



Power Java

제7장 객체지향소개



© 2009 인피니티박스 All rights reserved



이번 장에서 학습할 내용

- 객체지향이란?
- 객체
- 메시지
- 클래스
- 객체 지향의 장점
- String 클래스

객체 지향
개념을
완벽하게
이해해야만
객체 지향
설계의 이점을
활용할 수 있다.



© 2009 인피니티박스 All rights reserved



객체 지향이란?

- 실제 세계를 모델링하여 소프트웨어를 개발하는 방법

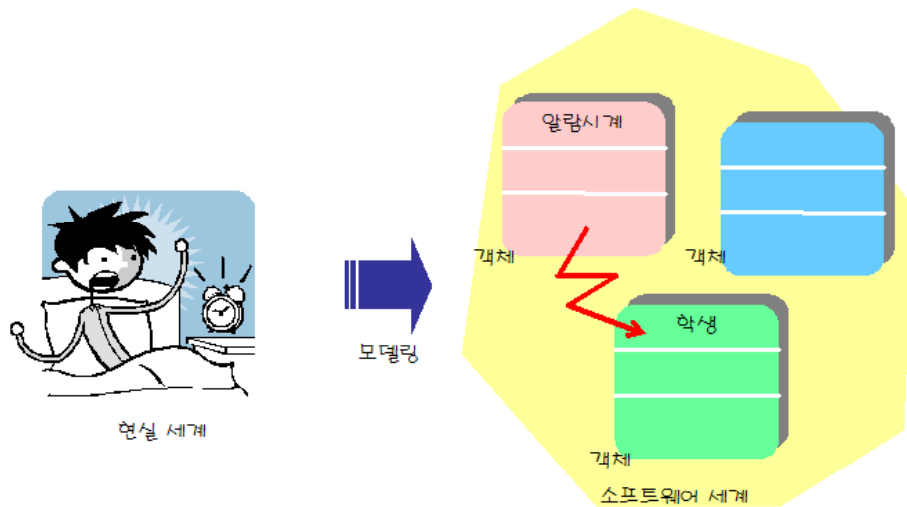


그림 7.1 객체 지향 방법은 현실 세계를 모델링하는 것

© 2009 인피니티박스 All rights reserved



절차 지향과 객체 지향

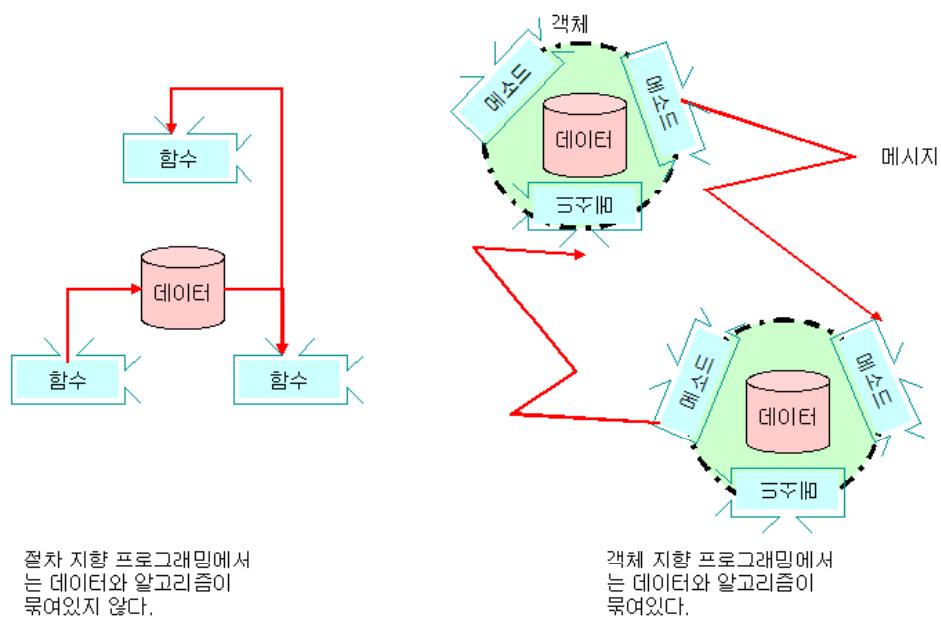


그림 7.2 절차 지향적인 방법과 객체 지향적인 방법의 비교

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 객체 지향 프로그래밍은 _____들을 조합하여서 프로그램을 작성하는 기법이다.
2. 객체 지향 프로그래밍의 시작은 _____년대에 개발된 _____언어이다.

© 2009 인피니티박스 All rights reserved



객체란?

- 객체(Object)는 상태와 동작을 가지고 있다.
- 객체의 상태(state)는 객체의 특징값(속성)이다.
- 객체의 동작(behavior) 또는 행동은 객체가 취할 수 있는 동작

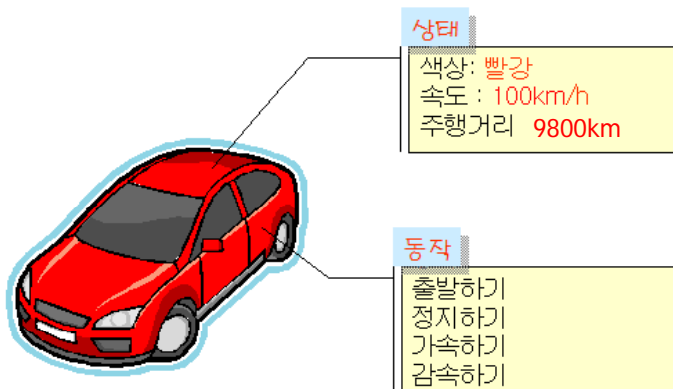


그림 7.3 자동차 객체의 예

© 2009 인피니티박스 All rights reserved



중간 점검 문제



중간점검

1. 다음과 같은 실제 세계의 객체에서 객체의 가능한 상태와 객체가 수행할 수 있는 동작을 정리하여 보자.

객체	상태	동작
전구		
라디오		
강아지		
자전거		
사자		



메시지

- 소프트웨어 객체는 메시지(message)를 통해 다른 소프트웨어 객체와 통신하고 서로 상호 작용한다.

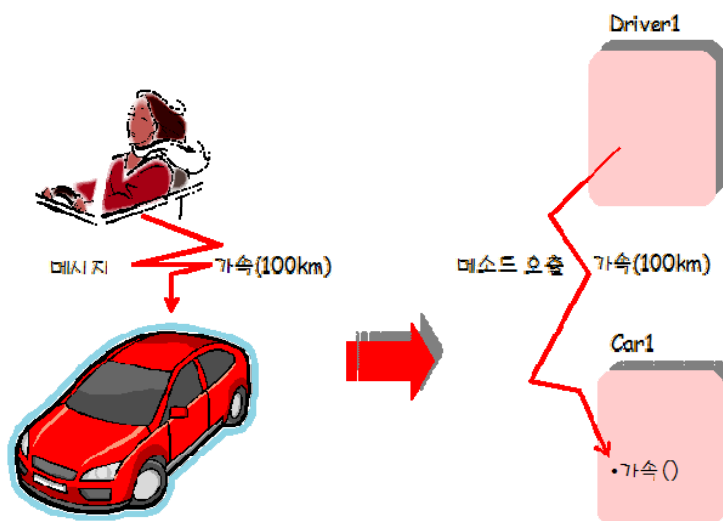


그림 7.5 메시지 전달



중간 점검 문제

1. 객체들은 _____전달을 통해서 서로 간에 상호 작용을 한다.
2. 자동차 객체에서 생각할 수 있는 메시지와 매개 변수에 대하여 나열하여 보라.

© 2009 인피니티박스 All rights reserved



클래스

- 클래스(class): 객체를 만드는 설계도
- 클래스로부터 만들어지는 각각의 객체를 특별히 그 클래스의 인스턴스(instance)라고도 한다.

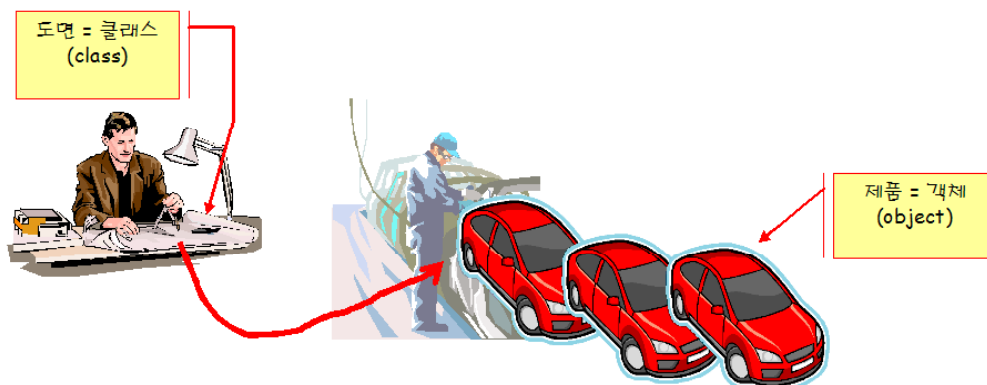


그림 7.6 객체를 클래스라는 설계도로 생성된다.

© 2009 인피니티박스 All rights reserved



자동차 클래스

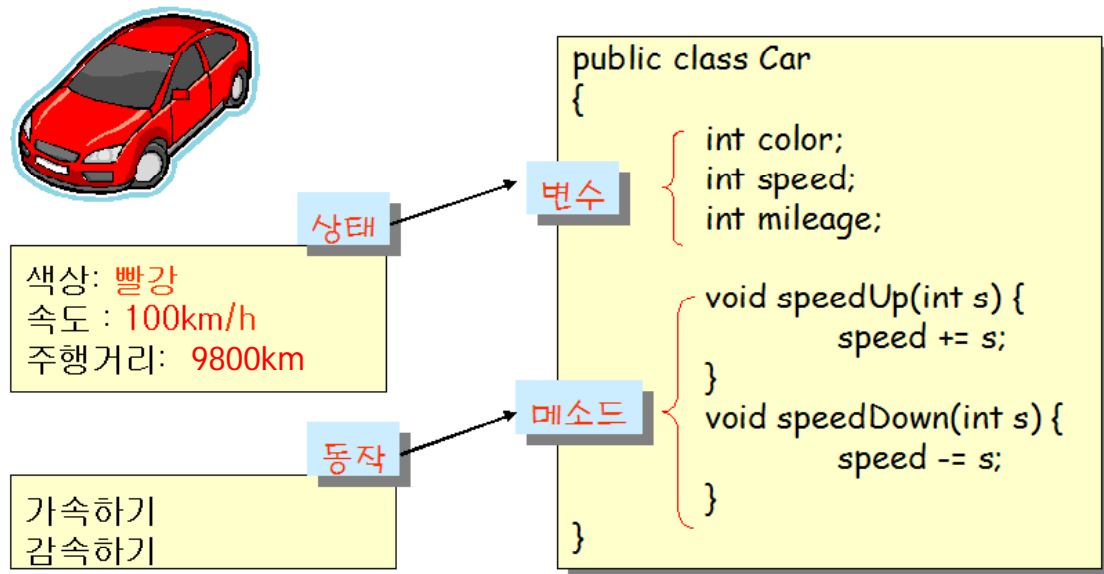


그림 7.7 Car 클래스

© 2009 인피니티북스 All rights reserved



소프트웨어 작성이 쉽다

- 부품을 구입하여 컴퓨터를 조립하듯이 소프트웨어를 작성할 수 있다.

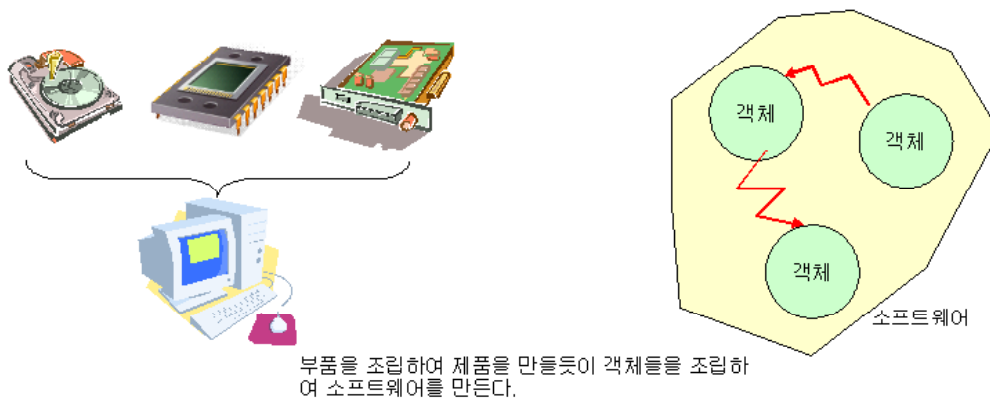


그림 7.8 객체 지향 방법

© 2009 인피니티북스 All rights reserved



객체 지향의 장점

- 신뢰성있는 소프트웨어를 쉽게 작성할 수 있다.
- 코드를 재사용하기 쉽다.
- 업그레이드가 쉽다.
- 디버깅이 쉽다.

© 2009 인피니티박스 All rights reserved



코드의 재사용

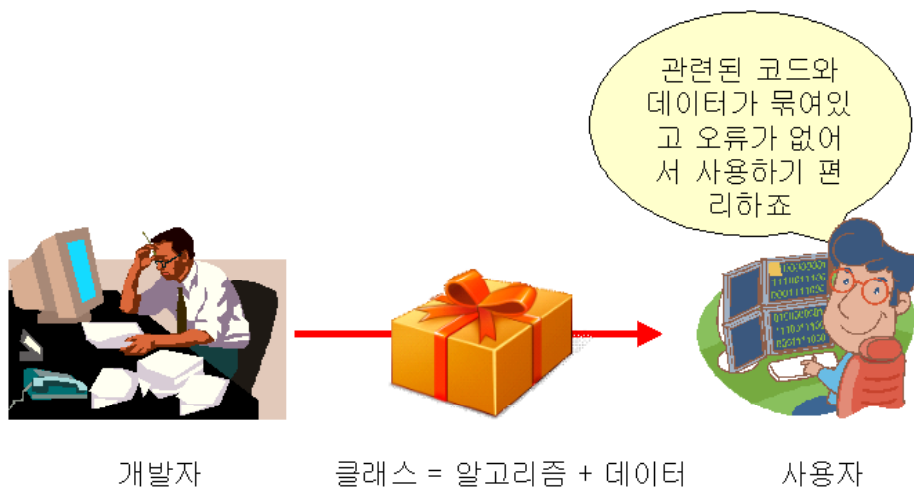


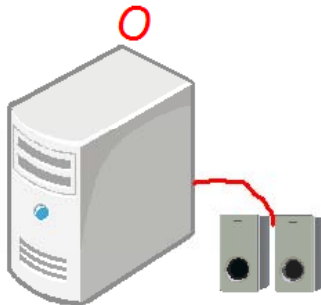
그림 7.9 객체 지향을 사용하면 코드를 재사용할 수 있다.

© 2009 인피니티박스 All rights reserved

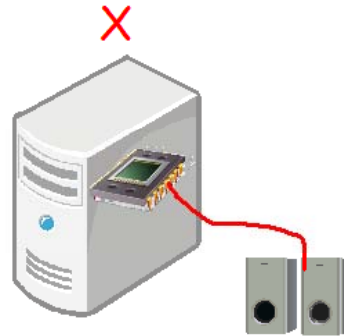


업그레이드가 쉽다.

- 라이브러리가 업그레이드되면 쉽게 바꿀 수 있다.
- 정보 은닉이 가능하기 때문에 업그레이드 가능



만약 외부의 표준 오디오 단자를 이용하였으면 내부의 사운드 카드를 변경할 수 있다.



만약 내부의 오디오 제어 칩의 단자에 연결하였으면 내부의 사운드 카드를 변경할 수 없다.

그림 7.10 정보 은닉의 장점

© 2009 인피니티박스 All rights reserved



쉬운 디버깅

- 예를 들어서 절차 지향 프로그램에서 하나의 변수를 1000개의 함수가 사용하고 있다고 가정해보자. -> 하나의 변수를 1000개의 함수에서 변경할 수 있다.
- 객체 지향 프로그램에서 100개의 클래스가 있고 클래스당 10개의 메소드를 가정해보자. -> 하나의 변수를 10개의 메소드에서 변경할 수 있다.
- 어떤 방법이 디버깅이 쉬울까?

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 자바에서 코드 재사용이 쉬운 이유는 관련된 _____와 _____이 하나의 덩어리로 묶여 있기 때문이다.
2. 정보 은닉이란 _____을 외부로부터 보호하는 것이다.
3. 정보를 은닉하면 발생하는 장점은 무엇인가?

© 2009 인피니티박스 All rights reserved



문자열 객체

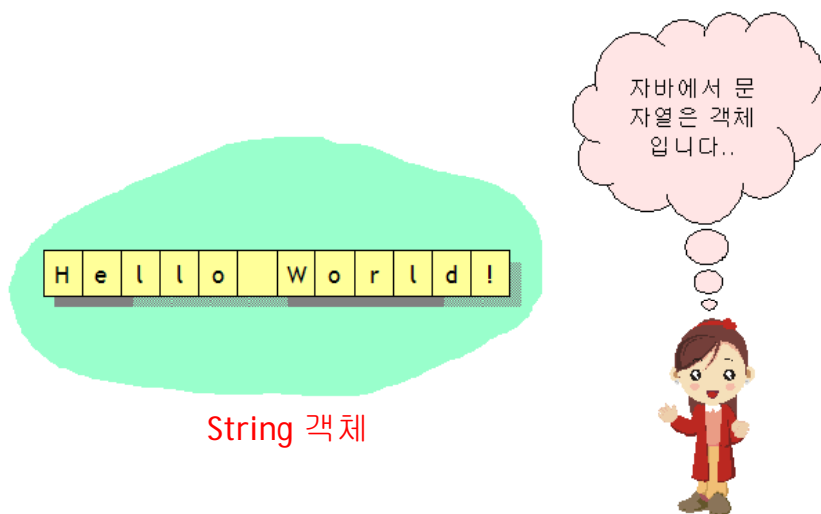


그림 7.11 문자열은 객체이다.

© 2009 인피니티박스 All rights reserved



클래스에서 객체를 생성하는 방법

- 단 하나의 방법만이 존재한다.
- `String s = new String("Hello World");`

new
연산자만이
객체를
생성합니다.



© 2009 인피니티박스 All rights reserved



기초 변수와 참조 변수

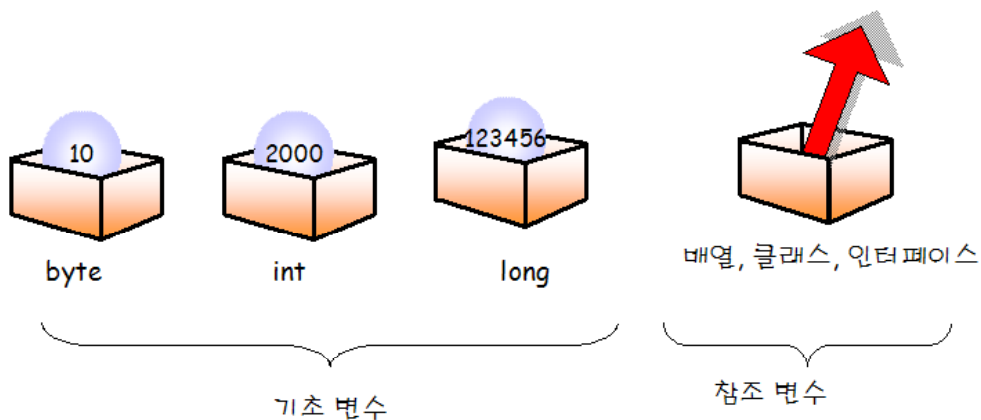


그림 7.12 변수의 종류

© 2009 인피니티박스 All rights reserved



기초 변수와 참조 변수의 비교

```
int i; // 기초 변수
String s; // 참조 변수
```

위와 같이 정의하면 두 변수 모두 처음에는 데이터를 담고 있지 않다. 즉 초기화가 되지 않은 상태이다.



기초 변수



참조 변수

```
i = 10; // 기초 변수에 값을 대입
s = new String("Hello World!"); // 객체를 생성하고 참조 변수에 객체의 주소를 대입
```



기초 변수



참조 변수

He l l o W o r l d !

객체

© 2009 인피니티박스 All rights reserved



객체의 메소드의 호출

- String s = "Hello World!";
- int size = s.length(); // size는 12가 된다.

.(도트)
연산자를
사용하여서
메소드를
호출합니다.



© 2009 인피니티박스 All rights reserved



String 클래스의 메소드

메소드 요약	
char	charAt(int index) 지정된 인덱스에 있는 문자를 반환한다.
int	compareTo(String anotherString) 사전적 순서로 문자열을 비교한다. 앞에 있으면 -1, 같으면 0, 뒤에 있으면 1이 반환된다.
String	concat(String str) 주어진 문자열을 현재의 문자열 뒤에 붙인다.
boolean	equals(Object anObject) 주어진 객체와 현재의 문자열을 비교한다.
boolean	equalsIgnoreCase(String anotherString) 대소문자를 무시하고 비교한다.
boolean	isEmpty() length()가 0이면 true를 반환한다.
int	length() 현재 문자열의 길이를 반환한다.
String	replace(char oldChar, char newChar) 주어진 문자열에서 oldChar를 newChar로 변경한, 새로운 문자열을 생성하여 반환한다.
String	substring(int beginIndex, int endIndex) 현재 문자열의 일부를 반환한다.
String	toLowerCase() 문자열의 문자들을 모두 소문자로 변경한다.
String	toUpperCase() 문자열의 문자들을 모두 대문자로 변경한다.

© 2009 인피니티북스 All rights reserved



메소드 사용의 예

StringTest.java

```

public class StringTest
{
    public static void main (String[] args)
    {
        String proverb = "A barking dog";           // new 연산자 생략
        String s1, s2, s3, s4;                       // 참조 변수로서 메소드에서 반환된 참조값을 받는다.

        System.out.println ("문자열의 길이 = " + proverb.length());

        s1 = proverb.concat (" never Bites!");       // 문자열 결합
        s2 = proverb.replace ('B', 'b');             // 문자 교환
        s3 = proverb.substring (2, 5);               // 부분 문자열 추출
        s4 = proverb.toUpperCase();                  // 대문자로 변환

        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s3);
        System.out.println(s4);
    }
}

```

출력결과

```

문자열의 길이 =13
A barking dog never Bites!
A barking dog
bar
A BARKING DOG

```

© 2009 인피니티북스 All rights reserved



중간 점검 문제

1. 변수를 크게 두 가지로 나누면 _____ 변수와 _____ 변수로 분류할 수 있다.
2. 객체를 생성하는 키워드는 _____이다.
3. 문자열은 클래스 _____의 객체이다.
4. 문자열의 길이를 반환하는 메소드는 _____이다.
5. 변수 i의 값을 문자열로 변환하는 가장 간단한 방법은 ____ 연산자를 사용하는 것이다.

© 2009 인피니티박스 All rights reserved



랩퍼 클래스

- 기초 자료형을 객체로 포장하여 주는 클래스

```
Integer obj = new Integer(10);
```

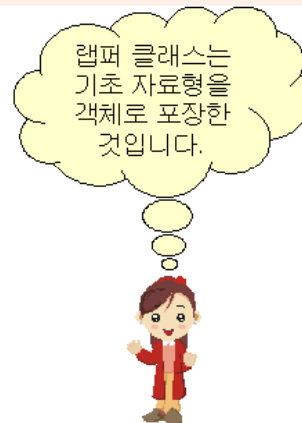


그림 7.14 랩퍼 클래스의 개념

© 2009 인피니티박스 All rights reserved



Q & A

