실습: super()

파일명:taxiTest.java

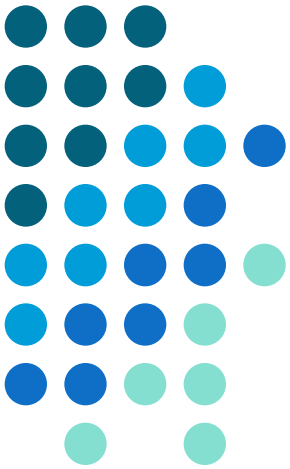
```
1  class Car {
2      String model;
3      String color;
4
5  }
6
7  class Taxi extends Car{
8      Boolean TaxiMeter;
9      int TaxiFare=2000;
10
11
12 }
13 public class texiTest {
14     public static void main(String[] args) {
15     }
16 }
```

```
1  class Car {
2      String model;
3      String color;
4      public Car(String model, String color) {
5          super();
6          this.model = model;
7          this.color = color;
8      }
9      public Car() {
10         super();
11     }
12 }
13 class Taxi extends Car{
14     Boolean TaxiMeter;
15     int TaxiFare=2000;
16     public Taxi(String model, String color, Boolean taxiMeter, int t
17         super(model, color);
18         TaxiMeter = taxiMeter;
19         TaxiFare = taxiFare;
20     }
21     public Taxi(Boolean taxiMeter, int taxiFare) {
22         super();
23         TaxiMeter = taxiMeter;
24         TaxiFare = taxiFare;
25     }
26     public Taxi() {
27         super();
28     }
29 }
```

실습: super()-계속

```
 1 class Car {
 2     String model;
 3     String color;
 4     public Car(String model, String color) {
 9     public Car() {
12
13     public void show() {
14         System.out.println("모델명:"+model+",칼라:"+color);
15     }
16 }
17 class Taxi extends Car{
18     Boolean TaxiMeter;
19     int TaxiFare=2000;
20     public Taxi(String model, String color, Boolean taxiMeter, int t
25     public Taxi(Boolean taxiMeter, int taxiFare) {
30     public Taxi() {
33
34     public void show() {
35         super.show();
36         System.out.println("택시메타기:"+TaxiMeter+",칼라:"+TaxiFare);
37     }
38 }
39 public class texiTest {
40     public static void main(String[] args) {
41         Taxi t=new Taxi();
42         t.show();
43         Taxi t2=new Taxi(true, 3000);
44         t2.show();
45         Car c2=new Car("모닝","흰색");
46         c2.show();
47     }
48 }
```

11. 레퍼런스 형 변환



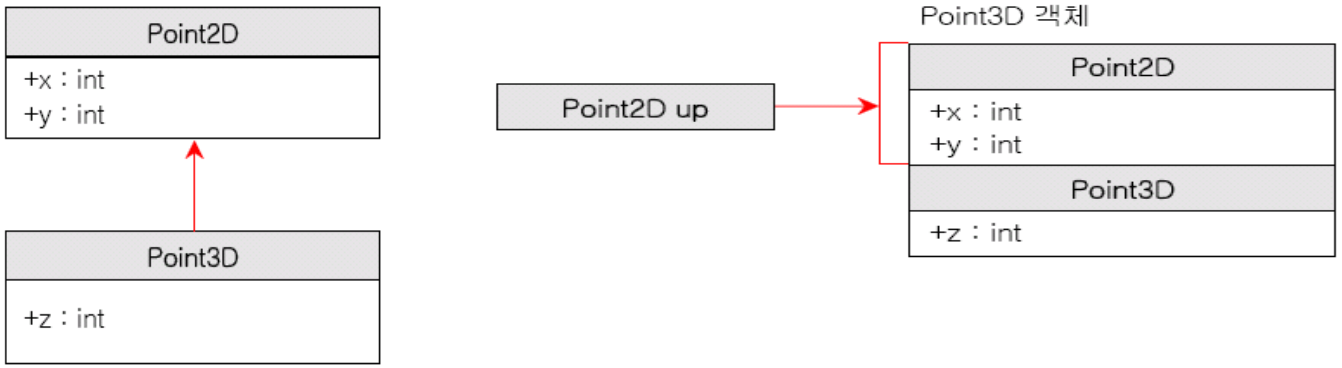
목차

01. 클래스에서의 형 변환

```
1  class Point2D {
2      public int x;
3      public int y;
4      public void show2D() {
5          System.out.println("2D쇼");
6      }
7  }
8  class Point3D extends Point2D {
9      public int z;
10     public void show3D() {
11         System.out.println("3D쇼");
12     }
13 }
14 public class Ch08Ex01 {
15     public static void main(String[] args) {
16         Point3D pt=new Point3D();    //자식클래스
17         Point2D up=pt;               //부모클래스의 객체에 자식클래스의 객체 할당
18         System.out.println(pt.x);
19         System.out.println(pt.y);
20         System.out.println(pt.z);
21         System.out.println(up.x);
22         System.out.println(up.y);
23         System.out.println(up.z); //컴파일 에러
24         up.show2D();
25         up.show3D(); //컴파일 에러
26     }
27 }
```

01. 클래스에서의 형 변환

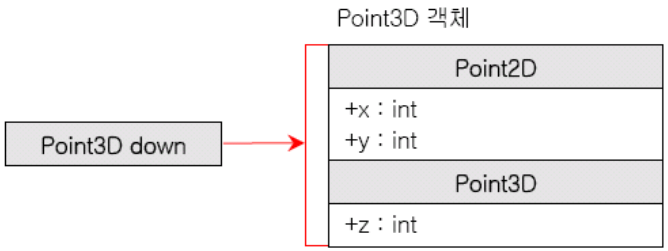
- 클래스 사이에서 객체의 형변환(1)
  - 업케스트
    - 부모 클래스형의 변수 = 자식 클래스의 객체(o)
    - 부모 클래스형의 변수를 통해서 부모 클래스에 선언된 속성에만 접근 가능
  - Point2D **up** = new Point3D( );



다운 캐스팅(명시적인 형 변환)

- 클래스 사이에서 객체의 형변환(2)
  - 다운케스트
    - 자식 클래스형의 변수 = 부모 클래스의 객체(x)
  - Point3D down= new Point2D( ); //컴파일 에러
  - Point3D down= (Point3D) new Point2D(); //실행 에러
  - 만약 업케스트후 다운케스트인 경우

```
Point2D  up=new Point3D();
Point3D  down=(Point3D) up; //명시적 형변환
```

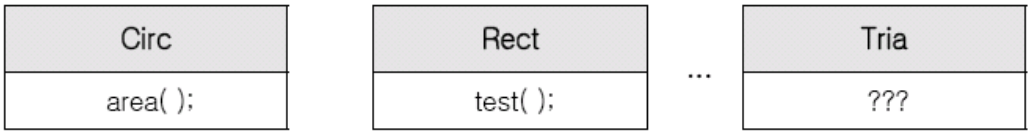


다운 캐스팅(명시적인 형 변환)

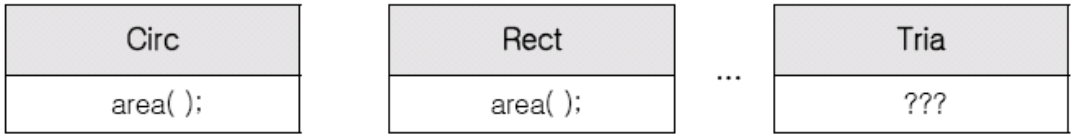
```
1  class Point2D {
2      public int x;
3      public int y;
4  }
5  class Point3D extends Point2D {
6      public int z;
7  }
8  public class Ch08Ex01 {
9      public static void main(String[] args) {
10         Point2D up=new Point3D();
11         System.out.println(up.x);
12         System.out.println(up.y);
13     //  System.out.println(up.z);
14         Point3D down=(Point3D) up;
15         System.out.println(down.x);
16         System.out.println(down.y);
17         System.out.println(down.z);
18     }
19 }
```

02. 다형성과 메소드 오버라이딩

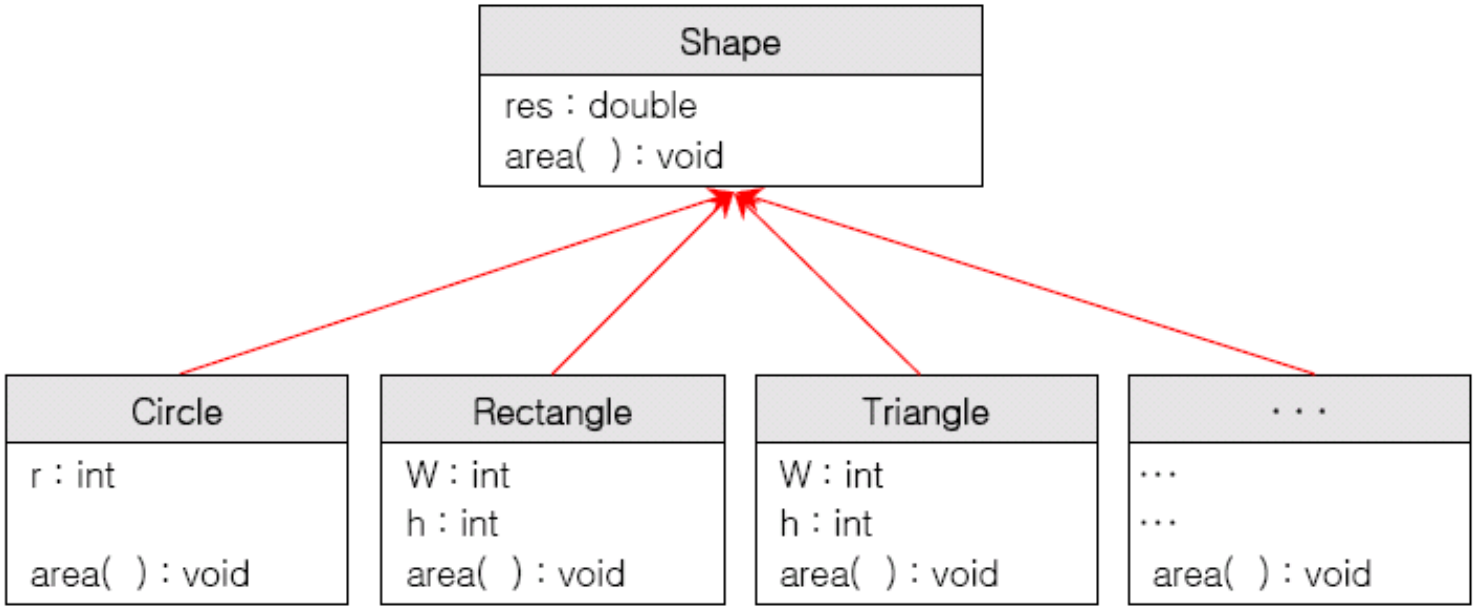
- 도형과 관련된 3개의 클래스
  - 원의 면적은 area()메소드
  - 사각형의 면적은 test()메소드
  - 삼각형의 면적은 extend()메소드



- 사용자의 편리를 위해
  - 원,사각형, 삼각형의 면적은 area()메소드로 통일



02. 다형성과 메소드 오버라이딩



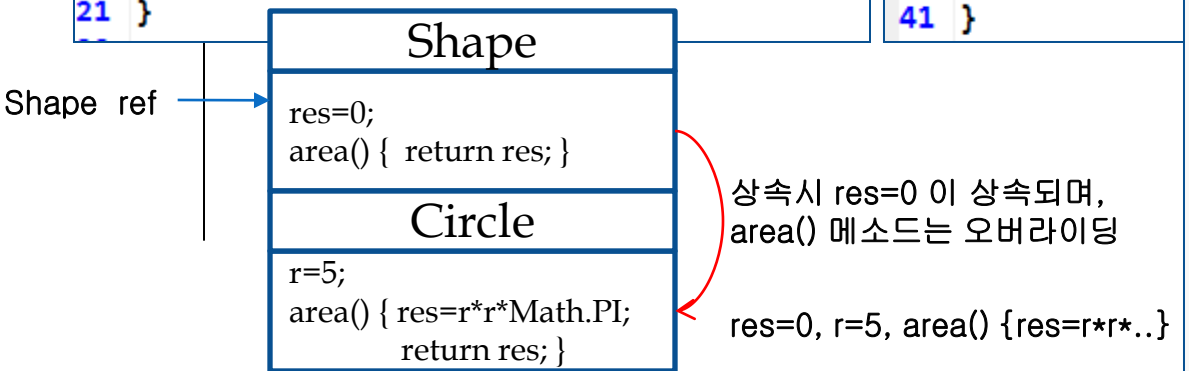
02. 다형성과 메소드 오버라이딩

```java
class Shape{
    public double res=0;
    public double area() {
        return res;
    }
}
class Circle extends Shape {
    public int r=5;
    public double area() {
        res=r*r*Math.PI;
        return res;
    }
}
class Rectangle extends Shape {
    public int w=10;
    public int h=10;
    public double area() {
        res=w*h;
        return res;
    }
}
```

```java
class Triangle extends Shape {
    public int w=10;
    public int h=10;
    public double area() {
        res=w*h*0.5;
        return res;
    }
}
public class Ch08Ex02 {
    public static void main(String[] args) {
        Shape ref=null;
        ref=new Circle();
        System.out.println(ref.area());
        ref=new Rectangle();
        System.out.println(ref.area());
        ref=new Triangle();
        System.out.println(ref.area());
    }
}
```

비교

```java
public class Ch08Ex02 {
    public static void main(String[] args) {
        Circle cref=new Circle();
        System.out.println(cref.area());
        Rectangle rref=new Rectangle();
        System.out.println(rref.area());
        Triangle tref=new Triangle();
        System.out.println(tref.area());
    }
}
```



is a 상속이란?

- 상속 관계는 is a 관계
  - Shape ref= new Rectangle(); //(상속 관계에 있을 때
    - 사각형(Rectangle) is a 도형(Shape)이다.
  - Circle c=ref; //컴파일 에러
    - 도형은 원이다.
  - Circle c=(Circle) ref; //명시적 변환해도 실행시 에러

  - 올바른 상속 관계(is a)

```
Shape ref=new Rectangle();
Rectangle r=(Rectangle) ref;
```

03. instanceof 연산자

- instanceof 연산자
  - 컴파일 또는 실행시에 에러를 방지하기 위해 형변환 전에 형변환 가능한지 체크하는 연산자
  - 레퍼런스 변수가 어떤 인스턴스를 참조하고 있는지를 검사하는 연산자

```
Shape ref = new Rectangle( );      // ····· ①
if(ref instanceof Circle)          // ····· ②
    Circle c = (Circle)ref;        // ····· ③
else if (ref instanceof  Rectangle) // ····· ④
Rectangle r=(Rectangle)ref;        // ····· ⑤
```

실습: instanceof 연산자

```
1  public class Ch08Ex03 {
2      public static void polyMethod(Shape ref){
3          ref.area();
4
5          if ( ref instanceof Circle) {
6            Circle c=(Circle)ref;
7            System.out.println("반지름이 "  + c.r + " 인 원의 면적은 " + c.res);
8          }
9          else if ( ref instanceof Rectangle) {
10           Rectangle r=(Rectangle)ref;
11           System.out.println("너비가 "+ r.w+" 이고, 높이가 "+ r.h+" 인 사각형의 면적은 " + r.res);
12         }
13         else if ( ref instanceof Triangle) {
14           Triangle t=(Triangle)ref;
15               System.out.println("너비가 "+ t.w+ " 이고, 높이가 "+ t.h+ " 인 삼각형의 면적은 " + t.res);
16         }
17          }
18       public static void main(String[] args) {
19
20         Shape c=new Circle();
21         Shape r=new Rectangle();
22         Shape t=new Triangle();
23
24         polyMethod(c);  polyMethod(r);  polyMethod(t);
25           }
26     }
```

04. final 예약어

- final 변수
  - 변수 앞 final은 한번 값 정하면 바꿀 수 없다
- final 메소드
  - 메소드 앞 final은 더 이상 상속이 불가능한 메소드
- final 클래스
  - 클래스 앞 final은 더 이상 상속이 불가능한 클래스

실습: final 예약어

```
1  class Point2D {
2      int x=10;
3      int y=20;
4      final public void show() {
5          System.out.println(x+","+y);;
6      }
7  }
8
9  final class Point3D extends Point2D {
10     int z=30;
11     public void show() {  //show()메소드는 오버라이딩이 불가능
12     }
13 }
14
15 class Point4D extends Point3D {  //Point3D는 final 클래스이므로 수퍼클래스로 사용x
16
17 }
18 public class Ch08Ex04 {
19     public static void main(String[] args) {
20     }
21 }
```