodA();
		System.out.println();

		InterfaceB ib = impl;
		ib.methodB();
		System.out.println();

		InterfaceC ic = impl;
		ic.methodA();
		ic.methodB();
		ic.methodC();
	}
}

////////
```

```
//////// default, 인터페이스, implements 설명 소스
////////

// MyInterface.java
//
public interface MyInterface {
	public void method1();

	public default void method2() {
		System.out.println("MyInterface-method2 실행");
	}
}

// MyClassA.java
//
public class MyClassA implements MyInterface {
	@Override
	public void method1() {
		System.out.println("MyClassA-method1() 실행");
	}
}

// MyClassB.java
//
public class MyClassB implements MyInterface {
	@Override
	public void method1() {
		System.out.println("MyClassB-method1() 실행");
	}

	@Override
	public void method2() {
		System.out.println("MyClassB-method2() 실행");
	}
}

// DefaultMethodExample.java
//
public class DefaultMethodExample {
	public static void main(String[] args) {
		MyInterface mi1 = new MyClassA();
		mi1.method1();
		mi1.method2();

		MyInterface mi2 = new MyClassB();
		mi2.method1();
		mi2.method2();
	}
}

////////
//////// default, 인터페이스, 상속 설명 소스
////////

// ParentInterface.java
//
 public interface ParentInterface {
    public void method1();
    public default void method2() { /*실행문*/ }
}

// ChildInterface1.java
```

```
//
 public interface ChildInterface1 extends ParentInterface {
        public void method3();
}

// ChildInterface2.java
//
public interface ChildInterface2 extends ParentInterface {
        @Override
        public default void method2() { /*실행문*/ }
        public void method3();
}

// ChildInterface3.java
//
public interface ChildInterface3 extends ParentInterface {
        @Override
        public void method2();
        public void method3();
}

// DefaultMethodExtendsExample.java
//
public class DefaultMethodExtendsExample {
        public static void main(String[] args) {
                ChildInterface1 ci1 = new ChildInterface1() {
                        @Override
                        public void method1() { /*실행문*/ }
                        @Override
                        public void method3() { /*실행문*/ }
                };
                ci1.method1();
                ci1.method2();  //ParentInterface의 method2() 호출
                ci1.method3();
                //-----------------------------------------------------------

                ChildInterface2 ci2 = new ChildInterface2() {
                        @Override
                        public void method1() { /*실행문*/ }
                        @Override
                        public void method3() { /*실행문*/ }
                };
                ci2.method1();
                ci2.method2();  //ChildInterface2의 method2() 호출
                ci2.method3();
                //-----------------------------------------------------------

                ChildInterface3 ci3 = new ChildInterface3() {
                        @Override
                        public void method1() { /*실행문*/ }
                        @Override
                        public void method2() { /*실행문*/ }
                        @Override
                        public void method3() { /*실행문*/ }
                };
                ci3.method1();
                ci3.method2();  //ChildInterface3 구현 객체의 method2() 호출
                ci3.method3();
        }
}
```