

수행 테스트 (변수, 자료형, 연산자)

2/29

이름 :

닉네임 :

1. 자바의 특징인 "플랫폼 독립성"이란 무엇을 말하는지 설명하시오

어떤 운영체제에서라도 동일한 코드로 동일한 결과를 얻을 수 있도록 자바가 상어선 (JVM) 이 코드영역을 관리하는 것.

2. 변수의 선언법 : Integer 타입의 변수 number를 선언하고 정수값 10을 값으로 입력하시오

`int number = 10`

3. 변수의 선언법 : String 타입의 변수 str을 선언하고 문자열 "Hello World"를 입력하시오
(String의 선언법이 두가지가 있는데 두가지를 모두 적으면 가산점 있음. 각 방법이 메모리 생성위치가 다름)

`String str = "Hello World"`

`String str = new String("Hello World")`

4. 자바의 기본 자료형 8가지를 적으시오

`byte, short, int, long, char, boolean, float, double`

5. 자바의 변수 이름짓기에서 사용할 수 있는 특수문자 2개를 적으시오

`_, $`

6. boolean 자료형이 가질 수 있는 두 가지 값을 영어로 적으시오

`true, false`

7. 산술연산자 중에 나누기를 수행한 이후의 나머지를 나타내는 연산자를 적으시오

`%`

8. 아래 코드의 네모안에 적당한 논리 연산자를 적으시오

`System.out.println(true false); // false가 출력되어야 함`

9. 아래 코드의 출력값을 적으시오

`int a = 5;`

`int b = 3;`

`int c = 4;`

`System.out.println(! (a < b || c % 2 == 0));`

`false`

10. 변수인 x 가 60보다 크거나 같으면 "합격입니다"를 출력하고 작으면 "탈락입니다"를 출력하도록 삼항연산자를 만드시오

`x >= 60 ? "합격입니다" : "탈락입니다"`

수행 테스트 (제어문,배열,문자열)

03/08

이름 :

닉네임 :

1. 아래 코드의 결과와 같이 배열의 참조복사를 하면 변수 하나를 수정할 경우 다른 하나도 같이 값이 바뀌는 이유를 설명하시오

```
int intArray1 = { 1, 2, 3 };
int intArray2 = intArray1;
intArray2[0] = 5;
System.out.println("Arrays.toString(intArray1) : " + Arrays.toString(intArray1));
System.out.println("Arrays.toString(intArray2) : " + Arrays.toString(intArray2));
```

배열과 같은 객체의 변수는 해당 객체가 생성되는 Heap 영역의 메모리 주소를 저장하고 있음. 즉, 두개의 변수 intArray1과 intArray2는 동일한 주소를 가지고 있으므로 동일한 메모리 주소에 있는 데이터를 공유함

2. score 변수가 90 이상이면 A, 80 이상 90 미만이면 B, 70 이상 80 미만이면 C, 60 이상 70 미만이면 D, 그 외 나머지는 F학점을 출력하는 아래 코드의 조건식을 완성하시오

```
if (
    System.out.println("A학점");
else if (
    System.out.println("B학점");
else if (
    System.out.println("C학점");
else if (
    System.out.println("D학점");
else
    System.out.println("F학점");
```

Score >= 90

Score >= 80 && Score < 90

Score >= 70 && Score < 80

Score >= 60 && Score < 70

3. 아래 코드의 출력결과를보고 for 반복문의실행 순서를 순서대로 적으시오. (반복문이므로 전체 반복되는 횟수를 감안하여 모두 적어야 함)

```
int intArray = { 1, 2 };
for (int i=0; i<intArray.length; i++)
    System.out.println(i);
```

System.out.println("END");

출력결과:

1
2
END

① ② ④ ③

② ④ ③

② ⑤

i=0

i=1

i=2

4. 아래 코드의 출력되는 결과를 적으시오

```
int[] intList = {1,2,3,4,5};
for (int i=0; i<intList.length; i++) {
    if (intList[i] > 3) {
        break;
    }
    System.out.println(intList[i]);
}
```

1
2
3

5. 출력 결과에 맞게 아래 코드의 네모에 적당한 코드를 넣으시오

```
int[] intList2 = {1,2,3,4,5};
for (int i=0; i<intList2.length; i++) {
    if (intList2[i] == 3) {
        _____
    }
    System.out.println(intList2[i]);
}
```

continue

출력결과:

1
2
4
5

6. String 타입의 배열을 colorList 라는 변수명을 사용하여 선언하고 new 키워드를 이용하여 생성한 후, "빨강", "초록", "파랑"를 순서대로 입력하시오

String[] colorList = new String[] { "빨강", "초록", "파랑" }

또는

String[] colorList = new String[3]

colorList[0] = "빨강"

colorList[1] = "초록"

colorList[2] = "파랑"

7. 아래 코드를 실행했을때출력되는 boolean 값을 쓰시오

```
String str1 = "Java";  
String str2 = "Java";  
String str3 = new String("Java");  
System.out.println(str1 == str2);  
System.out.println(str1 == str3);
```

true
false

8. 아래 코드는 컴파일 에러가 발생하는 상태입니다. 네모 속의 코드를 수정하십시오. (타입 변환이 필요)

```
int a = 1000;  
String str = a; // 에러 발생!!
```

`String str = String.valueOf(a)`

9. 아래 코드의 두 문자열 변수를 대소문자 구분없이 비교하기 위하여 사용할 수 있는 메소드를 적으시오

```
String name1 = "Abc";  
String name2 = "abc";  
System.out.println
```

`name1.equalsIgnoreCase(name2)`

// true

10. 아래 배열에서 가장 큰 수를 찾는 코드를 완성하십시오(2점)

28 3

🔍 복사 📄 클립 ...

```
int[] intArr = {19, 4, 33, 30, 34, 22};  
int max = 0;
```

```
for (int i=0; i<intArr.length; i++) {  
    if (intArr[i] > max) {  
        max = intArr[i];  
    }  
}
```

```
System.out.println(max); // 34
```