

JAVA프로그래밍

과제1

2019305059

이현수

5.8

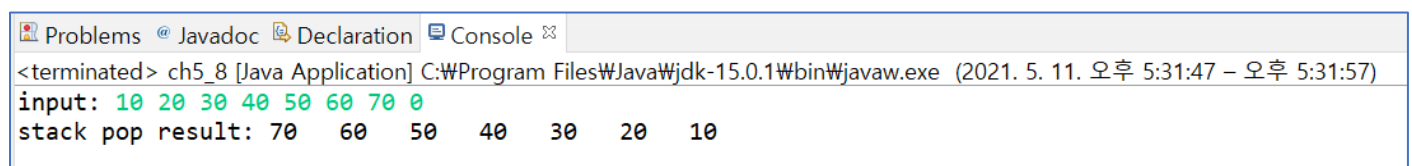
<문제>

- (1) 디폴트 생성자를 작성하시오. 이 경우에 스택의 크기는 100이다.
- (2) 메소드 push()와 pop()을 작성하시오.
- (3) 클래스 Stack을 이용하여 일련의 정수 입력(입력의 끝은 0)을 역순으로 출력하는 자바 프로그램을 작성하시오.

<코드>

```
1 import java.util.Scanner;
2
3 class Stack{
4     private int stack[];
5     int sp=0;
6     Stack(){
7         stack = new int[100];
8     }
9     Stack(int size){
10        stack = new int[size];
11    }
12    public void push(int data) {
13        stack[sp++]=data;
14    }
15    public int pop() {
16        return stack[--sp];
17    }
18 }
19
20 public class ch5_8 {
21     public static void main(String[] args) {
22         Scanner sc = new Scanner(System.in);
23         Stack s=new Stack();
24         int cnt=0;
25
26         System.out.print("input: ");
27         while(true) {
28             int num = sc.nextInt();
29             if(num==0) break;
30             s.push(num);
31             cnt++;
32         }
33
34         System.out.print("stack pop result: ");
35         for(int i=0;i<cnt;i++) {
36             System.out.printf("%-4d ", s.pop());
37         }
38
39         sc.close();
40     }
41 }
```

<실행결과>



```
<terminated> ch5_8 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 5:31:47 - 오후 5:31:57)
input: 10 20 30 40 50 60 70 0
stack pop result: 70 60 50 40 30 20 10
```

<설명>

Stack 클래스에 int형 배열을 만들기위해 stack[]을 만든다. 스택포인터 역할을 하는 int형 변수 sp=0으로 초기화 한다. Stack 디폴트 생성자를 만들어 스택크기를 100으로 만든다. push메소드는 매겨변수 data를 배열에 넣고 후 위 연산자를 통해 sp를 1증가 시킨다. pop메소드는 전위연산자를 통해 sp를 먼저 1 줄이고 스택에 있는 정수를 반환한다. main메소드에서 정수를 입력받기 위해서 Scanner객체 sc를 선언하고 Stack객체 s를 만든다. while반복문을 통해 사용자로부터 0을 받을 때까지 숫자를 받은 후 s.push(num);을 통해 스택s에 사용자가 입력한 정수를 저장한다. 사용자가 몇 개의 데이터를 입력했는지 알기위해 cnt변수를 0으로 초기화하고 push할때마다 반복문 내에서 1씩 증가시키고, 이 cnt수를 기반으로 for반복문을 돌려 s.pop()을 통해 정수를 나열한다.

5.9.(1)

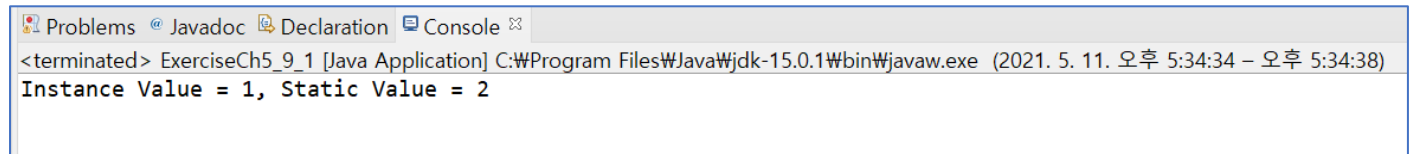
<문제>

다음 프로그램의 실행 결과는 무엇인가?

<코드>

```
1 class Count{
2     public static int scount=0;
3     public int count=0;
4     public static void sIncrement() {
5         scount++;
6     }
7     public void increment() {
8         count++;
9     }
10 }
11 public class ExerciseCh5_9_1 {
12     public static void main(String[] args) {
13         Count c = new Count();
14         c.increment(); Count.sIncrement(); c.sIncrement();
15         System.out.print("Instance Value = "+c.count);
16         System.out.println(", Static Value = " + Count.scount);
17     }
18 }
```

<실행결과>



Problems @ Javadoc Declaration Console

<terminated> ExerciseCh5_9_1 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 5:34:34 - 오후 5:34:38)

Instance Value = 1, Static Value = 2

<설명>

클래스 Count내에 static변수 scount=0으로 인스턴스 변수 count=0으로 초기화한다. 그리고 static 메소드 sIncrement에서 static 변수 scount를 1증가 시킨다. 참고로 static 메소드에서는 static 변수만 접근이 가능하다.

인스턴스 메소드 increment에서는 인스턴스 변수 count를 1증가 시킨다.

main메소드에서는 Count 객체 c를 만든다.

c.increment(); Count.sIncrement(); c.sIncrement();를 실행 후 count와 scount를 출력한다.

첫번째 c.increment();를 통해 인스턴스 변수 count가 1증가한다. (count=0 -> count=1)

다음으로 Count.sIncrement(); 를 통해 static 변수 scount를 1증가시킨다. (scount=0->scount=1) 참고로 static 메소드는 객체뿐만 아니라 클래스이름.static메소드 형태로 호출이 가능하다.

마지막 c.sIncrement();를 통해 static 변수 scount를 1증가 시킨다. (scount=1 -> scount=2) static 메소드는 위에서 본 클래스이름.static메소드 뿐만 아니라 객체이름.static메소드를 통해서도 호출이 가능하다.

그래서 결과적으로 인스턴스 변수는 1, static변수는 2라고 결과가 출력된다.

5.9.(2)

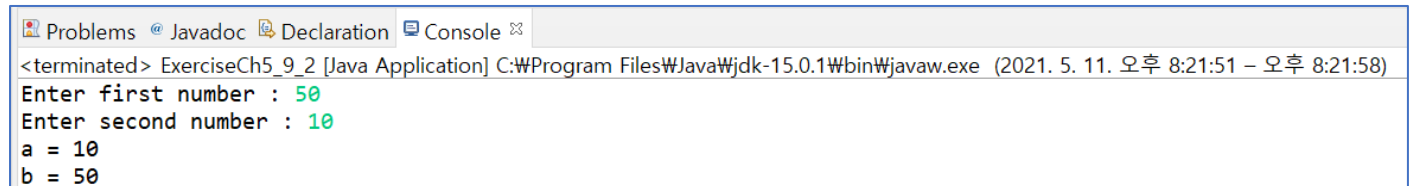
<문제>

다음 프로그램의 실행 결과는 무엇인가?

<코드>

```
1 import java.io.*;
2
3 public class ExerciseCh5_9_2 {
4     static int method(int x, int y) {
5         while(x!=y) {
6             if(x>y) x-=y;
7             else y-=x;
8         }
9         return x;
10    }
11    public static void main(String[] args) throws NumberFormatException, IOException {
12        BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
13        int i, j;
14        int a, b;
15
16        System.out.print("Enter first number : ");
17        i = Integer.parseInt(input.readLine());
18        System.out.print("Enter second number : ");
19        j = Integer.parseInt(input.readLine());
20        a = method(i, j);
21        System.out.println("a = " + a);
22        b = (i/a)*(j/a)*a;
23        System.out.println("b = " + b);
24    }
25 }
```

<실행결과>



```
<terminated> ExerciseCh5_9_2 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 8:21:51 - 오후 8:21:58)
Enter first number : 50
Enter second number : 10
a = 10
b = 50
```

<설명>

main메소드에서 변수 i, j에 사용자가 정수를 입력한다. 이때 입력할 때 BufferedReader로 입력하고 Integer.parseInt 메소드안에 input.readLine()을 통해 입력을 받는다. Input.readLine()은 문자열형태로 입력이 들어오지만 Integer.parseInt를 통해 문자열을 정수로 바꿔줄 수 있다. 그래서 i에 j에 각각 정수를 입력받는다.

그리고 method메소드의 반환값을 a에, $(i/a)*(j/a)*a$ 는 b에 저장하고 출력하는 프로그램이다.

이때 method메소드는 매개변수 x,y를 받고 x, y가 같아질때까지 더 큰 숫자에서 작은 숫자를 빼는 연산을 반복한다. 그리고 첫번째 매개변수, 즉 사용자가 처음에 입력받은 숫자를 반환한다. 여기서 x가 숫자가 바뀌더라도 메인 메소드에 있는 i, j는 전혀 영향이 없다. method메소드에 매개변수를 전달할 때 참조가 아닌 복사본을 전달했기 때문에 사용자가 입력한 i, j숫자는 입력한 숫자로 유지된다.

실제로 사용자가 입력한 대로 프로그램을 설명해보겠다. 사용자가 i에 50, j에 10을 입력했다. 여기서 method메소드를 통해 $x=50, y=10$ 이 되고 x가 더 크므로 $x-=y$ 즉 $50-=10$ 이되 $x=40$ 이것이 반복되 결국 x는 40, 30, 20, 10이 되다가 10이 되고 10이 반환되 a는 10이다. 다음 메인메소드에 와서 $b = (50/10)*(10/10)*10=5*1*10=50$ 이 되어 b는 50이 된다. 결과적으로 a=10, b=50이 된다.

5.10.(4)

<문제>

다음과 같은 자바프로그램에 대한 질문에 답하시오.

다음은 퀵 정렬을 이용하여 정렬하는 클래스이다. 이 클래스를 테스트하는 프로그램을 작성하시오. 즉, 일련의 데이터(입력 데이터의 끝은 0)을 읽어 정렬한 후 출력하는 프로그램을 작성하는 것이다.

<코드>

```
1 import java.util.Scanner;
2
3 class Sort{
4     void swap(int v[], int i, int j) {
5         int temp;
6         temp = v[i];
7         v[i] = v[j];
8         v[j] = temp;
9     }
10    public void Qsort(int a[], int left, int right) {
11        int pe;
12        int i, last;
13
14        if(left>=right) return;
15        pe = (left+right)/2;
16        swap(a, left, pe);
17        last = left;
18        for(i = left+1; i<=right; i++)
19            if(a[i]<a[left]) swap(a, ++last, i);
20        swap(a, left, last);
21        Qsort(a, left,last-1);
22        Qsort(a, left+1, right);
23    }
24 }
25
26 public class ch5_10_4 {
27     public static void main(String[] args) {
28         Scanner sc = new Scanner(System.in);
29         int a[] = new int[100];
30         Sort s = new Sort();
31         int cnt=0;
32
33         System.out.print("input: ");
34         while(true) {
35             int num = sc.nextInt();
36             if(num==0) break;
37             a[cnt]=num;
38             cnt++;
39         }
40
41         s.Qsort(a, 0, cnt-1);
42
43         System.out.print("quick sort result: ");
44         for(int i=0;i<cnt;i++) {
45             System.out.print(a[i]+" ");
46         }
47
48         sc.close();
49     }
50 }
```

<실행결과>

```
Problems @ Javadoc Declaration Console
<terminated> ch5_10_4 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 5:43:15 - 오후 5:43:37)
input: 10 5 3 4 1 6 8 9 7 2 0
quick sort result: 1 2 3 4 5 6 7 8 9 10
```

<설명>

퀵정렬은 정렬한 원소들 중 하나를 기준 값으로 선정하여 이 값보다 작은 원소들과 같거나 큰 원소들로 분리하여 이들 각각에 대해 순환적으로 퀵 정렬을 적용하여 크기순으로 정렬하는 알고리즘이다. 이때, 기준이 되는 값을 분할 원소라 부른다.

(코드 4-9) Swap 함수는 배열 v의 i 번째 인덱스와 j 번째 인덱스 내용을 바꿔준다. 이때 temp 변수를 사용해 i 번째 배열값을 저장하고 i 번째 인덱스에 j 번째 인덱스 값을 저장후 temp 에 저장된 값을 j 번째 인덱스에 저장한다.

(코드 10-23) 조건문에서 $left \geq right$ 라면 바로 해당 Qsort 메소드를 빠져나간다. 그리고 피벗 변수 pe 에 중앙자리를 설정한다. 그래서 $pe = (left + right) / 2$ 를 한다. 그리고 맨 왼쪽 값과 피벗값을 Swap 함수를 통해서 바꿔준다. 그리고 $last = left$ 로 설정한다. 그리고 for 반복문을 통해서 $i = left + 1$, 즉 피벗이 위치한 left 다음의 인덱스부터 맨 끝 인덱스 right 까지 반복문이 돌면서 만약에 $a[i]$ 가 $a[left]$ (피벗값)보다 작으면 last 를 1 증가시키고 그곳에 있는 값과 i 인덱스에 있는 값을 교환한다. 그리고 left 인덱스에 있는 값 즉 피벗값과 last 인덱스에 있는 값을 교환한다. 퀵정렬은 피벗값을 중심으로 작은값과 같거나 큰값을 분리해 위치시켜 $Qsort(a, left, last - 1); Qsort(a, left + 1, right);$ 를 호출한다. 즉 퀵정렬 알고리즘은 재귀호출을 통해서 이루어진다.

(코드 27~)메인메소드에서 사용자가 입력한 정수를 저장할 a배열을 100크기로 만든다. 그리고 Scanner를 통해 사용자의 정수를 입력받는다. while반복문을 통해 사용자가 0을 입력할때까지 입력받아 저장한다. 이때 nextInt()메소드를 통해 정수를 입력받는다. 이때 cnt=0으로 초기화하고 입력을 받아 배열a에 입력값을 저장할때마다 1증가시켜 몇개의 정수를 입력받았는지 알 수 있고, 추가적으로 $a[cnt]=num;$ 을 통해 정수가 어디에 저장되어야 할지 가리켜주는 역할도 해준다. 그리고 Sort클래스 객체 s에서 메소드 Qsort를 호출해 배열a와 0, cnt-1을 매개변수로 넘겨준다. 배열에서는 마지막 원소 인덱스가 배열크기-1이기 때문에 cnt-1로 한다. 그 후 정렬 결과를 확인하기 위해서 for반복문을 이용해 cnt번만큼 a배열을 출력한다. 그럼 정상적으로 정렬된 것을 확인할 수 있다.

6.6.(1)

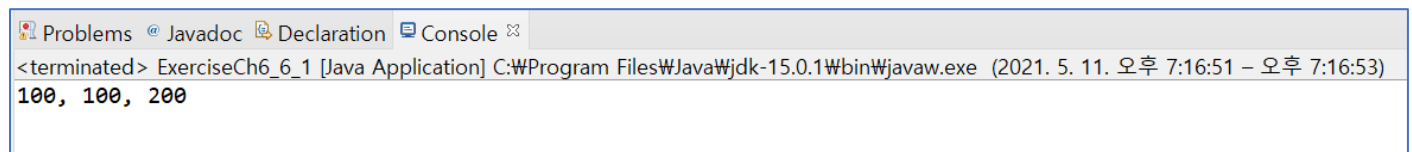
<문제>

다음 프로그램의 실행 결과를 쓰시오.

<코드>

```
1 class SuperClass{
2     static int supNum = 100;
3 }
4 class SubClass extends SuperClass{
5     static int subNum = 200;
6 }
7 public class ExerciseCh6_6_1 {
8     public static void main(String[] args) {
9         System.out.print(SuperClass.supNum+", "+SubClass.supNum);
10        System.out.println(", "+SubClass.subNum);
11    }
12 }
```

<실행결과>



Problems @ Javadoc Declaration Console

<terminated> ExerciseCh6_6_1 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 7:16:51 - 오후 7:16:53)

100, 100, 200

<설명>

SuperClass 클래스를 만들고 static변수 int형 supNum을 100으로 초기화한다.

SubClass클래스는 SuperClass를 상속받고 static 변수 int형 subNum을 200으로 초기화한다.

메인메소드에서는 SuperClass.supNum, SubClass.supNum, subclass.subNum을 출력한다.

우선 supNum, subNum변수 모두 static 변수이기 때문에 클래스이름으로 호출이 가능하다.

superClass.supNum은 SuperClass내의 supNum변수가 100이므로 100이 출력된다.

subClass.supNum은 SubClass클래스가 SuperClass클래스를 상속받고 있기 때문에 SuperClass변수인 supNum에 접근이 가능하므로 역시 100이 출력된다.

subclass.subNum은 subclass 클래스 내에 있는 subNum변수를 접근하므로 200이 출력된다.

6.6.(2)

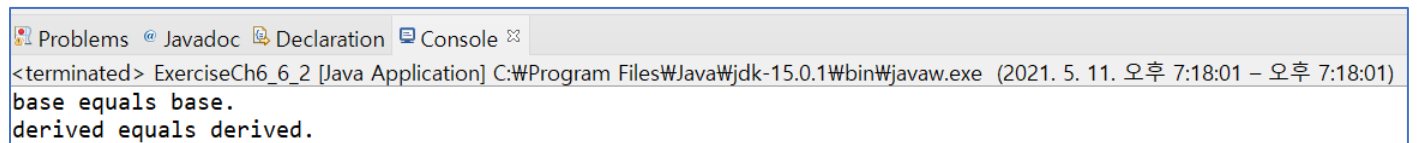
<문제>

다음 프로그램의 실행 결과를 쓰시오.

<코드>

```
1 class BaseClass{
2     public boolean equals(Object obj) {
3         if(getClass() == obj.getClass()) return true;
4         else return false;
5     }
6 }
7 class DerivedClass extends BaseClass{}
8 public class ExerciseCh6_6_2 {
9     public static void main(String[] args) {
10         BaseClass b1 = new BaseClass();
11         BaseClass b2 = new BaseClass();
12         DerivedClass d1 = new DerivedClass();
13         DerivedClass d2 = new DerivedClass();
14         if(b1.equals(d1)) System.out.println("derived equals base.");
15         if(d1.equals(b1)) System.out.println("base equals derived.");
16         if(b1.equals(b2)) System.out.println("base equals base.");
17         if(d1.equals(d2)) System.out.println("derived equals derived.");
18     }
19 }
```

<실행결과>



```
<terminated> ExerciseCh6_6_2 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 7:18:01 - 오후 7:18:01)
base equals base.
derived equals derived.
```

<설명>

BaseClass 클래스를 만들고 public으로 boolean형을 반환하는 equals메소드를 만든다. Object를 매개변수로 받는다. 여기서 getClass메소드를 통해 매개변수로 입력받은 클래스와 현재 BaseClass의 getClass가 같다면 true를 반환하고 아니면 false를 반환한다. 그리고 DerivedClass클래스를 만들고 BaseClass클래스를 상속받는다.

메인 메소드에서는 BaseClass객체 b1, b2를 만들고, DerivedClass 객체 d1, d2를 만든다.

getClass()는 클래스의 주소값을 리턴한다.

14줄부터 b1과 d1이 같은지를 보고 만약에 같다면 해당 문장을 출력한다.

첫번째 b1.equals(d1) 은 BaseClass와 DerivedClass가 다르므로 false

두번째 d1.equals(b1) 은 DerivedClass와 BaseClass가 다르므로 false

세번째 b1.equals(b2) 은 BaseClass와 BaseClass가 같으므로 true

네번째 d1.equals(d2) 은 DerivedClass와 DerivedClass가 같으므로 true

그래서 결과적으로 3,4 번째 조건문만 만족이되 실행결과와 같이 출력된다.

6.7.(1)

<문제>

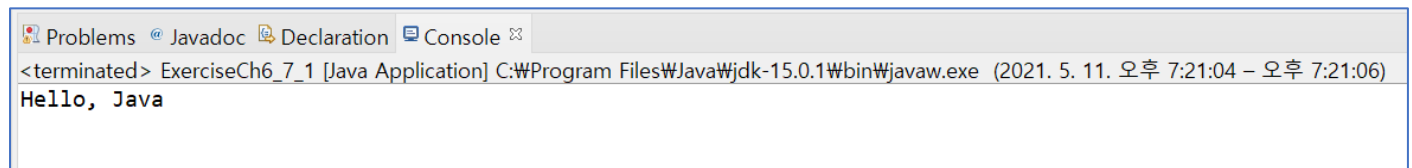
다음 프로그램의 에러를 모두 지적하고 옳게 고치시오.

<코드>

```
1 class SuperClass{
2     static String greeting() { return "Good Bye";}
3     String name() {return "Oak";}
4 }
5 class SubClass extends SuperClass{
6     String greeting() {return "Hello";}
7     String name() {return "Java";}
8 }
9 public class ExerciseCh6_7_1 {
10     public static void main(String[] args) {
11         SubClass s = new SubClass();
12         System.out.println(s.greeting()+" "+s.name());
13     }
14 }
```

```
1 class SuperClass{
2     static String greeting() { return "Good Bye";}
3     String name() {return "Oak";}
4 }
5 class SubClass extends SuperClass{
6     static String greeting() {return "Hello";}
7     String name() {return "Java";}
8 }
9 public class ExerciseCh6_7_1 {
10     public static void main(String[] args) {
11         SubClass s = new SubClass();
12         System.out.println(s.greeting()+" "+s.name());
13     }
14 }
```

<실행결과>



Problems @ Javadoc Declaration Console

<terminated> ExerciseCh6_7_1 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 7:21:04 - 오후 7:21:06)

Hello, Java

<문제>

오류수정 : 원본코드 6줄에 오류가 발생한다. SubClass가 SuperClass를 상속받기 때문에 SuperClass클래스 내에 있는 greeting메소드를 SubClass에서 오버라이딩을 해야 한다. 하지만 여기서 슈퍼클래스에서 static으로 되었으므로 서브클래스에서도 static으로 만들어 줘야한다. 혹은 슈퍼클래스의 greeting메소드에서 static을 지워도 된다.

SuperClass에서 greeting메소드는 "Good Bye"를 반환하고, name은 "Oak"를 반환한다. 서브클래스인 SubClass에서는 greeting메소드와 name메소드를 두개다 슈퍼클래스에서 오버라이딩하고 각각 "Hello"와 "Java"를 반환한다.

메인메소드에서 SubClass s = new SubClass();를 통해 객체를 만들어주고 서브클래스 객체 s의 메소드 greeting메소드와 name메소드를 호출한다. 여기서 상속받은 SubClass가 슈퍼클래스 메소드 greeting과 name을 모두 오버라이딩 하기 때문에 서브클래스에서 오버라이딩 된 메소드가 호출되 "Hello, Java"가 최종적으로 출력된다.

6.7.(2)

<문제>

다음 프로그램의 에러를 모두 지적하고 옳게 고치시오.

<코드>

```
1 class SuperClass{
2     int value;
3     SuperClass(int i){
4         value = i;
5     }
6     void output() {
7         System.out.println("SuperClass : "+value);
8     }
9 }
10 class SubClass extends SuperClass{
11     int value;
12     SubClass(int i){
13         value=1;
14     }
15     void output() {
16         System.out.println("SubClass : "+value);
17     }
18 }
19 }
20 public class ExerciseCh6_7_2 {
21     static void print(SubClass obj) {
22         obj.output();
23     }
24     public static void main(String[] args) {
25         SuperClass obj1 = new SuperClass(1);
26         SubClass obj2 = new SubClass(1);
27         print(obj1);
28         print(obj2);
29     }
30 }

1 class SuperClass{
2     int value;
3     SuperClass(int i){
4         value = i;
5     }
6     void output() {
7         System.out.println("SuperClass : "+value);
8     }
9 }
10 class SubClass extends SuperClass{
11     int value;
12     SubClass(int i){
13         super(i);
14         value=i;
15     }
16     void output() {
17         System.out.println("SubClass : "+value);
18     }
19 }
20 }
21 public class ExerciseCh6_7_2 {
22     static void print(SuperClass obj) {
23         obj.output();
24     }
25     public static void main(String[] args) {
26         SuperClass obj1 = new SuperClass(1);
27         SubClass obj2 = new SubClass(1);
28         print(obj1);
29         print(obj2);
30     }
31 }
```

<실행결과>

```
Problems Javadoc Declaration Console
<terminated> ExerciseCh6_7_2 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 11. 오후 7:26:00 - 오후 7:26:02)
SuperClass : 1
SubClass : 1
```

<설명>

오류수정 : 원본코드 2곳에서 오류가 발생한다. 12 ,27줄에서 오류가 발생하는데 우선 12줄에서 오류가 발생하는 이유는 26줄에서 SubClass객체를 만들 때 SuperClass를 상속받고 있기 때문에 SuperClass의 생성자도 호출이 되어야한다. 이때 SubClass 객체를 만들 때 매개변수 생성자를 호출한다. 그렇다고 하더라도 명시적으로 기입하지 않은이상 SuperClass에서는 매개변수 없는 생성자를 호출하게 된다. 하지만 superClass에는 매개변수있는 생성자만 있기때문에 결과적으로 매개변수없는 생성자가 호출을 못하게 된다. 그래서 SuperClass내에 매개변수 없는 생성자를 만들어 주거나 수정한 코드에서 13줄처럼 super(1)처럼 슈퍼클래스의 매개변수있는 생성자를 명시적으로 적어줘야 오류가 나지 않는다.

두번째 27줄 오류는 print메소드의 매개변수를 SubClass obj를 SuperClass obj로 수정한다. 기본적으로 superclass 객체는 SubClass를 통해 output메소드를 당연히 호출할 수 없다. 즉 원래 의도대로 동적 바인딩이 불가능하다.

SuperClass 클래스는 정수형변수 value를 가지고 매개변수 있는 생성자가 있고, 매개변수 i는 value에 저장된다. 그리고 output메소드는 value를 출력한다. SubClass클래스는 SuperClass클래스를 상속받아 동일한 value변수가 정의되고 매개변수 있는 생성자로 value를 초기화한다. 그리고 output메소드는 오버라이딩을 통해 구현해준다. 역시 value값을 출력한다.

메인메소드에서는 value값 1을 가지는 SuperClass 객체와 SubClass 객체를 만들어 static메소드인 print메소드에 넘겨주어 output메소드를 호출해준다. 여기서 SuperClass는 원래 자신의 output메소드 이고, SubClass 객체의 output메소드는 동적바인딩을 통해서 실행되어 "SuperClass : 1" "SubClass : 1"이 출력된다

6.10

<문제>

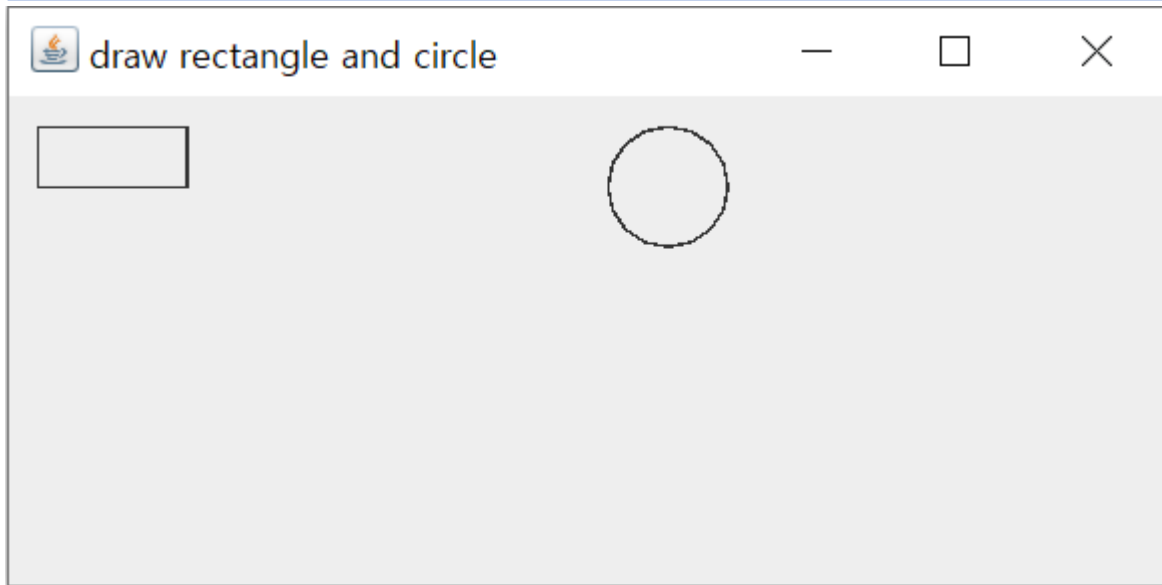
다음과 같은 추상 클래스를 이용하여 사각형과 원에 대한 클래스를 각각 정의하고 테스트하는 프로그램을 작성하시오. (8장 참조하여 구현)

<코드>

```
1 import java.util.Scanner;
2 import java.awt.*;
3 import javax.swing.*;
4
5 abstract class Figure{
6     abstract void area();
7     abstract void girth();
8     abstract void draw(Graphics g);
9 }
10
11 class Rect extends Figure{
12     private int width, height;
13     public Rect(int w, int h) {
14         width=w;
15         height=h;
16     }
17     public void area() {
18         System.out.println("Rect area=" + (width*height));
19     }
20     public void girth() {
21         System.out.println("Rect Girth= " + (2*width+2*height));
22     }
23     public void draw(Graphics g) {
24         g.drawRect(10, 10, width, height);
25     }
26 }
27
28 class Circle extends Figure{
29     private int radius;
30     public Circle(int r) {radius=r;}
31     public void area() {
32         System.out.println("Circle Rect= " + (3.14*radius*radius));
33     }
34     public void girth() {
35         System.out.println("Circle Girth= " + (2*3.14*radius));
36     }
37     public void draw(Graphics g) {
38         g.drawOval(200, 10, 2*radius, 2*radius);
39     }
40 }
41
42 class UserDefinedComponent extends JComponent{
43     public void paintComponent(Graphics g) {
44         Scanner sc = new Scanner(System.in);
45         System.out.print("사각형의 너비와 높이를 입력하시오: ");
46         int w = sc.nextInt();
47         int h = sc.nextInt();
48         Rect rect = new Rect(w, h);
49
50         System.out.print("원의 반지름을 입력하시오: ");
51         int r = sc.nextInt();
52         Circle circle = new Circle(r);
53
54         rect.area();
55         rect.girth();
56         rect.draw(g);
57
58         circle.area();
59         circle.girth();
60         circle.draw(g);
61
62         sc.close();
63     }
64 }
65
66 public class ch6_10{
67     public static void main(String[] args) {
68         JFrame frame = new JFrame();
69
70         frame.setTitle("draw rectangle and circle");
71         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
72         frame.setSize(400,200);
73
74         UserDefinedComponent component = new UserDefinedComponent();
75         frame.add(component);
76
77         frame.setVisible(true);
78     }
79 }
```

<실행결과>

```
Problems @ Javadoc Declaration Console
ch6_10 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 5. 13. 오전 10:32:42)
사각형의 너비와 높이를 입력하시오: 50 20
원의 반지름을 입력하시오: 20
Rect area=1000
Rect Girth= 140
Circle Rect= 1256.0
Circle Girth= 125.60000000000001
```



<설명>

추상클래스 Figure을 상속받아 Rect, Circle 클래스를 만들어 준다. 추상클래스에 있는 3개의 메소드는 모두 구현해야 한다. area메소드는 도형 넓이를, girth메소드는 도형 길이를 draw메소드는 실제로 그래픽을 통해 그려준다.

Rect클래스는 너비와 높이를 나타내는 private변수 width, height변수를 만들고 매개변수 있는 생성자를 통해 초기화할 수 있게 만들었다. Circle클래스는 반지름을 의미하는 private 정수형 변수 radius를 만들고 매개변수 있는 생성자를 통해 radius를 초기화 한다.

Rect클래스의 area메소드는 사각형 넓이가 가로*세로 이므로 width*height값을 출력해준다. girth메소드는 사각형 둘레길이이므로 $2 * width + 2 * height$ 를 출력해준다.

Circle클래스의 area메소드는 원의 넓이이므로 $3.14 * radius * radius$ 를 출력해주고, girth 메소드는 원의 둘레 길이이므로 $2 * 3.14 * radius$ 를 출력해준다.

메인메소드에서 JFrame 객체 frame을 만들어주고, frame의 상단바 제목을 setTitle메소드를 통해 지정해주고, setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);를 통해 창이 닫히면 완전히 프로그램이 종료될 수 있게 설정해준다. setSize메소드는 너비400, 높이 200으로 창 크기를 설정해주고 setVisible(true); 를 통해 보여주게 설정해준다.

그리고 JComponent를 상속받은 UserDefinedComponent클래스를 만들어주고 이 클래스를 객체를 만들어 아까 만든 frame에 frame.add(component);를 통해 JFrame에 추가시켜준다.

UserDefinedComponent클래스에서 Graphics를 매개변수로 가지는 paintComponent메소드를 오버라이딩하고 그 안에Scanner객체 sc를 만들어 사각형의 가로, 세로길이와, 원의 반지름 길이를 사용자로부터 입력받아 Rect객체와 Circle객체를 만들어준다. 그리고 각각 area메소드, girth메소드, draw메소드를 호출한다. Rect클래스와 Circle클래스의 draw메소드에는 매개변수 Graphics가 있다. 그래서 paintComponent를 통해 매개변수로 받는 Graphics g를 Rect, Circle클래스 객체의 draw메소드에 넘겨준다.

Rect클래스의 draw메소드에서는 g.drawRect(a,b,c,d);가 있다. (a,b)에서 c*d 크기의 사각형을 그린다. 여기서는 (10,10)부터 시작해서 너비 width, 높이 height인 사각형을 그릴 수 있다.

Circle클래스의 draw메소드에서는 g.drawOval(a,b,c,d);가 있다. 여기서 . (a,b)에서 c*d 크기의 사각형에 내접하는 타원을 그린다. 여기서는 (200,10)부터 시작해서 가로 세로 길이가 2*radius인 사각형에 내접하는 타원을 그려주게 된다. 저렇게 쓴 이유는 원의 지름길이가 2*radius이기 때문이다.

그래서 프로그램을 실행시키면 사각형과 원의 넓이와 둘레가 출력되고 사각형과 원이 그려지게 된다.