

```

//////////
//////////polymorphism, array 설명 소스
//////////

//
//Tire.java
//
package test_polymorphism;

public class Tire {
    //필드
    public int maxRotation;           //최대 회전수(최대 수명)
    public int accumulatedRotation;   //누적 회전수
    public String location;           //타이어의 위치

    //생성자
    public Tire(String location, int maxRotation) {
        this.location = location;
        this.maxRotation = maxRotation;
    }

    //메소드
    public boolean roll() {
        ++accumulatedRotation;
        if(accumulatedRotation<maxRotation) {
            System.out.println(location + " Tire 수명: " +
(maxRotation-accumulatedRotation) + "회");
            return true;
        } else {
            System.out.println("*** " + location + " Tire 펑크 ***");
            return false;
        }
    }
}

//
//KumhoTire.java
//
package test_polymorphism;

public class KumhoTire extends Tire {
    //필드
    //생성자
    public KumhoTire(String location, int maxRotation) {
        super(location, maxRotation);
    }

    //메소드
    @Override
    public boolean roll() {
        ++accumulatedRotation;
        if(accumulatedRotation<maxRotation) {
            System.out.println(location + " KumhoTire 수명: " +
(maxRotation-accumulatedRotation) + "회");
            return true;
        } else {
            System.out.println("*** " + location + " KumhoTire 펑크 ***");
            return false;
        }
    }
}

//
//HankookTire.java

```

```
//
package test_polymorphism;

public class HankookTire extends Tire {
    //필드
    //생성자
    public HankookTire(String location, int maxRotation) {
        super(location, maxRotation);
    }
    //메소드
    @Override
    public boolean roll() {
        ++accumulatedRotation;
        if(accumulatedRotation<maxRotation) {
            System.out.println(location + " HankookTire 수명: " +
(maxRotation-accumulatedRotation) + "회");
            return true;
        } else {
            System.out.println("*** " + location + " HankookTire 펑크 ***");
            return false;
        }
    }
}

//
//Car.java
//
package test_polymorphism;

public class Car {
    //필드
    Tire[] tires = {
        new Tire("앞왼쪽", 6),
        new Tire("앞오른쪽", 2),
        new Tire("뒤왼쪽", 3),
        new Tire("뒤오른쪽", 4)
    };

    //메소드
    int run() {
        System.out.println("[자동차가 달립니다.]");
        for(int i=0; i<tires.length; i++) {
            if(tires[i].roll()!=false) {
                stop();
                return (i+1);
            }
        }
        return 0;
    }

    void stop() {
        System.out.println("[자동차가 멈춥니다.]");
    }
}

//
//CarExample.java
//
package test_polymorphism;

public class CarExample {
    public static void main(String[] args) {
        Car car = new Car();
    }
}
```

```

        for(int i=1; i<=5; i++) {
            int problemLocation = car.run();
            if(problemLocation != 0) {
                System.out.println(car.tires[problemLocation-1].location + "
HankookTire로 교체");
                car.tires[problemLocation-1] = new
HankookTire(car.tires[problemLocation-1].location, 15);
            }
            System.out.println("-----");
        }
    }
}

/////////
///////// method_polymorphism 설명 소스
/////////

//
//Vehicle.java
//
package method_polymorphism:

public class Vehicle {
    public void run() {
        System.out.println("차량이 달립니다.");
    }
}

//
//Bus.java
//
package method_polymorphism:

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

//
//Taxi.java
//
package method_polymorphism:

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

//
//Driver.java
//
package method_polymorphism:

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

```

```
//
//DriverExample.java
//
package method_polymorphism;

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);
    }
}

/////////
///////// casting_instanceof 설명 소스
/////////

//
//Parent.java
//
package casting_instanceof;

public class Parent {
    public String field1;

    public void method1() {
        System.out.println("Parent-method1()");
    }

    public void method2() {
        System.out.println("Parent-method2()");
    }
}

//
//Child.java
//
package casting_instanceof;

public class Child extends Parent {
    public String field2;

    public void method3() {
        System.out.println("Child-method3()");
    }
}

//
//ChildExample.java
//
package casting_instanceof;

public class ChildExample {
    public static void main(String[] args) {
        Parent parent = new Child();
        parent.field1 = "data1";
        parent.method1();
        parent.method2();
    }
}
```

```

        /*
        parent.field2 = "data2";  //(불가능)
        parent.method3();         //(불가능)
        */

        Child child = (Child) parent;
        child.field2 = "yyy";      //(가능)
        child.method3();          //(가능)
    }
}

//
//ChildExample.java
//
package casting_instanceof;

public class InstanceofExample {
    public static void method1(Parent parent) {
        if(parent instanceof Child) {
            Child child = (Child) parent;
            System.out.println("method1 - Child로 변환 성공");
        } else {
            System.out.println("method1 - Child로 변환되지 않음");
        }
    }

    public static void method2(Parent parent) {
        Child child = (Child) parent;
        System.out.println("method2 - Child로 변환 성공");
    }

    public static void main(String[] args) {
        Parent parentA = new Child();
        method1(parentA);
        method2(parentA);

        Parent parentB = new Parent();
        method1(parentB);
        method2(parentB); //예외 발생
    }
}

/////////
///////// abstract_method 설명 소스
/////////

//
//Animal.java
//
package abstract_method;

public abstract class Animal {
    public String kind;

    public void breathe() {
        System.out.println("숨을 쉰니다.");
    }

    public abstract void sound();
}

//
//Cat.java

```

```
//
package abstract_method;

public class Cat extends Animal {
    public Cat() {
        this.kind = "포유류";
    }

    @Override
    public void sound() {
        System.out.println("야옹");
    }
}

//
//Dog.java
//
package abstract_method;

public class Dog extends Animal {
    public Dog() {
        this.kind = "포유류";
    }

    @Override
    public void sound() {
        System.out.println("멍멍");
    }
}

//
//AnimalExample.java
//
package abstract_method;

public class AnimalExample {
    public static void main(String[] args) {
        Dog dog = new Dog();
        Cat cat = new Cat();
        dog.sound();
        cat.sound();
        System.out.println("-----");

        //변수의 자동 타입 변환
        Animal animal = null;
        animal = new Dog();
        animal.sound();
        animal = new Cat();
        animal.sound();
        System.out.println("-----");

        //매개변수의 자동 타입 변환
        animalSound(new Dog());
        animalSound(new Cat());
    }

    public static void animalSound(Animal animal) {
        animal.sound();
    }
}
```