

## JAVA 프로그래밍 2 ~ 3강 보충학습

### 제 2 장 - 자바 기본 문법 연습 문제

1. C/C++ 프로그램에서는 대소문자가 구별된다. 자바 프로그램은 어떠한가?

(답) 마찬가지로 구별된다.

2. 클래스 이름은 명사형으로 붙이고 대문자로 시작하며 메소드 이름은 동사형의 소문자로 시작하는 관례가 있다. 반드시 지켜야 하는가?

(답) 반드시 그렇지는 않으나 일반적 관례이다.

3. 자바에서 제공하는 기본 자료형을 모두 나열하라.

(답) byte, short, int, long, float, double, char, boolean

4. 255의 2진수 "0000 0000 1111 1111"을 16진수로 표현하라. 또한 이것의 2의 보수를 구하고 마찬가지로 16진수로 표현하라.

(답) 255의 16진수 표현: 0x00FF

2의 보수로 -255의 표현 : 1111 1111 0000 0001, 0xFF01

정수형의 음수 표현은 2의 보수를 사용하여 표현한다. 2의 보수는 1의 보수에 1을 더하여 얻어지며 1의 보수는 이진수에서 0을 1로 1을 0으로 바꾸면 얻을 수 있다. 255와 -255의 이진 표현을 더해보면 0이 됨을 알 수 있다.

정수를 2의 보수 방식으로 표현할 경우 모든 양수는 왼쪽 첫 비트의 값이 항상 0이고

모든 음수는 왼쪽 첫 비트의 값이 항상 1이다.

정수형 리터럴의 경우 숫자로 시작하면 10진수를 의미하고 '0'으로 시작하면 8진수이며 '0x'로 시작하면 16진수를 의미한다.

5. 자바 언어에서 char형은 부호 없는 정수로 취급될 수 있다. 이 때 최대값은 얼마가 되는가?

(답)  $2^{16}-1$ 로 65535

6. 다음 중 문제가 없는 것은 무엇인가?

- |                      |                       |                 |
|----------------------|-----------------------|-----------------|
| a) char c="cafe";    | b) char c='cafe';     | c) char c='\\"; |
| d) char c='\\ucafe'; | e) char c='\\u10100'; |                 |

(답) d) char c = '\\ucafe';

☞ 문자형에는 2byte 길이를 갖는 char형이 있다. 자바에서 사용되는 문자코드로는 2byte 길이를 갖는 UNICODE를 사용한다. char형은 정수형과 호환이 되는데 이때 char 형은 부호 없는 정수로 취급되어 표현 범위가 0 ~  $2^{16}-1$ 이 된다.

1번은 큰따옴표로 표현되는 문자열이므로 틀렸으며, char형에는 문자 한개만 넣을 수 있으므로 2번도 틀리다. 3번도 틀린데 '\\"가 아니고 '\"라면 맞다.

'\u'는 유니코드 문자열을 표현할 때 쓴다. 키보드만으로 유니코드를 입력할 수 없을 때 '\u' 뒤에 16진수 네 자리(2byte)로 나타내면 그 값에 해당하는 유니코드 문자가 표현된다. 5번은 4자리가 나와야 하는데 5자리수가 나왔기 때문에 잘못되었다. 4번의 cafe는 16진수 네 자리로 이것에 해당하는 문자는 '꺠'이다.

7. 다음과 같은 자바 코드가 있다고 할 때, 컴파일 에러가 있는가? C/C++에서는 어떠한가?

```
while (3*1) { }
```

(답) 괄호 안에 결과가 boolean인 식이 나와야하므로 에러가 있다. C/C++에서는 문제가 없으며 괄호 안은 참으로 해석된다.

8. 다음과 같이 double 형의 값을 float 형의 변수에 대입하려는 경우 에러가 발생한다. 이것을 강제로 수행시키는 형변환 방법은 무엇인가?

```
double d = 3 ;
```

```
float f = d;
```

(답) double d = 3; float f = (float)d;

☞ 자바에서는 소스 파일이 컴파일 될 때 변수나 상수의 자료형을 엄격하게 검사한다. 만일 변수나 상수에 대입하는 값이 변수나 상수가 선언될 당시의 자료형과 일치하지 않을 경우에 에러를 발생시키거나 형변환을 일으킨다.

같은 종류의 자료형이면서 자료의 손실이 발생하지 않을 경우에는 실제 자료형이 다르더라도 묵시적으로 형변환을 해준다. 그러나 그 반대의 경우 에러가 발생하는데 이 문제에서는 double 형이 float 형보다 표현범위가 넓은데 double 형의 값을 float 형에 대입하려고 하기 때문에 에러가 난다.

따라서 '(자료형)'의 형변환 연산자를 사용하여강제적으로 형변환을 시켜주면 에러가 나지 않는다. 그렇지만 자료의 손실이 발생할 수도 있다.

9. 상수의 선언을 위해 필요한 자바 키워드는 무엇인가?

(답) final

10. boolean 형의 값에 논리 연산자 &&와 &이 적용될 때, 둘의 차이점은 무엇인가?

(답) 똑같이 둘 다 true인 경우에만 결과가 true가 되며 나머지는 결과가 false이 된다. 단 && 연산에서는 숏-서킷(short-circuit) 로직을 사용한다. 즉 첫 번째 피연산자가 false이면 두 번째 피연산자가 계산되지 않는다.

11. 비트이동연산자 >>와 >>>의 차이는 무엇인가?

(답) >> 연산자는 부호를 확장시켜 왼쪽 부분을 채우지만 >>>는 그렇지 않다.

12. 다음 문장을 수행한 후 i와 n이 가지는 값은 얼마인가?

```
int i = 10;
int n = i++%5;
```

(답) i는 11이고 n은 0

13. 다음 프로그램의 출력 결과는 무엇인가?

```
class Bitwise {
    public static void main(String[] args) {
        int x = 0x9C;
        int y = 1<<8;
        int z = x&y;
        System.out.println(x+" "+y+" "+z);
    }
}
```

(답) 156 256 0

x는 16진수로 9C이므로 2진수로는 00000000 00000000 00000000 10011100이고 10진수로 바꾸면 156이다.

y는 이진수 1을 8비트 왼쪽으로 이동시킨 것으로 이진수 00000000 00000000 00000001 00000000이고 10진수로 바꾸면 256이다.

z는 x와 y를 비트별로 논리곱 한 것이므로 결과는 00000000 00000000 00000000 00000000이 되어 10진수로 0이 된다.

14. 아래와 같은 명령행으로 프로그램을 실행하였을 때 결과는 무엇인가?

c:\W>java TestJava 123

```
public class TestJava {
    public static void main(String args[]) {
        System.out.println(args[1]);
    }
}
```

(답) “ArrayIndexOutOfBoundsException” 예외가 발생하면서 프로그램이 종료된다.

명령행 인자는 프로그램을 실행시킬 때 프로그램 이름 뒤에 나열하는 인자를 말한다. 자바 프로그램을 실행시킬 때는 ‘java 클래스이름’의 형식으로 실행시키는데 이때 클래스 이름 뒤에 나열하는 인자를 명령행 인자라고 한다.

명령행 인자로 넘겨준 데이터는 문자열 배열인 args에 저장되어 main 메소드에 전달된다. 만일 데이터를 문자열로 다루고자 할 때에는 별다른 처리 없이 그냥 사용하면 되고 데이터를 정수형으로 바꿔 사용하고자 할 경우에는 Integer.parseInt() 메소드를 사용한다.

1개 이상의 인자도 가능하며 이 인자들은 띄어쓰기나 개행으로 구분되며, 각각 배열 원소 args[0], args[1], ... 등에 저장된다. 이 문제에서는 인자가 1개이므로 args[0]에 문자열 "123"이 들어가는데, 위 프로그램처럼 args[1]에 접근하는 경우에는 `ArrayIndexOutOfBoundsException`이라는 예외가 발생한다. 이 예외는 `RuntimeException`의 한 종류이며 예외 처리가 없다면 발생과 동시에 프로그램은 종료된다.

15. 다음 프로그램의 컴파일 결과를 예상하라.

```
class Test {
    static int myArg=1;
    public static void main(String[] args){
        int myArg;
        System.out.println(myArg);
    }
}
```

(답) myArg 변수의 초기 값이 주어지지 않아 컴파일 에러가 난다.

일반적으로 변수의 유효범위는 변수가 선언/정의된 곳에서부터 변수의 선언/정의를 포함하는 블록의 맨 끝까지 이다. main 메소드에서 선언된 변수는 main 메소드 내부가 그 변수의 유효 범위가 된다. 이 문제의 "System.out.println(myArg);" 문장에서 myArg 변수는 main 메소드 위에 선언된 myArg 변수가 아닌 main 메소드 안에 선언된 myArg를 가리킨다. 이렇게 메소드 안에 선언된 변수를 로컬 변수라 하는데 초기 값이 자동으로 주어지지 않으므로 사용 전에 반드시 변수의 값을 주어야 한다.

컴파일을 해보면 변수가 초기화 되지 않았다는 의미로 "variable myArg might not have been initialized" 메시지가 나타난다. 만약 프로그램에서 "int myArg;" 문장이 없다면 클래스 Test에서 정의된 static 변수 myArg의 값이 출력된다.

16. 다음 프로그램에서 메소드 guessWhat()의 기능은 무엇인가? 출력 결과는 무엇인가?

```
class Test {
    public static void main(String args[]) {
        int[] x={0, '0', 'A', 'a'};
        System.out.println(guessWhat(x));
    }
    public static int guessWhat(int arr[]) {
        int x=-1;
        for(int i = 0 ; i < arr.length ; i++)
            x = x < arr[i] ? arr[i] : x ;
        return x;
    }
}
```

(답) guessWhat은 배열에서 가장 큰 값을 반환하며 출력 결과는 97이다.

"A ? B : C ;" 문장은 A의 결과가 참(true)이면 B의 결과가 전체 문장의 결과가 되고, 거짓(false)이면 C의 결과가 전체 문장의 결과가 되게 하는 제어문이다. 이 문제에서 " x = x < arr[i] ? arr[i] : x ;" 문장은 arr[i]가 x보다 크면 x에 arr[i] 값을 넣고, arr[i]가 x보다 작거나 같으면 x값을 그대로 유지하라는 뜻이다. 따라서 배열의 원소를 인덱스 0부터 검사하면서 큰 값을 만나면 그 값을 x의 값으로 하는 것이다. 결과적으로 guessWhat()

함수는 배열 arr의 원소를 인덱스 0부터 차례대로 검사하면서, 변수 x에 더 큰 값을 넣어주며 최종적으로 가장 큰 값을 반환한다. 이 프로그램에서 주어지는 배열의 원소 값이 각각 0, 48, 65, 97이므로 출력 결과는 97이다.

17. 다음의 출력 결과를 생성하는 자바 프로그램을 2차원 배열을 사용하여 작성하여라.

```
*
**
***
****
*****
```

(답)

```
class Test{
    public static void main(String[] args){
        //2차원 배열을 선언하고, 1차원 원소들을 먼저 할당함.
        char arr[][]=new char[5][];
        int i,j;
        for(i=0; i<arr.length ; i++){
            arr[i]=new char[i+1]; // 2차원 원소들을 할당함.
            for(j=0; j<arr[i].length;j++)
                arr[i][j]='*'; // 배열의 원소를 '*'로 초기화 함.
        }
        for(i=0; i<arr.length ; i++){
            for(j=0; j<arr[i].length;j++){
                System.out.print(arr[i][j]); // '*'를 출력
            }
            System.out.println(""); // 1차원별로 개행을 해준다.
        }
    }
}
```

18. 2에서 1,000,000 사이에 존재하는 소수를 구하는 프로그램을 작성하라.

(답)

```
class Test {
    public static void main(String[] args) {
        boolean isPrime;
        int start = Integer.parseInt(args[0]);
        int end = Integer.parseInt(args[1]);
        for (int i=start; i<=end; i++) {
            isPrime = true;
            for (int j=2; j<=(int)Math.sqrt(i); j++)
                if (i%j==0) {
                    isPrime = false;
                    break;
                }
            if (isPrime) System.out.println(i);
        }
    }
}
```