

[프로젝트 이름]:

Project_Car

[프로젝트를 만든 목적]:

임의의 자동차 판매소가 있다고 가정하고, 고객이 원하는 자동차의 각 부분을 Java 를 이용하여 객체로 생성하고, 그것을 제어하는 메소드를 이용하여 자동차 객체를 생성하는 것이 목표이다.

[프로젝트 패키지과 클래스의 내용 설명]

[Car 패키지]

: Body, Engine, Seat, Tire, SpeedGauge, Vendor, Car, CarMain 클래스를 포함한다.

설문조사로 얻은 고객의 데이터로 실제 자동차를 생성한다.

자동차의 각부분을 생성하고 main 메소드를 실행한다.

CarMain (class) : 설문조사를 실시하고, 자동차를 생성 및 테스트를 하는 클래스.

Car (class) : 모든 객체를 통합하여 자동차를 생성하는 클래스.

Body (class) : 자동차의 차체를 생성하는 클래스.

Engine (class) : 자동차의 엔진을 생성하는 클래스.

Seat (class) : 자동차의 시트를 생성하는 클래스.

Tire (class) : 자동차의 타이어를 생성하는 클래스.

SpeedGauge (class) : 자동차의 속도계를 생성하는 클래스.

Vendor (class) : 자동차의 브랜드명을 지정하는 클래스.

[Doc 패키지]

: DocBody, DocEngine, DocSeat, DocSpeedGauge, DocTire, DocVendor 인터페이스를 포함하고 있으며, 이를 상속하는 DocCar 인터페이스와 DocCar 를 실제 구현하는 Doc 클래스를 포함한다.

고객이 원하는 자동차를 만들기위한 설문 조사를 담당하는 모듈이다.

Doc (class) : 실제 각 설문조사 메소드를 구현하는 클래스로서 DocCar 인터페이스를 구현한다. 자동

차 판매회사에서 사용할 수 있는 계약서 메소드의 리스트이다.

DocCar (interface) : Doc 패키지 내의 모든 인터페이스를 상속받아 하나의 통합된 인터페이스를 만든다.

DocBody (interface) : 자동차의 차체 설정에 관한 설문조사 메소드를 포함하는 인터페이스.

DocEngine (interface) : 자동차의 엔진 설정에 관한 설문조사 메소드를 포함하는 인터페이스.

DocSeat (interface) : 자동차의 좌석 설정에 관한 설문조사 메소드를 포함하는 인터페이스.

DocSpeedGauge (interface) : 자동차의 속도계의 설정에 관한 설문조사 메소드를 포함하는 인터페이스.

DocTire (interface) : 자동차의 바퀴 설정에 관한 설문조사 메소드를 포함하는 인터페이스.

DocVendor (interface) : 자동차의 벤더명 설정에 관한 설문조사 메소드를 포함하는 인터페이스.

[Test 패키지]

: Speed 인터페이스를 포함하고, 이를 상속하는 TestCar 인터페이스와 TestCar 를 상속받아 실제 구현하는 Test 클래스를 포함한다.

고객이 원하는 자동차를 만든시 실제로 운전하는 테스트 드라이빙을 담당하는 모듈이다.

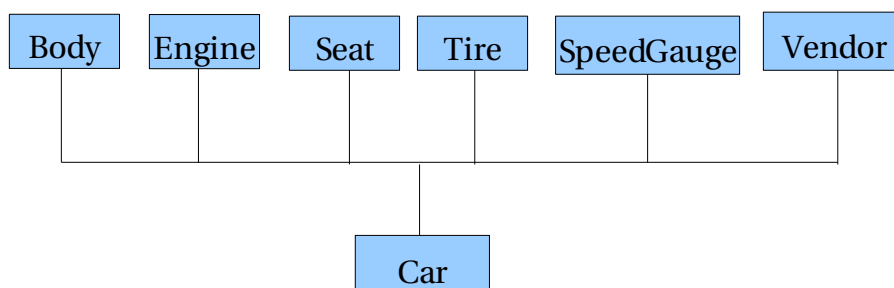
Test (class) : 실제 각 테스트 메소드를 구현하는 클래스로서 TestCar 인터페이스를 구현한다.

TestCar (interface) : Test 패키지에 있는 모든 인터페이스를 상속받아 하나의 통합된 인터페이스를 만든다.

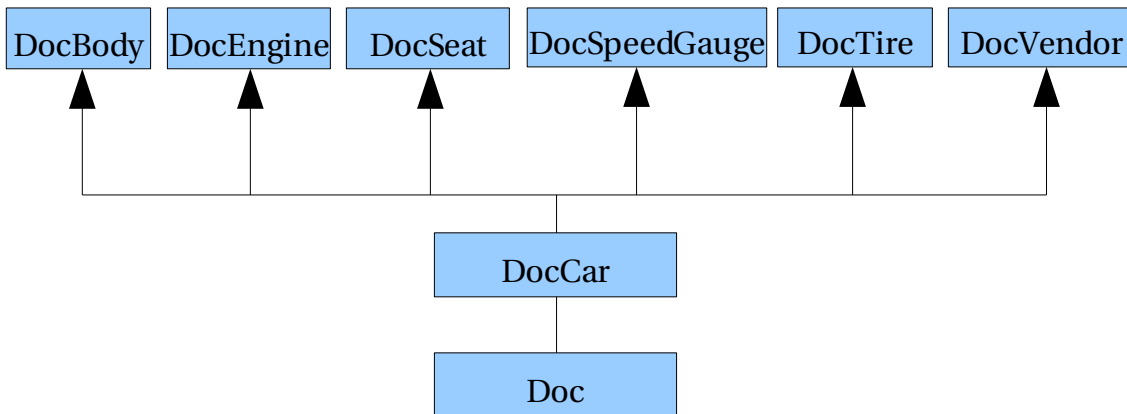
Speed (interface) : 자동차의 속도계 테스트에 관한 메소드를 포함하는 인터페이스.

[객체 상관관계]

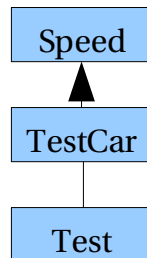
[Car 패키지]



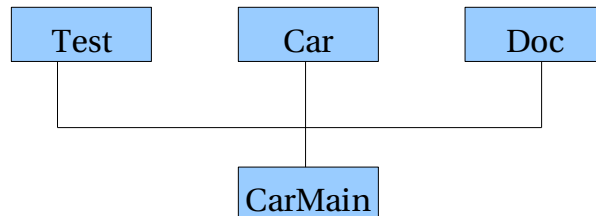
[Doc 패키지]



[Test 패키지]



[Main 메소드]



[각 객체별 멤버변수 및 메소드 소개]

[Package Car]

(Body)

String material : 차체의 재질을 저장하는 변수

Body() : 생성자. 멤버변수를 초기화 한다.

void setBody(String data) : 차체의 재질을 설정하는 메소드. 인자값으로 차체의 재질을

변경한다.

String getBody() : 차체의 재질을 반환하는 메소드. 리턴값으로 재질의 설정값을 리턴한다.

(Engine)

int cylinder : 실린더의 갯수를 저장하는 변수

int max_cylinder : 설정할 수 있는 최대 실린더의 수

Engine() : 생성자. 멤버변수를 초기화 한다.

boolean setCylinder(int number) : 실린더의 갯수를 설정하는 메소드. 실린더의 갯수가 너무 적거나 많을 경우 에러 메시지를 출력한다. 정상종료시 true 를 반환. 오류일 경우 false 를 반환.

int getCylinder() : 실린더의 갯수를 반환하는 메소드. 리턴값으로 실린더의 갯수를 리턴한다.

(Seat)

int seat_number : 시트의 갯수를 저장하는 변수

Seat() : 생성자. 멤버변수를 초기화 한다.

boolean setSeat(int number) : 시트의 갯수를 설정하는 메소드. 인자값으로 들어온 수만큼 시트의 갯수를 설정한다. 정상 종료시 true, 그외의 경우에는 false 를 리턴한다.

int getSeat() : 시트의 갯수를 반환하는 메소드. 리턴값으로 시트의 갯수를 리턴한다.

(SpeedGauge)

int max_speed : 최대 속도를 저장하는 변수

int current_speed : 현재 속도를 저장하는 변수

int distance : 주행거리를 저장하는 변수

SpeedGauge() : 생성자. 변수를 초기화한다. 최대 속도를 설정한다.

boolean accelSpeed(int speed) : 엑셀 메소드. 현재속도와 이동거리를 증가시킨다. 만약 현재 속도보다 높은 속도가 인자값으로 들어오지 않을 경우 에러메

시지를 Dp한다.

정상종료시 true, 그 외의 모든 경우에 false 를 리턴한다.

boolean breakSpeed(int speed) : 브레이크 메소드. 현재속도를 감소시키고 이동거리를 증가시킨다. 만약 현재 속도보다 높은값이 나오거나 마이너스 값이 인자값으로 들어올 경우 에러메시지를 Dp한다.

정상종료시 true, 그 외의 모든 경우에 false 를 리턴한다.

int getSpeed() : 현재 스피드를 반환하는 메소드. 리턴값으로 current_speed 값을 반환한다.

int getDistance() : 현재 주행거리를 반환하는 메소드. 리턴값으로 distance 값을 반환한다.

(Tire)

int tire_number : 타이어의 갯수를 저장하는 변수

int max_tire : 타이어의 최대 갯수를 저장하는 변수

Tire() : 생성자. 변수를 초기화한다.

boolean setTire(int number) : 타이어의 갯수를 설정하는 메소드. max_tire 의 값과 비교하여 설정가능한 갯수인지 확인하고 오류시 에러메시지를 DP 한다.

정상종료시 true, 그 외의 모든 경우에 false 를 리턴한다.

int getTire() : 타이어의 갯수를 반환하는 메소드.

(Vendor)

String vendor_name : 벤더명을 저장하는 변수

Vendor() : 생성자. 변수를 초기화한다.

void setVendor(String name) : 벤더명을 설정하는 메소드.

String getVendor() : 벤더명을 반환하는 메소드. vendor_name 값을 확인하여 NULL 값일 경우 에러메시지를 DP 하고 NULL 값을 리턴한다.

(Car)

static Engine engine : Engine 객체.

static Body body	: Body 객체.
static Seat seat	: Seat 객체.
static SpeedGauge speedGauge	: SpeedGauge 객체.
static Tire tire	: Tire 객체.
static Vendor vendor	: Vendor 객체.
Car()	: 생성자. 각 객체들을 생성한다.
boolean engineSet(int cylindernum)	: engine 객체를 이용하여 엔진의 기통수를 설정한다. 정상종료시 true, 그 외의 경우에는 false 를 리턴한다.
void engineGet()	: engine 객체를 이용하여 엔진의 기통수를 리턴한다.
void bodySet(String name)	: body 객체를 이용하여 차체의 재질을 설정한다.
void bodyGet()	: body 객체를 이용하여 차체의 재질을 리턴한다.
boolean seatSet(int number)	: seat 객체를 이용하여 시트의 갯수를 설정한다. 정상종료시 true, 그 외의 경우에는 false 를 리턴한다.
void seatGet()	: seat 객체를 이용하여 시트의 갯수를 리턴한다.
boolean speedGaugeAccel(int accel)	: speedgauge 객체를 이용하여 현재 속도를 증가한다. 정상종료시 true, 그 외의 경우에는 false 를 리턴한다.
boolean speedGaugeBreak(int breaks)	: speedgauge 객체를 이용하여 현재 속도를 감소한다. 정상종료시 true, 그 외의 경우에는 false 를 리턴한다.
boolean tireSet(int tirenum)	: tire 객체를 이용하여 타이어의 갯수를 설정한다. 정상종료시 true, 그 외의 경우에는 false 를 리턴한다.
void tireGet()	: tire 객체를 이용하여 타이어의 갯수를 리턴한다.
void vendorSet(String vendorname)	: vendor 객체를 이용하여 벤더명을 설정한다.
void venderGet()	: vendor 객체를 이용하여 벤더명을 리턴한다.

[Package Doc]

(DocBody)

String docGetBody() : 차체의 재질 설문조사를 위한 추상 메소드.

(DocEngine)

int docGetEngine() : 자동차의 엔진 기통 설문조사를 위한 추상 메소드.

(DocSeat)

int docGetSeat() : 자동차의 좌석 갯수 설문조사를 위한 추상 메소드.

(DocSpeedGauge)

`void docGetSpeedGauge()` : 자동차의 속도계 설정 설문조사를 위한 추상 메소드.

(DocTire)

`int docGetTire()` : 자동차의 바퀴 갯수 설문조사를 위한 추상 메소드.

(DocVendor)

`String docGetVendor()` : 자동차의 벤더명 설문조사를 위한 추상 메소드.

(DocCar)

: 인터페이스를 통합해주는 인터페이스.

(Doc)

`String docGetBody()` : 고객으로부터 원하는 차체의 재질을 입력받는 메소드.
입력받은 재질값을 `String` 의 형태로 리턴한다.

`int docGetEngine()` : 고객으로부터 원하는 엔진의 기통수를 입력받는 메소드.
입력받은 기통수를 `int` 의 형태로 리턴한다.

`int docGetSeat()` : 고객으로부터 원하는 좌석의 갯수를 입력받는 메소드.
입력받은 좌석수를 `int` 의 형태로 리턴한다.

`void docGetSpeedGauge()` : 고객으로부터 속도계의 옵션을 입력받는 메소드.
현재로서는 아무런 역할이 없다.

`String docGetVendor()` : 고객으로부터 원하는 벤더의 이름을 입력받는 메소드.
입력받은 벤더명을 `String` 의 형태로 리턴한다.

`int docGetTire()` : 고객으로부터 원하는 바퀴의 갯수를 입력받는 메소드.
입력받은 바퀴의 갯수를 `int` 의 형태로 리턴한다.

[Package Test]

(Speed)

int setSpeed() : 자동차의 속도를 조절하는 추상 메소드.

(TestCar) : 인터페이스를 통합해주는 인터페이스.

(Test)

int setSpeed() : 사용자로부터 원하는 속도를 입력받아 입력받은 값을 int 의 형태로 리턴한다.

[Package Car]

(Body.java)

```
package Car;
```

```
/*  
 * Body.java  
 * Written by 200301582 ST. Kim  
 * Discription : body class  
 *               create body of car.  
 */
```

```
public class Body {
```

```
    String material;
```

```
    public Body() {  
        //System.out.println("Body Class!");  
        material = null;  
    }
```

```
    void setBody(String data) {  
        material = data;  
    }
```

```
    String getBody() {  
        if(material == null) {  
            System.out.println("Choice material!");  
            return null;  
        }  
        else  
            return material;  
    }
```

```
}
```

(Car.java)

```
package Car;

/*
 * Car.java
 * Written by 200301582 ST. Kim
 * Discription : car class
 *               create all of car.
 */

public class Car {

    static Engine engine;
    static Body body;
    static Seat seat;
    static SpeedGauge speedGauge;
    static Tire tire;
    static Vendor vendor;

    public Car() {

        //System.out.println("Car Class!");

        engine = new Engine();
        body = new Body();
        seat = new Seat();
        speedGauge = new SpeedGauge();
        tire = new Tire();
        vendor = new Vendor();
    }

    boolean engineSet(int cylindernum) {
        return this.engine.setCylinder(cylindernum);
    }

    void engineGet() {
        System.out.println("engine set : " + engine.getCylinder());
    }

    void bodySet(String name) {
        this.body.setBody(name);
    }

    void bodyGet() {
        System.out.println("body set : " + body.getBody());
    }

    boolean seatSet(int number) {
        return this.seat.setSeat(number);
    }

    void seatGet() {
        System.out.println("seat set : " + this.seat.getSeat());
    }

    boolean speedGaugeAccel(int accel) {
        if(this.speedGauge.accelSpeed(accel) == false)

```

```

        return false;
    else {
        System.out.println("speedGauge : " +
this.speedGauge.getSpeed());
        System.out.println("speedGaugeDistance : " +
this.speedGauge.getDistance());
        return true;
    }
}

    boolean speedGaugeBreak(int breaks) {
        if(this.speedGauge.breakSpeed(breaks) == false)
            return false;
        else {
            System.out.println("speedGauge : " +
this.speedGauge.getSpeed());
            System.out.println("speedGaugeDistance : " +
this.speedGauge.getDistance());
            return true;
        }
    }

    boolean tireSet(int tirenum) {
        return this.tire.setTire(tirenum);
    }

    void tireGet() {
        System.out.println("tire number : " + this.tire.getTire());
    }

    void vendorSet(String vendorname) {
        this.vendor.setVendor(vendorname);
    }

    void venderGet() {
        System.out.println("vendor name : " + this.vendor.getVendor());
    }
}

```

(CarMain.java)

```
package Car;
```

```
import Doc.Doc;  
import Test.Test;
```

```
/*  
 * CarMain.java  
 * Written by 200301582 ST. Kim  
 * Discription : CarMain class  
 * just main.  
 */
```

```
public class CarMain {
```

```
    /**  
     * @param args  
     */
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        boolean a = false;
```

```
        System.out.println("안녕하세요!");  
        System.out.println("저희 정통 자동차 판매소에 오신것을 환영합니다!");  
        System.out.println("지금부터 고객님의 요구에 맞는 자동차를 제작할테니 묻는 말  
에 대해.");  
        System.out.println("대금은 일시불로 only 현찰, 가격문의는 안받습니다.\n");
```

```
        Doc doct    = new Doc();  
        Test test   = new Test();  
        Car car     = new Car();
```

```
        car.vendorSet(doct.docGetVendor());
```

```
        while(a == false) {  
            System.out.println();  
            a = car.engineSet(doct.docGetEngine());  
        }  
        a = false;
```

```
        System.out.println();  
        car.bodySet(doct.docGetBody());
```

```
        System.out.println();  
        doct.docGetSpeedGauge();
```

```
        while(a == false){  
            System.out.println();  
            a = car.tireSet(doct.docGetTire());  
        }  
        a = false;
```

```
        System.out.println();  
        System.out.println("지금부터 주문하신 내역에 대한 정리를 해 드리겠습니다.  
\n");
```

```
        car.venderGet();  
        car.engineGet();  
        car.bodyGet();  
        car.tireGet();
```

```
System.out.println();
System.out.println("지금부터 간단한 시승 Test 를 하겠습니다.");
System.out.println("먼저 엑셀을 밟아 보세요.");
System.out.println();

while(a == false) {
    a = car.speedGaugeAccel(test.setSpeed());
}
a = false;

System.out.println("잘 나가는군요. 그럼 이번엔 브레이크를 밟아 보세요.");
System.out.println();

while(a == false) {
    a = car.speedGaugeBreak(test.setSpeed());
}

System.out.println();
System.out.println("마음에 드시나요?");
System.out.println("새로운 자동차의 오너가 되신것을 축하드립니다!");
```

```
}
```

```
}
```

(Engine.java)

```
package Car;
```

```
/*
```

```
 * Engine.java
```

```
 * Written by 200301582 ST. Kim
```

```
 * Discription : engine class
```

```
 *           create engine of car.
```

```
*/
```

```
public class Engine {
```

```
    int cylinder;
```

```
    // number
```

```
of cylinder
```

```
    int max_cylinder;
```

```
    // number of
```

```
maxium cylinder
```

```
    public Engine() {
```

```
    // constructor
```

```
        //System.out.println("Engine class!");
```

```
        max_cylinder = 10;
```

```
        cylinder = 0;
```

```
    }
```

```
    public boolean setCylinder(int number) {
```

```
    // setting of
```

```
cylinder number
```

```
        if(number > max_cylinder) {
```

```
            System.out.println("too many cylinder. please choice 1 ~" +
```

```
max_cylinder);
```

```
            return false;
```

```
        }
```

```
        else if(number <= 0) {
```

```
            System.out.println("too low cylinder. please choice 1 ~" +
```

```
max_cylinder);
```

```
            return false;
```

```
        }
```

```
        else {
```

```
            cylinder = number;
```

```
            return true;
```

```
        }
```

```
    }
```

```
    public int getCylinder() {
```

```
    // getting of
```

```
cylinder number
```

```
        return cylinder;
```

```
    }
```

```
}
```

(Seat.java)

```
package Car;

/*
 * Seat.java
 * Written by 200301582 ST. Kim
 * Discription : seat class
 *               create seat of car.
 */

public class Seat {

    int seat_number; // number of
seat

    public Seat() { //
constructor
    // System.out.println("Seat Class!");
    seat_number = 0;
    }

    boolean setSeat(int number) { // setting of
seat number
    if(number < 0) {
        System.out.println("알맞은 숫자를 입력하세요. (0 ~ 10)");
        return false;
    }
    else if(number > 10) {
        System.out.println("알맞은 숫자를 입력하세요. (0 ~ 10)");
        return false;
    }
    seat_number = number;
    return true;
    }

    int getSeat() { //
getting of seat number
    return seat_number;
    }
}
```

(SpeedGauge.java)

```
package Car;

/*
 * SpeedGauge.java
 * Written by 200301582 ST. Kim
 * Discription : speedGauge class
 *               create speedgauge of car.
 */

public class SpeedGauge {                                // constructor

    int max_speed;                                       // max of
speed                                                    // current
    int current_speed;                                   //
speed                                                    //
    int distance;                                        //
cureent distance

    public SpeedGauge() {                                // constructor
        //System.out.println("SpeedGuage Class!");
        max_speed = 300;
        current_speed = 0;
        distance = 0;
    }

    boolean accelSpeed(int speed) {                     // setting of
accel
        if(speed > max_speed) {
max_speed);
            System.out.println("too high speed. please choice 0 ~ " +
                return false;
            }
            else if(speed <= 0) {
                System.out.println("please. this car is only go.");
                return false;
            }
            else if(speed <= current_speed) {
                System.out.println("please. use breakspeed()");
                return false;
            }
            else {
                for(int i = 0; current_speed < speed; current_speed++) {
                    distance++;
                }
                return true;
            }
        }

    boolean breakSpeed(int speed) {                     // setting of
break
        if(speed >= current_speed) {
            System.out.println("please. use accelspeed()");
            return false;
        }
        else if(speed < 0) {
            System.out.println("please. this car is only go.");
            return false;
        }
        else {
```



```
        distance += current_speed - speed;
        current_speed = speed;
        return true;
    }
}

int getSpeed() { // getting of
current speed
    return current_speed;
}

int getDistance() {
    return distance;
}
}
```

(Tire.java)

```
package Car;

/*
 * Tire.java
 * Written by 200301582 ST. Kim
 * Discription : tire class
 *               create tire of car.
 */

public class Tire {

    int tire_number;           // number of
    tire                       // number
    int max_tire;             of maxium tire

    public Tire() {           //
    constructor
        // System.out.println("Tire Class!");
        tire_number = 0;
        max_tire = 10;
    }

    boolean setTire(int number) { // setting of
    tire number
        if(number > max_tire) {
            System.out.println("too many tire. please choice 4 ~ " +
    max_tire);
            return false;
        }
        else if(number < 4) {
            System.out.println("too low tire. please choice 4 ~ " +
    max_tire);
            return false;
        }

        tire_number = number;
        return true;
    }

    int getTire() {           //
    setting of tire number
        if(tire_number == 0) {
            System.out.println("please. make tire!");
            return 0;
        }
        else
            return tire_number;
    }
}
```

(Vendor.java)
package Car;

```
/*  
 * Vendor.java  
 * Written by 200301582 ST. Kim  
 * Discription : vendor class  
 *               create vendor of car.  
 */
```

```
public class Vendor {  
    String vendor_name;  
  
    public Vendor() {  
        //System.out.println("Vendor Class!");  
        vendor_name = null;  
    }  
  
    void setVendor(String name) {  
        vendor_name = name;  
    }  
  
    String getVendor() {  
        if(vendor_name == null) {  
            System.out.println("please. make vendor!");  
            return null;  
        }  
        else  
            return vendor_name;  
    }  
}
```

[Package Doc]

(Doc.java)

```
package Doc;

import java.util.Scanner;

public class Doc implements DocCar {

    Scanner s = new Scanner(System.in);
    String temp;
    String check;

    public String docGetBody() {

        System.out.println("원하시는 차체의 재질을 입력하세요.");
        System.out.print("차체의 재질 : ");
        try {
            temp = s.next();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return temp;
    }

    public int docGetEngine() {

        int cylinder = 0;

        System.out.println("원하시는 엔진의 기통수를 입력하세요.");
        System.out.print("기통수 : ");
        try {
            check = s.next();
            cylinder = Integer.parseInt(check);
        } catch (Exception e) {
            System.out.println("올바른 숫자를 입력해 주세요.");
        }

        return (int)(cylinder);
    }

    public int docGetSeat() {

        int seat = 0;

        System.out.println("원하시는 시트의 갯수를 입력하세요.");
        System.out.print("시트수 : ");
        try {
            check = s.next();
            seat = Integer.parseInt(check);
        } catch (Exception e) {
            System.out.println("올바른 숫자를 입력해 주세요.");
        }

        return seat;
    }
}
```

```

public void docGetSpeedGauge() {
    System.out.println("최송합니다.");
    System.out.println("속도계에 대해서는 사용자가 조정할 수 있는 부분이 없습니
다.");
}

public String docGetVendor() {
    int a = 0;
    int i = 1;
    while(i > 0) {

        System.out.println("원하시는 벤더명을 입력하세요.");
        System.out.println("1.HyunDie, 2.Ventz, 3.SsanSung, 4.ETC...
");
        System.out.print("어느것을 원하세요? 번호선택 : ");

        try {
            check = s.next();
            a = Integer.parseInt(check);
            //a = s.nextInt();
        } catch (Exception e) {
            System.out.println("올바른 숫자를 입력해주세요.");
        }

        if(a == 1) {
            temp = "HyunDie";
            break;
        }
        else if(a == 2) {
            temp = "Ventz";
            break;
        }
        else if(a == 3) {
            temp = "SsanSung";
            break;
        }
        else if(a == 4) {
            System.out.println("'아무거나'를 선택하셨네요!");
            System.out.println("벤더명은 '아무거나' 입니다!");

            temp = "아무거나";
            break;
        }
        else {

            if( i == 1) {
                System.out.println("엉뚱한짓 말고 번호나 찍으라구!");
                i++;
            }
            else
                System.out.println("형아 화낸다?");
        }
    }

    return temp;
}

public int docGetTire() {

```

```
Integer tire = 0;

System.out.println("원하시는 바퀴의 갯수를 입력하세요.");
System.out.print("바퀴수 : ");
try {
    check = s.next();
    tire = Integer.parseInt(check);
} catch (Exception e) {
    System.out.println("올바른 숫자를 입력해주세요.");
}

return tire;

}

}
```

```
(DocBody.java)
package Doc;

public interface DocBody {                                // DocBody interface.
    abstract String docGetBody();                       // get the Body. and return it.
}

```

```
(DocCar.java)
package Doc;

public interface DocCar extends DocBody, DocEngine, DocSeat, DocSpeedGauge,
DocVendor, DocTire {
}

```

```
(DocEngine.java)
package Doc;

public interface DocEngine {                            // DocEngine interface.
    int docGetEngine();                                // get the engine cylinder.
and return it.
}

```

```
(DocSeat.java)
package Doc;

public interface DocSeat {                             // DocSeat interface.
    int docGetSeat();                                 // get Seat number. and return it.
}

```

```
(DocSpeedGauge.java)
package Doc;

public interface DocSpeedGauge {
    void docGetSpeedGauge();
}

```

(DocTire.java)

```
package Doc;

public interface DocTire {

    int docGetTire();

}
```

(DocVendor.java)

```
package Doc;

public interface DocVendor {                                // DocVendor interface.

    abstract String docGetVendor();                        // get the vendorname. and
return it.

}
```


[Package Test]

(Test.java)

```
package Test;

import java.util.Scanner;

public class Test implements TestCar {

    Scanner s = new Scanner(System.in);
    String check;

    public int setSpeed() {
        int speed = 0;

        System.out.println();
        System.out.println("원하시는 속도를 입력하세요.");
        System.out.print("SPEED : ");
        try {
            check = s.next();
            speed = Integer.parseInt(check);
        } catch (Exception e) {
            System.out.println("올바른 속도를 입력하세요. (0 ~ 300)");
            return -1;
        }

        return speed;
    }

}
```

(TestCar.java)

```
package Test;

public interface TestCar extends Speed{

}
```

(Speed.java)

```
package Test;

public interface Speed {

    int setSpeed();

}
```