

03

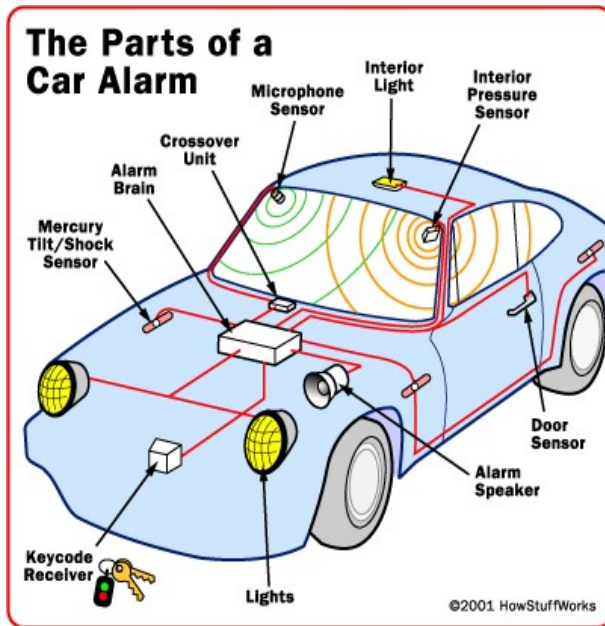
03-1. 객체와 클래스

Java 객체지향

객체지향 프로그래밍

- 객체지향 프로그래밍(Object-Oriented Programming: OOP)
- 객체(Object)
 - 사람, 장소, 사물 등
 - 객체는 상태(State)를 가짐 – 인스턴스변수 (Instance Variable)
 - 객체는 행동(Behavior)를 가짐 – 메소드(Method)
- 클래스(Class)
 - 객체(Object)를 생성하기 위한 형판(型板; Template)
 - 하나의 클래스는 같은 상태와 행동을 가진 다수의 객체를 생성함
 - 하나의 객체(Object)는 해당 클래스(Class)의 구체적인 사례(Instance)

객체와 클래스 – 클래스(Class)

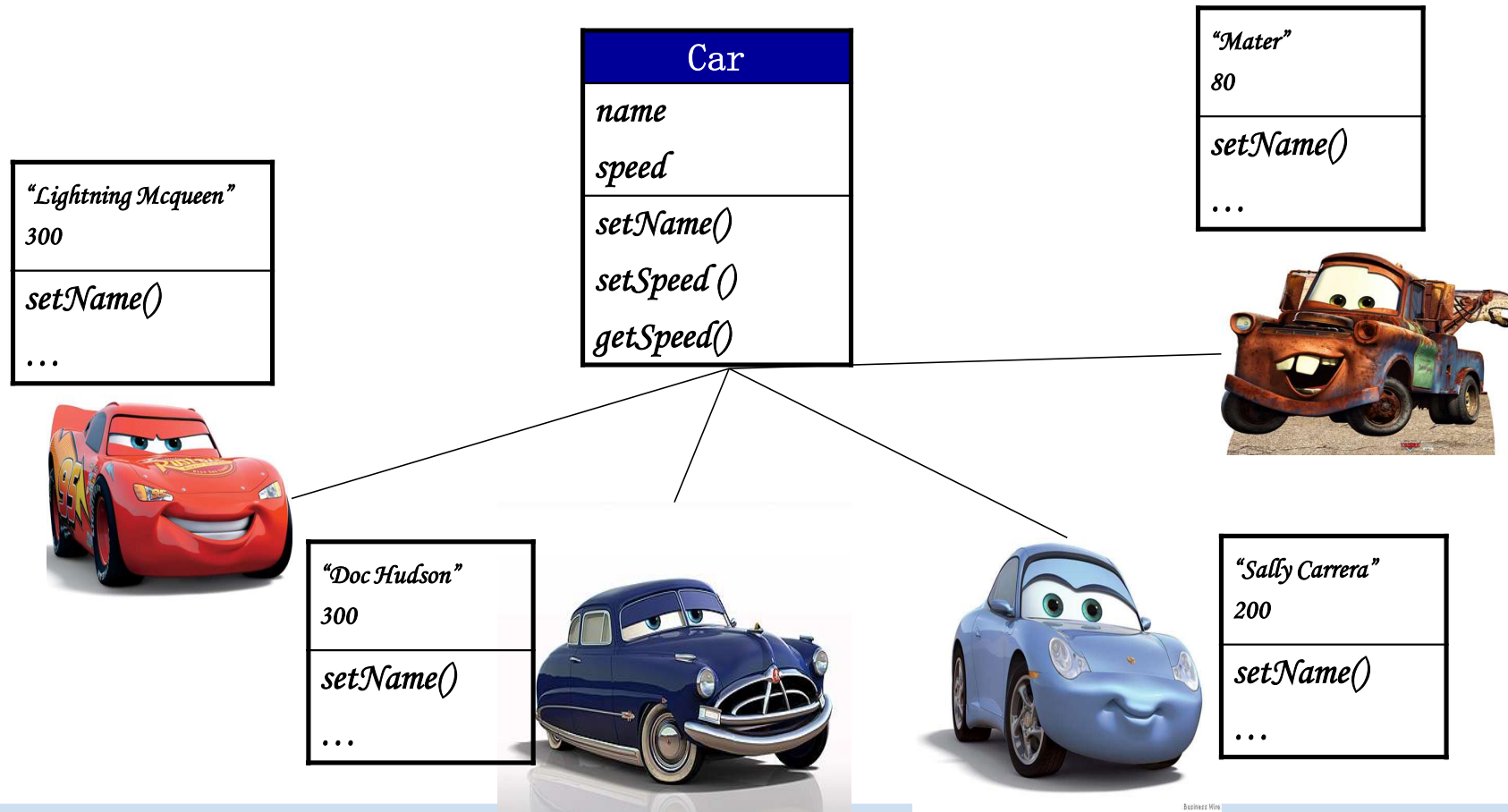


상태(State):
Instance Variable

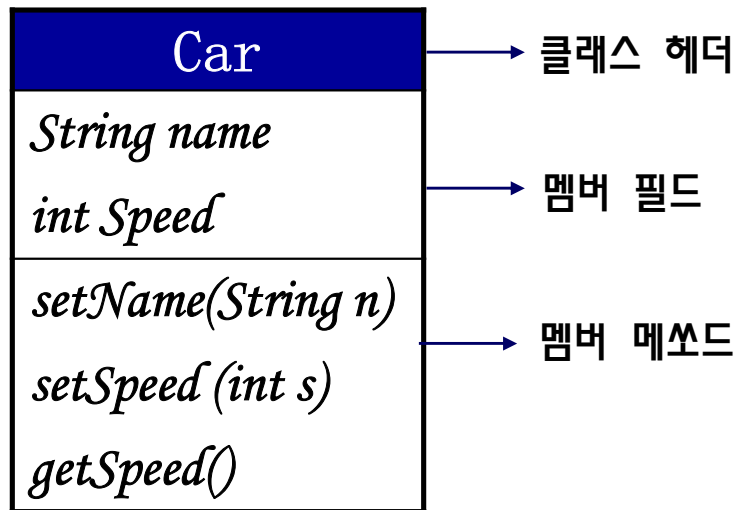
행동(Behavior):
Method

Car	
<i>name</i>	<i>knows</i>
<i>speed</i>	
<i>setName()</i>	<i>does</i>
<i>setSpeed()</i>	
<i>getSpeed()</i>	

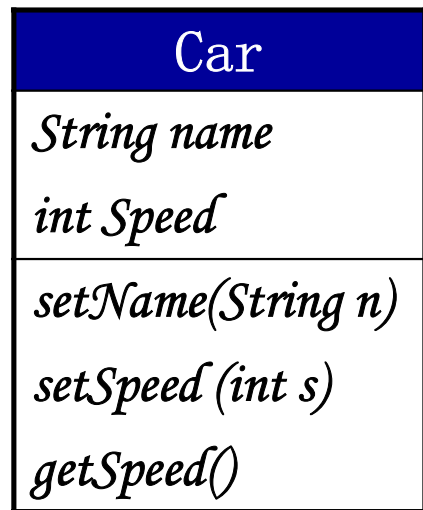
객체와 클래스 – One Class & Multiple Object



객체(Object) 생성 – 클래스 설계



객체(Object) 생성 – 클래스 선언



클래스 헤더

멤버 필드

멤버 메소드

```
class Car {
```

```
    String name;
```

```
    int speed;
```

```
    public void setName(String n) {  
        name = n;  
    }
```

```
    public void setSpeed(int s) {  
        speed = s;  
    }
```

```
    public int getSpeed() {  
        return speed;  
    }
```

```
}
```

객체(Object) 생성 – 객체 생성

Car
<i>name</i>
<i>speed</i>
<i>setName()</i>
<i>setSpeed()</i>
<i>getSpeed()</i>

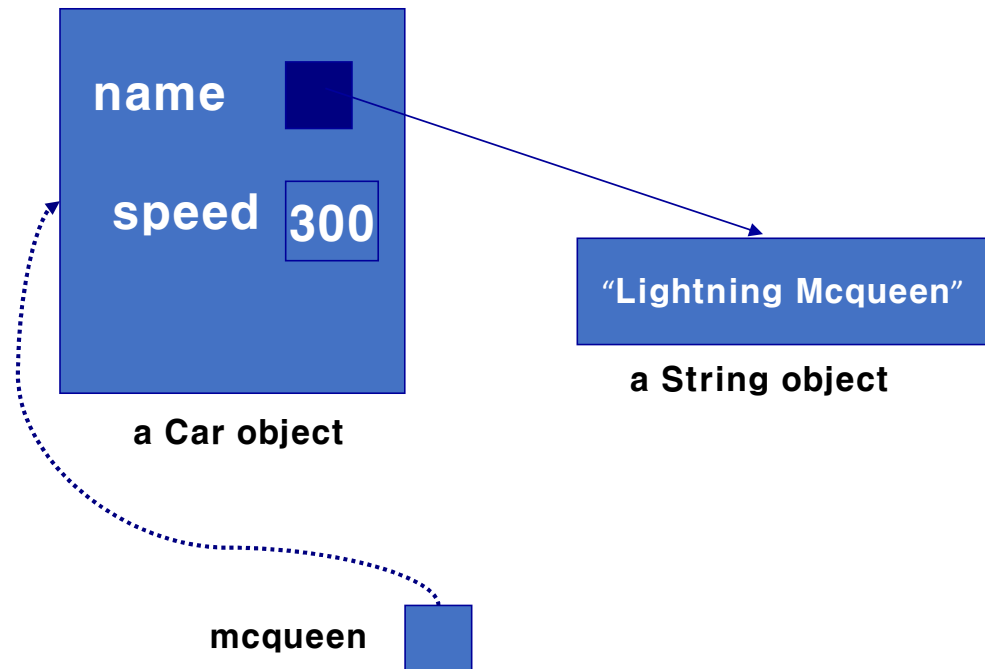
```
public class CarTest {  
    public static void main(String[] args)  
    {  
        Car mcqueen = new Car();  
  
        mcqueen.setName("Lightning Mcqueen");  
        mcqueen.setSpeed(300);  
  
        System.out.println(mcqueen.name + " " +  
                             mcqueen.getSpeed());  
    }  
}
```

<i>mcqueen</i>
<i>"Lightning Mcqueen"</i>
300
<i>setName()</i>
...



객체(Object) 생성 – 객체 생성

```
public class CarTest {  
    public static void main(String[] args)  
    {  
        Car mcqueen = new Car();  
  
        mcqueen.setName( "Lightning Mcqueen" );  
        mcqueen.setSpeed(300);  
  
        System.out.println(mcqueen.name + " " +  
            mcqueen.getSpeed());  
    }  
}
```



자바의 자료형 - 기본 자료형 vs. 참조 자료형

```
int myInt = 19;
```

myInt **19**

```
String myStr = new myString();
```

①

③

②

myStr

①

③

②

a String object



자바의 자료형 – 기본 자료형 vs. 참조 자료형

	기본 자료형	참조 자료형
변수값	실제값	Object 참조값
정의방식	Java 내부에 이미 정의됨	클래스 정의
생성방식	"19", "3.14", "true"	"new"*
초기화방식	default	생성자(Constructor)*
사용방식	연산자 ("+", "-", "*", "/" 등)	메쏘드

클래스 선언 및 객체 생성

Car
double KILOMETERS_PER_LITER double kilometerDriven double lightOil
Car() Car(double)
getKilometersPerLiter() getKilometersDriven() getLiteOil() addLiteOil() drive()

- ❑ KILOMETERS_PER_LITER
 - 일단 초기화된 후에는 수정이 불가능한 상수로 선언할 것
 - 연비
- ❑ kilometerDriven
 - 주행거리
- ❑ lightOil
 - 현재 주유되어 있는 기름
- ❑ Car()
 - 연비 30, 주행거리 및 기름 0
- ❑ Car(double)
 - 주어진 연비가 0보다 클 경우 연비 세팅
- ❑ addLiteOil(double)
 - 기름 주유
- ❑ drive()
 - 주행하려는 거리보다 주행가능한 거리가 크다면 false 를 리턴하고 , 운행하지 않는다는 메시지를 띄움
 - 주행 가능한 거리이면, 주행거리를 증가시키고 잔여 기름은 감소시킨 후 true를 리턴함



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare