

2024년 상반기 K-디지털 트레이닝

인터페이스

[KB] IT's Your Life



♥ 다음과 같이 Tire 인터페이스를 정의하세요.

```
package ch08.sec11.exam01;

public interface Tire {
  void roll();
}
```

☑ Tire 인터페이스를 구현하는 HankookTire, KumhoTire 클래스를 정의하세요.

```
package ch08.sec11.exam01;
public class HankookTire {
}
package ch08.sec11.exam01;
public class KumhoTire {
}
```

Tire.java

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

KumhoTire.java

```
package ch08.sec11.exam01;

public class KumhoTire implements Tire {
    //추상 메소드 재정의
    @Override
    public void roll() {
        System.out.println("금호 타이어가 굴러갑니다.");
    }
}
```

- 다음 조건을 만족하는 Car 클래스를 정의하세요.
 - o Tire 타입의 tire1, tire2 멤버 변수 정의
 - o 기본값으로 HankookTire 인스턴스로 정의
 - o run() 메서드에서 각 멤버 변수의 roll() 메서드 호출

```
package ch08.sec11.exam01;
public class Car {
}
```

Car.java

```
public class Car {
   //필드
   Tire tire1 = new HankookTire();
   Tire tire2 = new HankookTire();

   //메소드
   void run() {
      tire1.roll();
      tire2.roll();
   }
}
```

♡ 다음과 같이 출력되도록 CarEample을 완성하세요.

```
package ch08.sec11.exam01;

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

    myCar.run();
}
```

```
한국 타이어가 굴러갑니다.
한국 타이어가 굴러갑니다.
금호 타이어가 굴러갑니다.
금호 타이어가 굴러갑니다.
```

CarExample.java

```
package ch08.sec11.exam01;
public class CarExample {
 public static void main(String[] args) {
  //자동차 객체 생성
  Car myCar = new Car();
  //run() 메소드 실행
  myCar.run();
  //타이어 객체 교체
  myCar.tire1 = new KumhoTire();
  myCar.tire2 = new KumhoTire();
  //run() 메소드 실행(다형성: 실행 결과가 다름)
  myCar.run();
               한국 타이어가 굴러갑니다.
               한국 타이어가 굴러갑니다.
               금호 타이어가 굴러갑니다.
               금호 타이어가 굴러갑니다.
```

○ 다음 클래스 다이어그램과 같이 인터페이스와 클래스를 정의하세요.

```
package ch08.sec11.exam02;
public interface Vehicle {
 void run();
package ch08.sec11.exam02;
public class Bus implements Vehicle {
 @Override
 public void run() {
   System.out.println("버스가 달립니다.");
package ch08.sec11.exam02;
public class Taxi implements Vehicle {
 @Override
 public void run() {
   System.out.println("택시가 달립니다.");
```

```
package ch08.sec11.exam02;

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

Driver.java

```
package ch08.sec11.exam02;

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

다음과 같이 출력되도록 아래 코드를 완성하세요.

```
package ch08.sec11.exam02;

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

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

   driver.drive( );
   driver.drive( );
}
```

```
버스가 달립니다.
택시가 달립니다.
```

☑ DriverExample.java

```
package ch08.sec11.exam02;
public class DriverExample {
 public static void main(String[] args) {
   //Driver 객체 생성
   Driver driver = new Driver();
   //Vehicle 구현 객체 생성
   Bus bus = new Bus();
   Taxi taxi = new Taxi();
   //매개값으로 구현 객체 대입(다형성: 실행 결과가 다름)
   driver.drive(bus); // 자동 타입 변환: Bus → Verhicle
   driver.drive(taxi); // 자동 타입 변환: Taxi → Verhicle
```

다음과 같이 인터페이스와 클래스를 정의하세요.

```
package ch08.sec12;
public interface Vehicle {
 void run();
package ch08.sec12;
public class Bus implements Vehicle {
 @Override
 public void run() {
   System.out.println("버스가 달립니다.");
 public void checkFare() {
   System.out.println("승차요금을 체크합니다.");
```

```
package ch08.sec12;

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

♡ 다음과 같이 출력되도록 InstanceofExample 클래스의 ride() 메서드를 추가하세요.

```
package ch08.sec12;

public class InstanceofExample {
   public static void main(String[] args) {
      Taxi taxi = new Taxi();
      Bus bus = new Bus();

      ride(taxi);
      System.out.println();
      ride(bus);
   }
}
```

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

InstanceofExample.java

```
public class InstanceofExample {
  public static void main(String[] args) {
    //구현 객체 생성
    Taxi taxi = new Taxi();
    Bus bus = new Bus();

    //ride() 메소드 호출 시 구현 객체를 매개값으로 전달
    ride(taxi);
    System.out.println();
    ride(bus);
}
```

☑ InstanceofExample.java

```
//인터페이스 매개변수를 갖는 메소드
public static void ride(Vehicle vehicle) {
 //방법1
 /*if(vehicle instanceof Bus) {
   Bus bus = (Bus) vehicle;
   bus.checkFare();
 }*/
 //방법2
 if(vehicle instanceof Bus bus) {
   bus.checkFare();
 vehicle.run();
```