

# Computer Programming

## CNU, NAOE, RUY

10장 참조형



- 참조형의 주요한 특징을 이해하고 응용해서 사용한다
- 서로 다른 형식 사이의 형 변환의 가능성을 결정할 수 있다
- 참조형 사이의 데이터 형 변환을 이해할 수 있다
- 값형과 참조형 사이의 박싱과 언박싱 변환을 이해하고 변환할 수 있다.



## 참조형 변수의 선언과 해제

- 변수는 스택영역에 저장되고, 실제 데이터는 힙영역에 저아된다.
- 스택에 저장되는 값은 실제 데이터가 저장된 힙영역의 주소값이다.
- 메모리의 해제 방법은 참조형 변수에 null을 할당하면 된다.

### 참조형 변수선언 및 메모리할당

참조형 변수 c를 선언하고 실제 데이터가 저장될 힙 영역의 메모리 확보는 new 키워드를 통해서 이루어 지며, 생성자 (Coordinate())를 통해서 초기화된다.

### <참조형 선언및 멤버 접근 예제>

```
class Coordinate
{
    public int x = 0 ;
}
...
Coordinate c = new Coordinate() ;

c.x = 1;
```

### 참조형의 메모리 해제

null을 참조형 변수에 할당하면 확보되어 있던 메모리가 해제된다.

### <참조형 메모리해제 예제>

```
Coordinate c = new Coordinate() ;

c = null ;
```

## ● 값형과 참조형의 비교

- 값형의 경우 ==와 !=의 연산자로 값을 비교한다.
- 참조형의 값은 ==와 !=로 비교할 수 없다.
- ==혹은 !=연산자는 단지 참조형 변수의 힙영역 주소값 만을 비교하게 된다.

두 객체의 멤버에 같은 값을 할당한다.

두 객체의 저장된 값은 같으나  
참조형 데이터 값은  
== 연산자로 비교할 수  
없다. 굳이 비교를 하게 되면 서로 다른 값으로 인식한다.

<예제>

```
class Coordinate
{
    public int x ;
}
...
Coordinate c1 = new Coordinate();
Coordinate c2 = new Coordinate();

c1.x = 1 ;
c2.x = 1 ;

if(c1 == c2)
    Console.WriteLine("c1==c2");
else
    Console.WriteLine("c1!=c2");
```

### ● 같은 객체에 대한 다중 참조

- 두 참조형 변수가 같은 객체의 주소를 가지고 있는 경우
- 참조변수하나의 데이터를 변경하면 다른 참조변수를 통해 얻은 값도 변경된 값을 읽게 된다

참조형 변수 c2는 새로운 힙영역 메모리를 확보하지 않고 c1의 저장된 내용을 참조하게 된다

두 참조형 변수는 결국 같은 메모리 영역의 주소를 저장하고 있으므로 c1과 c2는 같은 값을 갖게 된다.

<예제>

```
class Coordinate
{
    public int x ;
}
...
Coordinate c1 = new Coordinate();
Coordinate c2 ;

c1.x = 1 ;

c2 = c1 ; //복사

if(c1 == c2)
    Console.WriteLine("c1==c2");
else
    Console.WriteLine("c1!=c2");
```

## ● 기반클래스와 파생 클래스형 간의 형 변환

- 파생클래스->기반클래스로의 변환은 암시적 가능
- 기반클래스->파생클래스로의 변환은 명시적으로 변환해야 가능
- 데이터 호환이 되지 않을 때는 `InvalidCastException` 발생

파생 클래스형을 기반 클래스형으로 형 변환  
암시적으로 형 변환됨

기반 클래스형을 파생 클래스형으로 형 변환  
명시적으로 형 변환해야 함

<예제>  

```
class Vehicle
{
}
class Car : Vehicle
{
}
...

Vehicle veh = new Vehicle();
Car car = new Car();

Vehicle a = car ;

Car b = (Car) a ;
```

## is 연산자

데이터 형 사이의 호환성을 테스트하는 데 사용

car 참조형 변수가 Vehicle형과 호환 가능한 형태인지 조사  
호환가능하면 true, 호환가능하지 않으면 false return

<예제>

```
class Vehicle
{
}
class Car : Vehicle
{
}
...
```

```
Vehicle veh = new Vehicle();
Car car = new Car();
```

```
if( car is Vehicle)
{
    ...
}else
    ...
```

## as 연산자

- 캐스트 연산과 비슷
- 형변환이 가능한 형인지 호환성 조사 후에 형 변환된 값 반환

참조형 변수 car를 Vehicle형으로 변환

형 변환 가능한지 호환성 검사 후에 변환된 값을 넘긴다. 만약 형 변환이 가능치 않으면 null을 반환하고 예외를 발생시키지 않는다.

<예제>

```
class Vehicle
{
}
class Car : Vehicle
{
}
...
```

```
Vehicle veh = new Vehicle();
Car car = new Car();
```

```
veh = car as Vehicle ;
```



## ● 박싱과 언박싱

- 값형->참조형 : 박싱(boxing)
- 참조형->값형 : 언박싱(unboxing)

### 박싱(boxing)

정수형(값형) 변수 x를 참조형 변수 o에 할당하게 되면 암시적으로 x는 참조형으로 형변환되게 된다.

### 언박싱(unboxing)

정수형 변수 y에 참조형 변수를 할당하려고 하면 명시적으로 캐스팅해서 참조형을 정수형으로 변환하게 된다.

### <예제>

```
int x = 123 ;
```

```
object o = x ;
```

```
int y = (int) o ;
```