

# 4부 상속

## - 17장 레퍼런스 형 변환

최문환



# 16장 상속

1. 업 캐스팅과 다운 캐스팅 슈퍼 클래스와 서브 클래스
2. 업캐스팅과 메서드 오버라이딩

# 1. 업 캐스팅과 다운 캐스팅

일반 기본 자료형에서 형 변환

```
int a=10;
```

```
long b=20;
```

```
b=a; //일반 변수들 사이의 암시적인 형 변환
```

```
a=(int)b; //일반 변수들 사이의 명시적인 형 변환
```

# 1. 업 캐스팅과 다운 캐스팅

레퍼런스 변수들 사이의 형 변환은 상속관계에서만 가능합니다.

상속관계에 있는 클래스 사이의 형 변환은 업 캐스팅(UpCasting)과 다운 캐스팅(DownCasting) 두 가지로 구분됩니다.

```
class Child extends Parent{  
    . .  
}
```

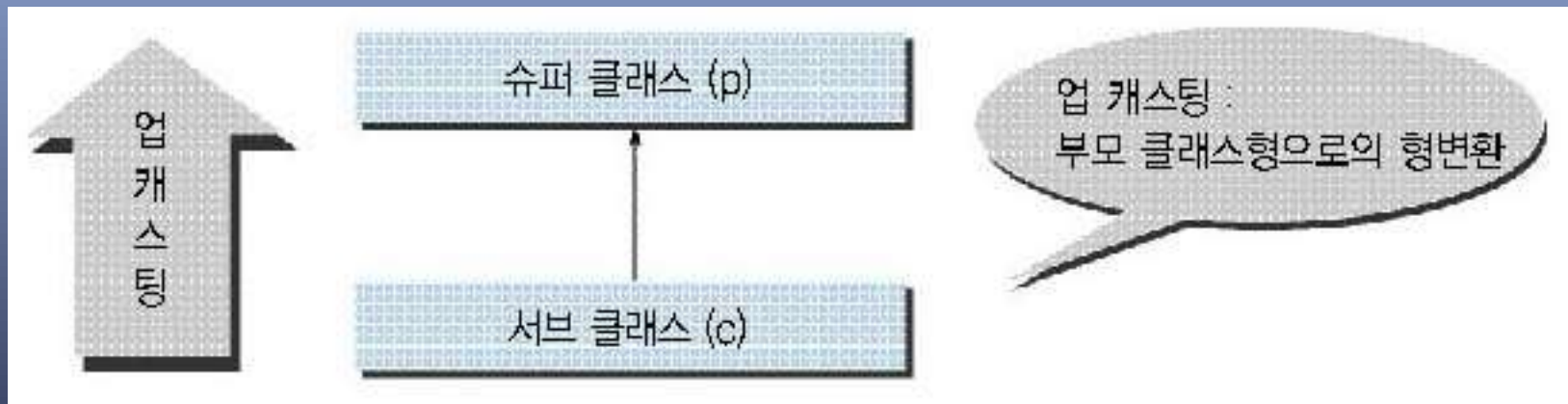
# 1.1 업 캐스팅

```
Child c = new Child(); //서브 클래스로 객체
```

```
Parent p; //슈퍼 클래스 형 레퍼런스 변수 선언
```

```
//슈퍼 클래스형 레퍼런스 변수에 서브 클래스의 레퍼런스 값이 대입됨
```

```
p = c; //이를 업 캐스팅이라하며 컴파일러에 의해서 자동적으로 형 변환됨
```



# 1.1 업 캐스팅

서브(하위) 클래스의 레퍼런스 값이 슈퍼(상위) 클래스의 레퍼런스 변수에 저장되면서 슈퍼(상위) 클래스 형으로 변환이 일어나는 데 이를 업 캐스팅(UpCasting)이라고 합니다.

업 캐스팅은 서브 클래스가 슈퍼 클래스로 형 변환이 일어나는 것으로서 이러한 형 변환(업 캐스팅)은 자바 컴파일러에 의해서 암시적으로 형 변환이 일어납니다.

# <예제> 업 캐스팅의 예

```
001:class Parent{
002:  public void parentPrn( ){
003:      System.out.println("슈퍼 클래스:ParentPrn 메서드");
004:  }
005:}
006://Parent를 슈퍼 클래스로 하는 서브 클래스 정의
007:class Child extends Parent{
008:  public void childPrn( ){
009:      System.out.println("서브 클래스 : ChildPrn 메서드");
010:  }
011:}
012:
```

# <예제> 업 캐스팅의 예

```
013: class RefTest01{
014: public static void main(String[] args){
015:     Child c = new Child(); //서브 클래스로 객체를 생성
016:     //Child 객체로 접근해서 호출할 수 있는 메서드는 2개
017:     c.parentPrn(); //부모로부터 상속 받은 메서드
018:     c.childPrn(); //자신의 메서드
019:     Parent p; //슈퍼 클래스형 레퍼런스 변수 선언
020:     p=c; //암시적으로 업 캐스팅이 일어남
021:     p.parentPrn(); //업 캐스팅 후에는 부모로부터 상속받은 메서드만 호출할 수 있다,
022:     //p.childPrn(); //컴파일 에러가 발생하게 된다.
023: }
024: }
```



# 1.1 업 캐스팅

업 캐스팅이란?

- 슈퍼 클래스로 형 변환하는 것이다.
- 참조 가능한 영역이 축소된다.
- 컴파일러에 의해서 자동 형 변환된다.

## 1.2 다운 캐스팅

```
001: class RefTest02{  
002:   public static void main(String[] args){  
003:     Parent p = new Parent(); //슈퍼 클래스로 인스턴스 선언  
004:     Child c;                //서브 클래스로 레퍼런스 변수 선언  
005:     //서브 클래스형 레퍼런스 변수에 슈퍼 클래스의 레퍼런스 값을 대입하면  
006:     c = p; //이를 DownCasting 이라하는데 컴파일러 에러가 발생한다.  
007:   }  
008: }
```



## 1.2 다운 캐스팅

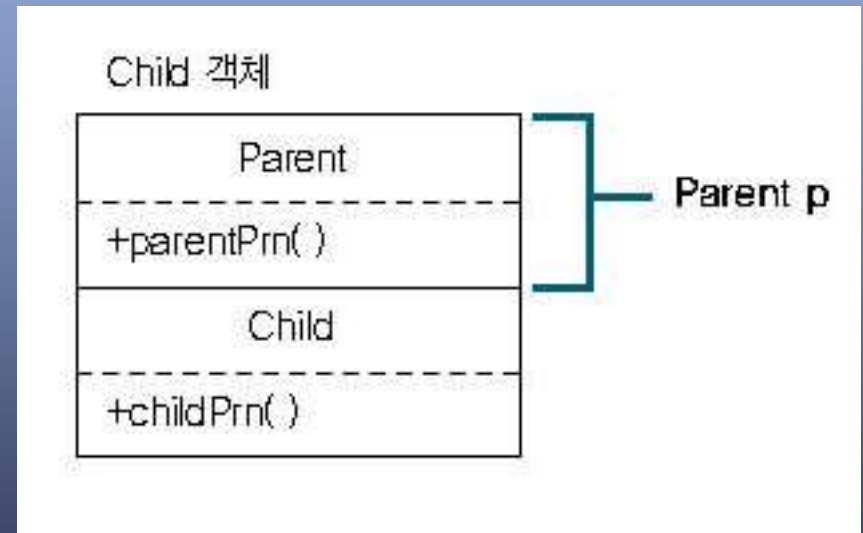
다운 캐스팅이란?

- 서브 클래스로 형 변환하는 것이다.
- 참조 가능한 영역이 확대되는 것을 의미한다.
- 컴파일러에 의해서 자동으로 형 변환되지 않는다.  
(왜냐하면 존재하지 않는 영역을 참조할 위험이 있기 때문이다.)

# 1.3 다운 캐스팅을 위한 강제 캐스팅

Parent p = new Child( ); //-----① ▪

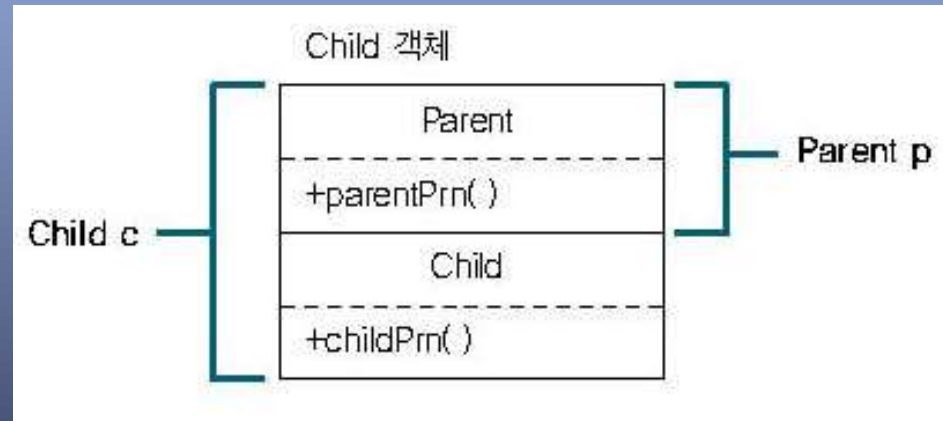
- ① 레퍼런스 p가 Child 클래스의 인스턴스를 가리키고 있습니다. 이 과정에서 서브 클래스를 슈퍼클래스로 선언된 레퍼런스가 가리키므로 업 캐스팅이 됩니다. 업 캐스팅은 명시적으로 기술하지 않아도 암시적으로 자바 컴파일러에 의해서 자동으로 형 변환이 일어납니다.



# 1.3 다운 캐스팅을 위한 강제 캐스팅

```
Parent p = new Child( ); //-----① ▪  
Child c = (Child)p;      //-----② ▪
```

② 에서 처럼 p 앞에 (Child)와 같이 캐스팅 연산자를 명시적으로 기술하여 강제 형 변환을 해야만 컴파일 에러가 발생하지 않습니다. 이렇게 강제로 캐스팅을 해주어야만 다운 캐스팅이 가능하게 됩니다.

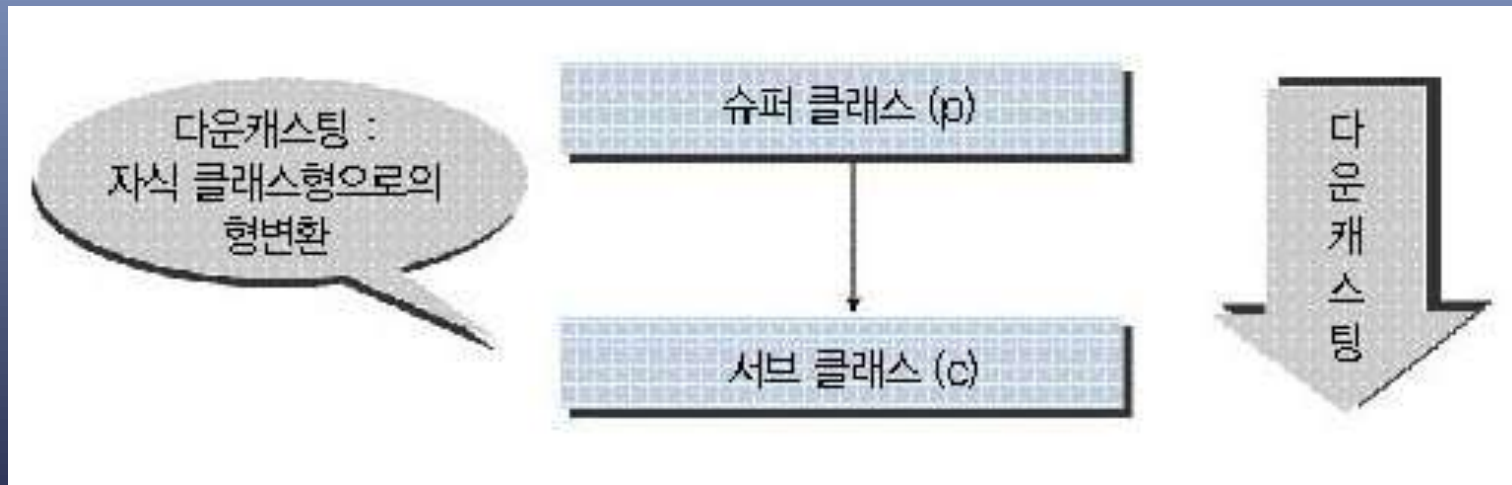


# 1.3 다운 캐스팅을 위한 강제 캐스팅

무사히 Child 레퍼런스 변수 *c*가 Child 객체를 가리키게 되면 두 메서드 (parentPrn, chilPrn) 모두를 사용할 수 있게 된다. ▪

`c.parentPrn();` //다운 캐스팅 후에는 부모로부터 상속받은 메서드는 물론

`c.childPrn();` //자신의 메서드도 호출가능하게 된다.



# <예제> 다운 캐스팅의 예

```
001: class RefTest03{
002:   public static void main(String[] args){
003:     Parent p = new Child( ); //서브 클래스로 인스턴스 선언
004:     p.parentPrn();
005:     //p.childPrn();//-컴파일 에러
006:     Child c;                //서브 클래스로 레퍼런스 변수 선언
007:     System.out.println("----->>");
008:     //서브 클래스 레퍼런스 변수에 슈퍼 클래스의 레퍼런스 값이 대입됨
009:     c = (Child) p; //강제 형변환으로 다운 캐스팅
010:     c.parentPrn();
011:     c.childPrn();
012: }
013: }
```

# <예제> 다운 캐스팅의 잘못된 예

```
001: class RefTest04{  
002: public static void main(String[] args){  
003:     Parent p = new Parent(); //슈퍼 클래스로 인스턴스 선언  
004:     Child c;                //서브 클래스로 레퍼런스 변수 선언  
005:     //서브 클래스 레퍼런스 변수에 슈퍼 클래스의 레퍼런스 값이 대입됨  
006:     c = (Child) p; //강제 형변환으로 다운 캐스팅  
007:     c.parentPrn();  
008:     c.childPrn();  
009: }  
010: }
```



# 1.3 다운 캐스팅을 위한 강제 캐스팅

다운 캐스팅이란?

- 서브 클래스로 형 변환하는 것이다.
- 참조 가능한 영역이 확대되는 것을 의미한다.
- 컴파일러에 의해서 자동으로 형 변환되지 않는다.
- 프로그래머에 의해서 명시적으로 캐스팅을 해 주어야만 한다.  
(컴파일상의 에러 발생을 막음)
- 캐스팅 후에도 실행 시 예외사항이 발생할 수 있으므로 인스턴스의 클래스 형과 참조하는 레퍼런스 변수의 상속 관계를 생각해서 명시적 캐스팅을 해야 한다.  
(한번 업 캐스팅이 된 레퍼런스 값을 다운 캐스팅하는 경우에만 안전하다.)

### 3. 업캐스팅과 메서드 오버라이딩

```
class Parent{  
    public void parentPrn( ){  
    }  
}
```

```
class Child extends Parent{  
    public void parentPrn( ){  
    }  
}
```

//서브 클래스로 객체를 생성하여 슈퍼 클래스형 레퍼런스 변수가 가리킴  
Parent p = new Child(); //암시적으로 업캐스팅이 됨

```
p.parentPrn();
```

# <문제>

1. 업 캐스팅과 다운 캐스팅에 대해서 설명하시오.

2. 다음 수행 결과값을 유추하시오.

```
class SubClass {  
    int i = 3;  
}  
  
public class Ex17_01{  
    public static void main(String args[]) {  
        Object o = new SubClass();  
        SubClass foo = (SubClass)o;  
        System.out.println("i = " + foo.i);  
    }  
}
```

No.19

## <문제>

3. 다음 프로그램의 결과를 유추하시오.

```
class Super {  
    public int getNumber(int a) { return a + 1; }  
}  
class CastingEx03 extends Super {  
    public int getNumber (int a) { return a + 2; }  
}  
public class Ex17_02{  
    public static void main (String args[]) {  
        Super a = new CastingEx03 ();  
        System.out.println(a.getNumber(0));  
    }  
}
```