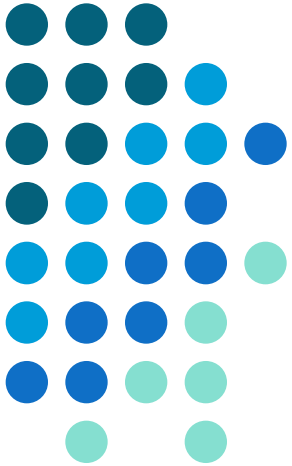


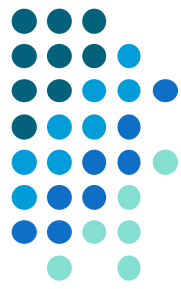
12. 추상클래스와 인터페이스





목차

- 01. 추상 클래스
- 02. 추상 메소드
- 03. 추상 클래스로 다형성 제공
- 04. 인터페이스
- 05. 다중 상속을 위한 인터페이스
- 06. 인터페이스의 상속



01. 추상 클래스

- 추상 클래스
 - 인스턴스를 생성할 수 없는 미완성 클래스
 - 인스턴스를 생성한다는 뜻은 (예, `new Circle();`)
- 실습 : 인스턴스 생성이 불가능한 추상 클래스

```
1 package kame.ch08.ex01;  
2  
3 public abstract class Shape {  
4  
5 }
```

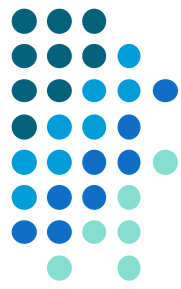
```
1 package kame.ch08.ex01;  
2  
3 public class Ch08Ex05 {  
4     public static void main(String[] args) {  
5         Shape obj=new Shape();  
6  
7     }  
8 }
```

추상클래스는 인스턴스를 생성할 수 없다. 3



02. 추상 메소드

- 추상 클래스
 - 인스턴스를 생성하지 못하는 클래스(new로 객체 생성할 수 없다.
 - 즉, 상속해서 자식클래스에서 인스턴스 생성
 - 추상 클래스 내부에 추상 메소드와 일반 메소드 모두 가질 수 있다.
- 추상 메소드
 - 내용 없이 선언만 한다.
 - 자식 클래스에서 메소드를 오버라이딩하여 내용을 갖는다



실습 : 추상 메소드를 갖는 추상 클래스 설계하기

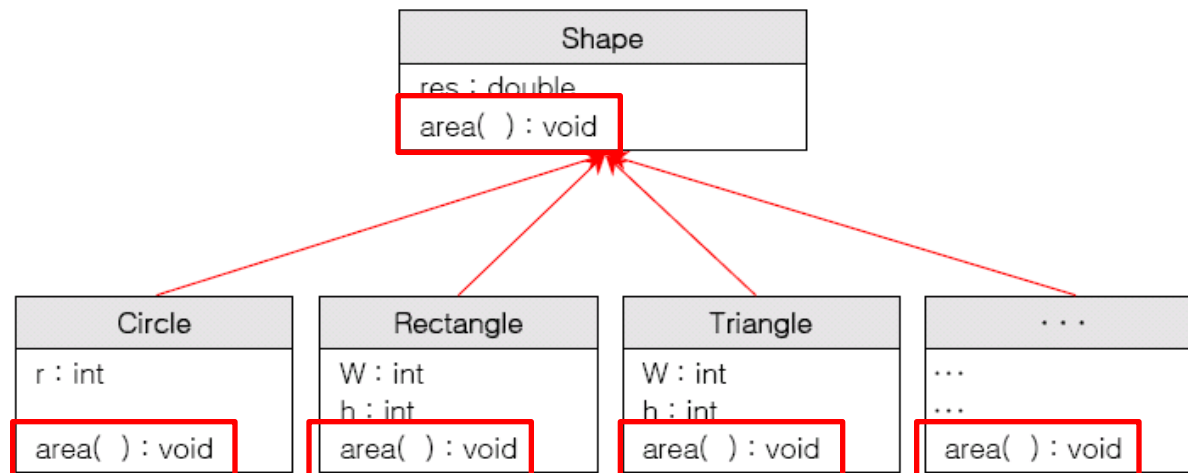
```
1 package kame.ch08.ex01;
2
3 public abstract class Shape {
4     public abstract void draw();           //추상 메소드
5     public void show() {                  //일반 메소드
6         System.out.println("그린다");
7     }
8 }
```

```
1 package kame.ch08.ex01;
2
3 public class Circle extends Shape {
4     public void draw() {                  //반드시 정의해야 한다.
5         System.out.println("원을 그린다");
6     }
7     public void show() {                  //재정의 해도 되고, 안해도 된다.
8         System.out.println("또 그린다");
9     }
10 }
```

```
1 package kame.ch08.ex01;
2
3 public class Ch08Ex05 {
4     public static void main(String[] args) {
5         Shape up;                        //추상 클래스로 레퍼런스 변수 선언은 가능
6         up=new Circle();                  //서브클래스인 Circle 인스턴스 생성된다.
7         up.draw();                        //업캐스팅후, Circle 클래스의 draw 메소드가 호출
8         up.show();
9     }
10 }
```

03. 추상 클래스로 다형성 제공

- 추상 클래스로 다형성 제공
 - 클래스들 사이의 공통적인 속성과 메소드를 부모 클래스에 선언
 - 자식클래스에 동일한 이름의 메소드를 강제적으로 정의





추상 클래스의 상속을 받는 서브 클래스

```
1 package kame.ch08.ex01;
2
3 public abstract class Shape {
4     public double res=0;
5     public abstract double area();
6     public void printArea(){
7         System.out.println("면적은"+res+"이다.");
8     }
9 }
```

```
1 package kame.ch08.ex01;
2 public class Circle extends Shape {
3     public int r=5;
4     public double area() {
5         res=r*r*Math.PI;
6         return res;
7     }
8 }
```

```
1 package kame.ch08.ex01;
2
3 public class Rectangle extends Shape {
4     public int w=10;
5     public int h=10;
6     public double area() {
7         res=w*h;
8         return res;
9     }
10
11 }
```

```
1 package kame.ch08.ex01;
2
3 public class Triangle extends Shape {
4     public int w=10;
5     public int h=10;
6     public double area() {
7         res=w*h*0.5;
8         return res;
9     }
10
11 }
```



추상 클래스의 상속을 받는 서브 클래스

- Shape s=new Shape(); (X)

```
1 package kame.ch08.ex01;  
2 public class Ch08Ex06 {  
3     public static void main(String[] args) {  
4         Shape ref=null;  
5         ref=new Circle();  
6         ref.area();  
7         ref.printArea();  
8  
9         ref=new Rectangle();  
10        ref.area();  
11        ref.printArea();  
12  
13        ref=new Triangle();  
14        ref.area();  
15        ref.printArea();  
16    }  
17 }
```




추상 클래스 정리

- 추상 클래스
 - 추상 메소드와 일반 메소드를 모두 가진다
 - 상속을 위해 extends를 사용한다
 - 단일 상속만 가능하다(오직 하나의 클래스에 대해서만 extends 해야 함)
 - 추상 클래스 내의 모든 추상 메소드는 자식 클래스에서 반드시 오버라이딩하여 사용
 - 업캐스팅이 가능하다
 - (추상클래스를 부모로 하고, 자식클래스가 부모클래스(추상클래스)에 업캐스팅한다



04. 인터페이스

- interface
 - 자바에서는 다중 상속을 가능하게 하기 위해 인터페이스를 제공
 - 클래스와 별개의 개념으로 메소드로 접근하기 위한 인터페이스를 제공하는 용도
 - 인터페이스 내의 메소드는 public abstract 생략된 것으로 인식(즉, 추상 메소드로 인식)
- 인터페이스의 메소드
 - abstract 메소드이므로 몸체만 있음
 - 접근지정자는 public으로 한다
- 인터페이스 내의 변수
 - static final이 생략된 것으로 인식
 - 인터페이스 내의 모든 변수는 상수로 인식



인터페이스의 메소드

- 클래스에서 인터페이스를 사용
 - implements 예약어를 사용하여 상속

```
1 package kame.ch08.ex02;  
2  
3 public interface Drawable {  
4     void draw(); //public abstract void draw();사용가능  
5 }
```

```
1 package kame.ch08.ex02;  
2  
3 public class Circle implements Drawable {  
4     public void draw() {  
5         System.out.println("원을 그린다");  
6     }  
7 }
```

```
1 package kame.ch08.ex02;  
2  
3 public class Ch08Ex07 {  
4     public static void main(String[] args) {  
5         Drawable ref;  
6         ref=new Circle();  
7         ref.draw();  
8     }  
9 }
```



인터페이스의 속성

- 인터페이스의 속성
 - 변수만 정의할 수 없고, 반드시 값을 할당하여 상수로 인식한다.
(반드시 초기화해야 한다.)
 - 모든 속성은 `public static final` 속성을 갖는다.
(생략가능)

실습:인터페이스 속성

```
1 package kame.ch08.ex03;
2 public interface Color {
3     int RED=1;        //public static final이 생략된 것이므로
4     int GREEN=2;      //상수로 인식
5     int BLUE=3;
6     void setColor(int c); //public abstract이 생략된 것이므로
7     int getColor();      //추상메소드로 인식
8 }
9
```

```
1 package kame.ch08.ex03;
2
3 public class Triangle implements Color {
4
5     public void setColor(int c) {
6
7     }
8     public int getColor() {
9         return 0;
10    }
11
12 }
```

```
1 package kame.ch08.ex03;
2
3 public class Triangle implements Color {
4     int color=0;
5     public void setColor(int c) {
6         color=c;
7     }
8     public int getColor() {
9         return color;
10    }
11 }
```

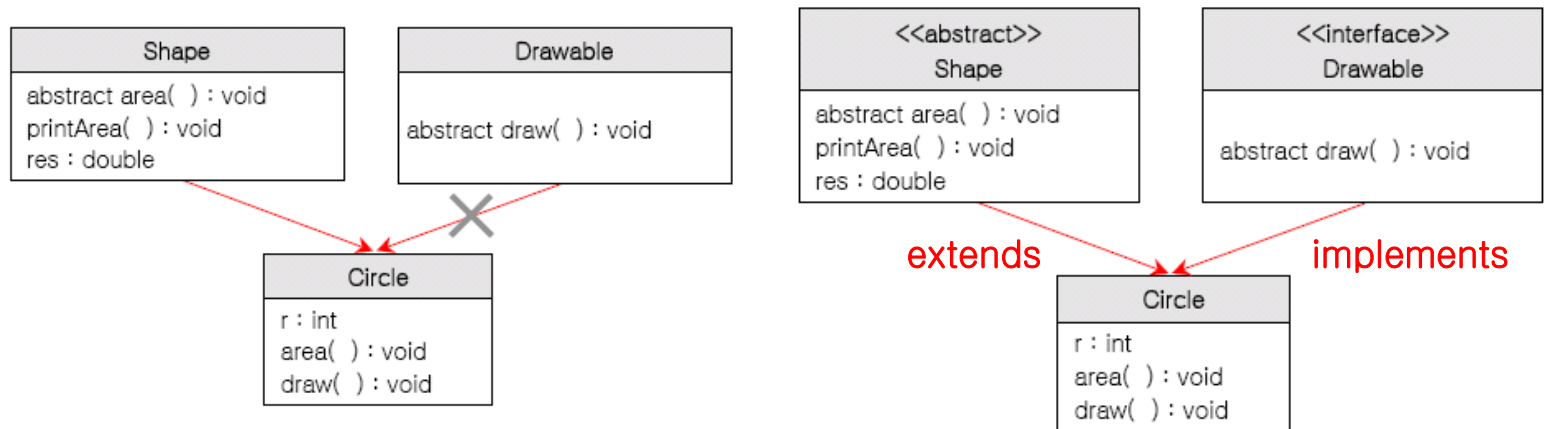


실습:인터페이스 속성(계속)

```
1 package kame.ch08.ex03;
2
3 public class Ch08Ex08 {
4     public static void main(String[] args){
5         Triangle ref=new Triangle();
6         ref.setColor(Color.RED);
7         switch(ref.getColor()){
8             case 1:
9                 System.out.print("빨강");
10                break;
11             case 2:
12                 System.out.print("초록");
13                break;
14             case 3:
15                 System.out.print("파랑");
16                break;
17         }
18         System.out.println("삼각형이다.");
19     }
20 }
```

05. 다중 상속을 위한 인터페이스

- 자바의 클래스
 - 단일 상속만 가능
- 인터페이스 개념으로 하위 클래스를 설계



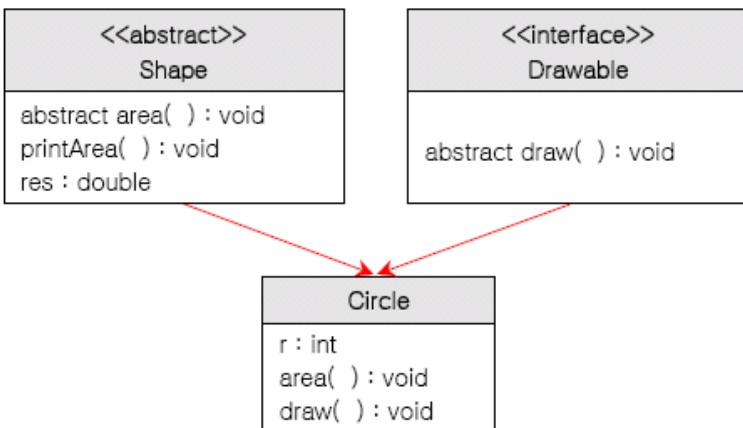
```
public class Circle extends Shape, Drawable { //(X)  
}
```

```
public class Circle extends Shape implements Drawable, ... { //(O)  
}
```

실습:인터페이스로 다중 상속

```
1 package kame.ch08.ex04;
2
3 public abstract class Shape {
4     public double res=0;
5     public abstract double area();
6     public void printArea(){
7         System.out.println("면적은"+res+"이다");
8     }
9 }
```

```
1 package kame.ch08.ex04;
2 import kame.ch08.ex02.Drawable;
3 public class Rectangle extends Shape implements Drawable {
4     public int w=10;
5     public int h=10;
6     public void draw() {
7         System.out.println("사각형그리다");
8     }
9
10    public double area() {
11        res=w*h;
12        return res;
13    }
14 }
```

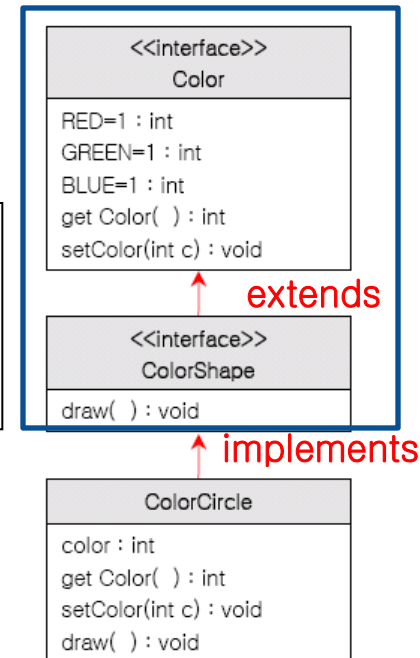


```
1 package kame.ch08.ex04;
2 public class Ch08Ex09 {
3     public static void main(String[] args) {
4         Rectangle ref=null;
5         ref=new Rectangle();
6         ref.area();
7         ref.printArea();
8         ref.draw();
9     }
10 }
```


06. 인터페이스의 상속

- 인터페이스의 상속
 - 인터페이스 선언시 필요에 따라 다른 인터페이스로부터 상속받을 수 있다.
 - extends 예약어 사용

```
interface 인터페이스이름 extends 인터페이스이름, 인터페이스 이름 {  
    상수선언  
    추상메소드 선언  
}
```



실습:인터페이스 상속

```
1 package kame.ch08.ex03;
2 public interface Color {
3     int RED=1;      //public static final이 생략된 것이므로
4     int GREEN=2;    //상수로 인식
5     int BLUE=3;
6     void setColor(int c); //public abstract이 생략된 것이므로
7     int getColor();      //추상메소드로 인식
8 }
9
```

```
1 package kame.ch08.ex04;
2
3 import kame.ch08.ex03.Color;
4
5 public interface ColorShape extends Color {
6     void draw();
7 }
```

```
1 package kame.ch08.ex04;
2
3 public class Ch08Ex10 {
4     public static void main(String[] args) {
5         ColorCircle ref=new ColorCircle();
6         ref.setColor(2);
7         ref.draw();
8     }
9 }
```

```
1 package kame.ch08.ex04;
2
3 public class ColorCircle implements ColorShape {
4     int color=0;
5     public void setColor(int c) {
6         color=c;
7     }
8     public int getColor() {
9         return color;
10    }
11    public void draw() {
12        switch(color){
13            case 1:
14                System.out.print("빨강");
15                break;
16            case 2:
17                System.out.print("초록");
18                break;
19            case 3:
20                System.out.print("파랑");
21                break;
22        }
23        System.out.println("원이다.");
24    }
25 }
26
```