

[lab03 보고서]

디지털미디어학과

2019111677

김지연

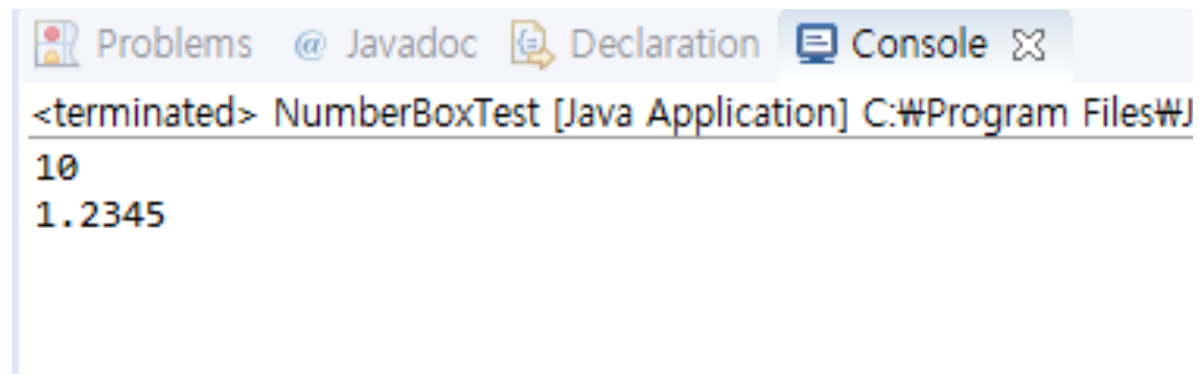
1. 다음과 같이 클래스가 정의되어 있다고 가정하다. 이 클래스의 객체를 생성하고 각 필드를 10과 1.2345로 초기화한 후 각 필드의 값을 출력하는 코드를 작성하세요. (이클립스에서 실행한 후, 소스코드 텍스트와 실행결과 이미지를 첨부할 것)

```
public class NumberBox {  
  
    public int ivalue;  
  
    public float fvalue;  
  
}
```

[소스코드]

```
public class NumberBoxTest {  
    public static void main(String[] args) {  
        NumberBox num = new NumberBox();  
        num.ivalue = 10;  
        num.fvalue = (float) 1.2345;  
  
        System.out.println(num.ivalue);  
        System.out.println(num.fvalue);  
  
    }  
}
```

[실행결과]



2. 다음 각 문장의 출력은 무엇인가요? 본인의 예상과 코드를 실행한 결과를 비교해 보세요. 그리고 출력 결과의 이유를 설명하세요.

```
(1) String verb = "현실이 된다";  
  
System.out.println("생각이" + verb);
```

생각이 현실이 된다 로 출력된다. 이 문장을 실행하면 String 클래스의 객체가 하나 생성되고, verb는 참조 변수로서 생성된 객체를 가리키는 변수가 된다. new 연산자를 사용하지 않아도 자동적으로 객체가 생성된다. 출력문에서 두 개의 문자열은 + 연산자를 이용하여 결합될 수 있기 때문에 결합하여 출력 가능하다.

```
(2) String s = "1234567";  
  
System.out.println("문자열의 길이는" + s.length());
```

문자열의 길이는7로 출력된다. String 클래스의 객체가 생성되고, 참조 변수 s는 문자열 1234567을 가리킨다. String 클래스는 length() 메소드를 가지고 있어 문자열의 길이를 계산하여 반환할 수 있다. 참조변수 s 뒤에 length() 메소드를 호출하여 문자열의 길이를 출력한다.

```
(3) String s = "ABCDEFGH";  
  
s.toLowerCase();  
  
System.out.println(s);
```

ABCDEFGH로 출력된다. toLowerCase() 메소드는 문자열의 문자들을 모두 소문자로 변경한다. 하지만 위의 코드에서는 s에 호출한 값을 저장하지 않고 s를 호출하였기 때문에 대문자 그대로 출력된다.

```
(4) System.out.println("2 + 3 = " + (2 + 3));
```

2 + 3 = 5로 출력된다. "2+3="은 그대로 출력된다. (2+3)은 더하기를 우선 계산하고, 답인 5는 + 연산자가 문자열로 적용되어 앞의 문자열에 결합되어 출력된다.

```
(5) System.out.println("2 + 3 = " + 2 + 3);
```

2 + 3 = 23으로 출력된다. "2+3="은 그래도 출력된다. 뒤의 2와 3은 각각 + 연산자가 문자열로 적용되기 때문에 23으로 출력된다.

3. 강아지를 나타내는 클래스 Dog을 정의하세요. 강아지 클래스는 품종(breed), 견령(age), 모색(color)의 필드를 가지며, barking(), hungry(), sleeping()의 메소드를 멤버로 갖습니다. (클래스 정의 코드만 작성)

[소스코드]

```
public class Dog {  
  
    String breed;  
    int age;  
    String color;  
  
    void barking() {  
        System.out.println("강아지가 짖는다.");  
    }  
    void hungry() {  
        System.out.println("강아지가 배고프다.");  
    }  
    void sleeping() {  
        System.out.println("강아지가 잔다.");  
    }  
}
```

4. 다음 프로그램의 출력은 무엇인가요? 본인의 예상과 코드 실행 결과를 비교해 보세요. 그리고 출력 결과의 이유를 설명하세요.

```
String a = "javaprogramming";  
a = a.substring(5,7);  
char b = a.charAt(1);  
a = a + b;  
System.out.println(a);
```

roo로 출력된다. a에는 javaprogramming에서 인덱스 값이 5인 것부터 7직전 값인 ro가 저장된다. b에는 a의 값 중 인덱스가 1인 문자를 반환하기 때문에 o이 저장된다. a와 b의 문자열을 합한 a를 출력하면 roo라는 문자열이 출력된다.

5. 다음 프로그램의 출력은 무엇인가요? 본인의 예상과 코드 실행 결과를 비교해 보세요. 그리고 출력 결과의 이유를 설명하세요.

```
String x = "abc";  
x.toUpperCase();  
String y = x.replace('a', 'd');  
y = y + "xyz";  
System.out.println(y);
```

dbcxyz가 출력된다. x에는 abc라는 문자열이 저장된다. toUpperCase()는 문자열의 문자들을 모두 대문자로 변경하는 메소드이지만 호출만 하고 값을 저장하지 않았기 때문에 실행되지 않는다. y에 x문자열 중 a를 d로 변경한 dbc문자열을 저장한다. y에 y문자열과 xyz를 결합한 dbcxyz를 저장하고 다음문장에서 출력한다.

6. 정육면체를 나타내는 클래스 Cube가 다음과 같은 경우, 아래 질문에 답하세요.

```
public class Cube {  
    private double side;        // 정육면체의 한 변의 길이  
    public double getSide() {  
        return side;  
    }  
    public double getVolume() {  
        return side*side*side;  
    }  
}
```

(1) 매개 변수가 없는 생성자를 정의해 보세요. 이 생성자는 side를 0으로 초기화 합니다.

[소스코드]

```
public class Cube {  
    private double side; // 정육면체의 한 변의 길이  
    public double getSide() {  
        return side;  
    }  
    public double getVolume() {  
        return side*side*side;  
    }  
    public Cube(){  
        side = 0;  
    }  
}
```

(2) 명시적 생성자를 중복 정의하세요. 이 생성자는 매개 변수로 전달된 값으로 side를 초기화 합니다.

[소스코드]

```
public class Cube {
    private double side; // 정육면체의 한 변의 길이
    public double getSide() {
        return side;
    }
    public double getVolume() {
        return side*side*side;
    }
    public Cube(){
        side = 0;
    }

    public Cube(double s){
        side = s;
    }
}
```

7. 책을 나타내는 Book 클래스를 정의해 보세요. Book 클래스는 제목(title)과 저자(author)를 나타내는 필드를 갖습니다. (필드는 모두 private으로 선언) 또한, 각 필드에 대하여 접근자와 설정자를 정의하세요.

Book 클래스에서 객체 묵시적으로 생성한 후, 설정자를 이용해서 필드를 초기화(초기값은 각자 결정) 하고 접근자를 이용해서 출력하는 테스트 코드를 작성하세요.

[소스코드]

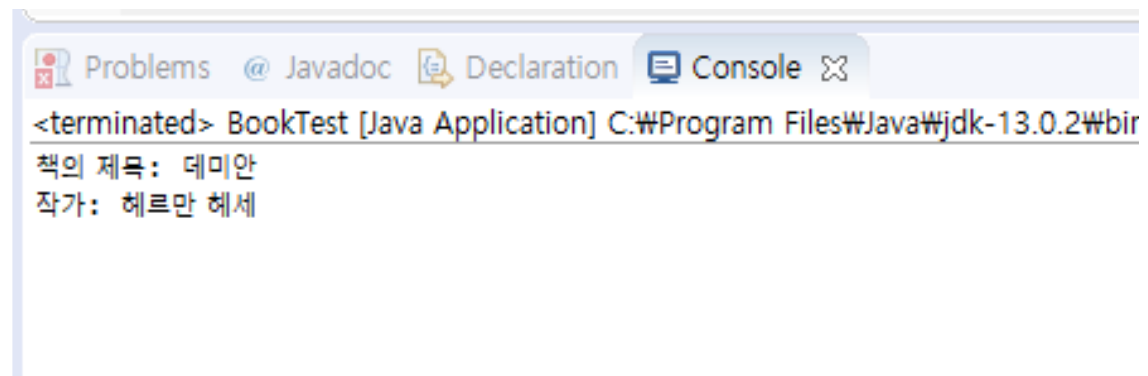
```
class Book {
    private String title;
    private String author;

    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }

    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
}
```

```
public class BookTest {  
    public static void main(String[] args) {  
  
        Book book = new Book();  
  
        book.setTitle( "데미안" );  
        book.setAuthor( "헤르만 헤세" );  
  
        System.out.println("책의 제목: " + book.getTitle() );  
        System.out.println("작가: " + book.getAuthor());  
  
    }  
}
```

[실행결과]



The screenshot shows an IDE's console window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the program. The output indicates that the program has terminated and shows the title and author of the book.

```
<terminated> BookTest [Java Application] C:\Program Files\Java\jdk-13.0.2\bin  
책의 제목: 데미안  
작가: 헤르만 헤세
```