


평가일시	2021년 05월 07일	훈련생	
과정명	자바(JAVA)기반 클라우드 융합 SW개발자-A		
교과목	프로그래밍 언어활용 (2001020231_19v4/ 3수준)	훈련교사	최수열 
능력단위요소명	1. 구조적 프로그래밍 언어 활용하기 2. 객체지향 프로그래밍 언어 활용하기 3. 스크립트 활용하기		
성취기준	5점(매우우수), 4점(우수), 3점(보통), 2점(미흡), 1점(매우미흡)		

<p>[서술형 문항1]</p> <p>웹프로그래밍에서 꼭 필요한 자바스크립트에 대해서 간단히 서술하시오</p> <p>답:</p> <p>자바스크립트란 웹 개발에서 여러 가지 이벤트를 적용시키기 위하여 사용하는 언어이다</p>	점수
<p>[서술형 문항2]</p> <p>자바로 프로그래밍에 필요한 JDK와 JRE의 차이점에 대해서 서술하시오</p> <p>답:</p> <p>JRE:</p> <p>먼저 JRE는 Java Runtime Environment의 약자로 자바 프로그램을 실행시켜주는 환경을 구성해 주는 도구다. 즉 JAVA를 개발할 필요는 없는데, 실행은 시켜줘야 하는 경우에는 꼭 JRE가 있어야 한다.</p> <p>JDK:</p> <p>JDK는 Java Development Kit의 약자로 말 그대로 자바 개발 시 필요한 툴킷을 제공하는 도구 모음이다. 개발하려면 당연히 실행도 시켜야 하므로 JDK 안에는 JRE가 포함되어 있다.</p>	
<p>[서술형 문항3]</p> <p>자바스크립트와 자바의 데이터 타입은 차이가 있다. 자바스크립트의 데이터 타입과 자바의 기본 데이터 타입을 나열하고 각 타입에 대해서 간단하게 서술하시오</p> <p>답:</p> <p>JS :</p> <p>Number : 숫자를 저장하는 타입, 64비트 부동 소수점 형태로 저장</p> <p>string : 문자를 저장 할때 사용 하는 타입</p> <p>boolean : 논리값을 저장하기 위해 사용</p> <p>undefined: 선언 먼 되고 한 번도 사용되지 않은 타입</p> <p>null : 변수 사용 후 의도적으로 '값이 없음'을 명시하기 위해 할당하는 값</p> <p>java :</p>	

<p>byte : 정수 타입중 가장 작은 범위의 수를 저장하는데 사용</p> <p>char : 유니코드를 문자 혹은 정수로 저장하는데 사용</p> <p>short : c언어와의 호환을 위해 사용하지만 java에서는 잘 사용되지 않는다</p> <p>int : 자바의 정수 데이터를 저장하기 위한 기본 데이터 타입</p> <p>long : 수치가 큰 데이터를 저장하기 위해 사용</p> <p>float , double : 실수를 저장하기위해 사용</p> <p>boolean : 논리값을 저장하기 위해 사용</p>	
<p>[서술형 문항4]</p> <p>프로그래밍에서 빠질 수 없는 요소로 변수가 있다. 이 변수에 대해서 서술하시오</p> <p>답:</p> <p>변수란 프로그래밍 시 메모리에 데이터를 저장하기 위하여 선언하는 것입니다.</p> <p>변수명은 어떤 규칙 안에서 자유롭게 이름을 지을 수 있습니다.</p> <p>조건으로는</p> <ol style="list-style-type: none"> 1. 변수명에는 영문과 숫자와 언더파(_)만 들어간다. 2. 변수명은 영문의 대문자와 소문자를 구별한다. 3. 변수 명의 첫 문자는 숫자가 올 수 없다. 4. 예약어는 사용할 수 없다. (예, if, for, while 등 외 다수) 	
<p>[서술형 문항5]</p> <p>자바에서는 여러 가지 데이터 타입이 존재하고 각각의 데이터 타입에서 다른 데이터 타입으로 변경되는 것을 '타입 변환'이라고 한다. 여기서 자동 타입 변환과 강제 타입 변환에 대해서 서술하시오</p> <p>답 :</p> <p>자동 타입 변환</p> <p>자동 타입 변환은 프로그램 실행 도중에 자동적으로 타입 변환이 일어나는 것을 말한다.</p> <p>자동 타입 변환은 작은 크기를 가지는 타입이 큰 크기를 가지는 타입에 저장될 때 발생한다.</p> <p>예) byte byteValue = 10;</p> <p>int intValue = byteValue;</p> <p>byteValue 가 int 타입으로 자동 변환</p> <p>강제 타입 변환</p> <p>큰 크기의 타입은 작은 크기의 타입으로 자동 타입 변환을 할 수 없다.</p> <p>마치 큰 그릇의 물을 작은 그릇 안에 모두 넣을 수 없는 것과 동일한 이치이다.</p> <p>하지만 큰 그릇을 작은 그릇을 작은 그릇 사이즈로 쪼개어서 한 조각만 작은 그릇에 넣는다면 가능하다.</p> <p>예) int intValue = 100000000;</p> <p>byte byteValue = (int) intValue;</p> <p>intValue 가 byte 타입으로 강제 변환 되어 값의 일부를 잘라서 표시된다.</p>	
<p>[서술형 문항6]</p>	

아래는 강제 타입 변환에 대한 내용입니다. 컴파일 에러가 발생하는 것은 무엇이며, 발생 원인에 대해서 서술하시오

```
int intValue = 10;
char charValue = 'A';
double doubleValue = 5.7;
String strValue = "A";
```

① double var = (double) intValue;

② byte var = (byte) intValue;

③ int var = (int) doubleValue;

④ char var = (char) strValue;

답:

4번

발생 원인 :

char 타입은 정수형 타입이면서 문자열 타입이지만 String 타입은 문자형 클래스 타입이라 강제 타입 변환을 할 수 없음.

[서술형 문항7]

프로그램에서의 연산, 연산자, 피연산자, 연산식에 대해서 간단하게 서술하시오

답:

프로그램에서 데이터를 처리하여 결과를 산출하는 것을 연산이라고한다.

연산에 사용되는 표시나 기호를 연산자라고 한다.

연산되는 데이터는 피연산자라고 한다.

연산자와 피연산자를 이용하여 연산의 과정을 기술한 것을 연산식이라고 부른다.

[서술형 문항8]

다음은 10과 20을 더해 합을 출력하려고 작성한 자바 프로그램이지만 오류가 있다.

```
public class Calculator {
    int i;
    int j;
    i = 10;
    j = 20;
    System.out.println(i + j);
}
```

오류를 찾아내서 수정하시오

답:

int i = 0;

int j = 0;

i = 10;

j = 20;

System.out.println(i + j);

[서술형 문항9]

위 8번 문항의 이 자바 클래스를 저장하는 소스 파일의 이름은 무엇인가?

답:

Calculator.java

[서술형 문항10]

위 8번 문항의 이 자바 프로그램을 콘솔에서 실행하는 방법은 무엇인가?

[서술형 문항11]

아래의 소스를 실행했을 때 결과는 무엇이며, 이유를 서술하시오

```
public class Exercise {  
    public static void main(String[] args) {  
        int x = 10;  
        int y = 20;  
        int z = (++x) + (y--);  
        System.out.println(z);  
    }  
}
```

답:

결과 : 31

이유 : 변수 z의 계산식에서 x의 증감식은 x의 앞에 붙어 먼저 계산하고, y의 증감식은 뒤에 붙어 z의 계산식이 끝난 후 계산하므로 z의 결과는 31이 나옴

[서술형 문항12]

다음 코드의 실행 결과는 무엇인가? i가 6의 배수인 경우 “6의 배수”를, i가 홀수인 경우 “홀수”라고 출력하며, 그 외 경우는 아무것도 출력되지 않게 하고자 한다. 소스를 올바르게 수정하시오

```
int i = 4;
if (i % 2 == 0) {
    if (i % 3 == 0) {
        System.out.println("6의 배수");
    }
}
else {
    System.out.println("홀수");
}
```

답:

```
int i = 4;
```

```
if (i%6 == 0) {
```

```
System.out.println("6의 배수");
```

```
}
```

```
if (i%2 != 0){
```

```
System.out.println("홀수");
```

```
}
```

실행결과 : 아무것도 출력되지 않음

[서술형 문항13]

break문과 continue문의 차이점에 대해서 서술하시오

답:

break 문은 해당 조건을 만났을때 반복문에서 탈출 하는 것

continue 문은 해당 조건을 만났을때 계속 반복문을 계속 진행 하는 것

[서술형 문항14]

다음 if 문을 switch문으로 수정하시오.

```
if (in.equals("가위")) {  
    System.out.println(1);  
}  
else if (in.equals("바위")) {  
    System.out.println(2);  
}  
else if (in.equals("보")) {  
    System.out.println(3);  
}  
else {  
    System.out.println(0);  
}
```

답:

```
switch (in) {  
case "가위":  
    System.out.println 1;  
    break;  
case "바위":  
    System.out.println 1;  
    break;  
case "보":  
    System.out.println 1;  
    break;  
default  
    System.out.println 0;  
}
```

[서술형 문항15]

아래의 소스에서 선언된 두 변수의 값이 동일하게 보이지만 동일하지 않다고 출력되는 이유에 대해서 서술하시오

```
String str1 = "Java 프로그래밍";
String str2 = new String("Java 프로그래밍");

System.out.println("str1 : " + str1);
System.out.println("str2 : " + str2);

System.out.println(str1 == str2);
```

<결과>

```
str1 : Java 프로그래밍
str2 : Java 프로그래밍
false
```

답:

String 타입의 변수는 참조 변수이므로 그 값에 데이터를 직접 저장하는 것이 아니라 주소를 저장한다.
위에서 사용된 연산식은 참조 변수에서 주소의 값이 같은지 비교하는 것이므로 서로의 참조 변수 주소가 다른 두 변수의 변수값은 같지만 연산식의 결과는 다르다고 출력된다.

[서술형 문항16]

자바의 3가지 메모리영역에 대해서 간단히 서술하시오

답:

3가지 메모리 영역으로는 메서드 영역, JVM 스택 영역, 힙 영역 이 있다
메서드 영역은 JVM이 시작할 때 생성되고 모든 스레드가 공유하는 영역이다.
JVM 스택 영역은 각 스레드마다 하나씩 존재하며 스레드가 시작될 때 할당된다.
힙 영역은 객체와 배열이 생성되는 영역이다.

[서술형 문항17]

자바스크립트와 자바 모두 배열을 사용하지만 두 언어의 배열은 차이점이 존재한다. 두 언어의 차이점에 대해서 서술하시오

답:

js 와 java 모두 개념은 동일하지만
java의 배열은 데이터 값을 저장하는 것이 아니라 참조 변수의 주소값을 저장한다.

[서술형 문항18]

배열을 선언 시 `int[] arrays = new int[5];` 와 같이 배열의 크기만 선언하고, 데이터를 저장하지 않을 경우 배열의 초기값이 어떻게 지정되는지 데이터 타입별로 초기값을 나열하시오

```
byte : 0
char : '\u0000'
short : 0
int : 0
long : 0L
float : 0.0F
double : 0.0
```

boolean : false

클래스 : null

인터페이스 : null

[서술형 문항19]

아래의 소스코드 중 틀린 곳을 수정하시오

```
int scores;
scores = {83, 90, 87};

System.out.println("scores[0] : " + scores[0]);
System.out.println("scores[1] : " + scores[1]);
System.out.println("scores[2] : " + scores[2]);

int sum = 0;
for (int i = 0; i < 3; i++) {
    int sum += scores[i];
}

System.out.println("총합 : " + sum);
double avg = (float) sum / 3;
System.out.println("평균 : " + avg);
```

<결과>

```
scores[0] : 83
scores[1] : 90
scores[2] : 87
총합 : 260
평균 : 86.66666666666667
```

답:

```
int scores[] = { 83, 90, 87 };
System.out.println("scores[0]:" + scores[0]);
System.out.println("scores[1]:" + scores[1]);
System.out.println("scores[2]:" + scores[2]);

int sum = 0;
for (int i = 0; i < 3; i++) {
    sum += scores[i];
}

System.out.println("총합 : " + sum);
double avg = (double) sum / 3;
System.out.println("평균 : " + avg);
```

[서술형 문항20]

아래의 소스와 같이 배열을 복사하면 발생하는 문제점과 정상적으로 배열을 복사하기 위한 방법에 대해서 간단히 서술하시오


```
int[] scores = new int[] {80, 90, 87};
int[] copyScores = new int[3];
```

```
copyScores = scores;
```

답:

문제점 :

보기의 코드로 배열을 복사하면 배열의 값이 복사되는 것이 아니라 배열의 주솟값이 복사가 되므로 정상적인 복사가 되는 것이 아님

정상적인 복사 방법:

```
int scores[] = { 80, 90, 97 };
int copyScores[] = new int[5];
for (int i = 0; i < scores.length; i++) {
    copyScores[i] = scores[i];
}
for (int i = 0; i < copyScores.length; i++) {
    System.out.println(copyScores[i] + ", ");
}
```

수행준거

- 1.1 응용소프트웨어 개발을 위하여 프로그램 설계서를 확인할 수 있다.
- 1.2 구조적 프로그래밍 언어를 활용하여 애플리케이션을 작성할 수 있다.
- 1.3 작성된 애플리케이션의 오류를 식별하고 수정할 수 있다.
- 2.1 응용소프트웨어 개발을 위하여 프로그램 설계서를 확인할 수 있다.
- 2.2 객체지향 언어를 활용하여 애플리케이션을 작성할 수 있다.
- 2.3 작성된 애플리케이션의 오류를 식별하고 수정할 수 있다.
- 3.1 응용소프트웨어 개발을 위하여 프로그램 설계서를 확인할 수 있다.
- 3.2 스크립트 언어를 활용하여 애플리케이션을 작성할 수 있다.
- 3.3 작성된 애플리케이션의 오류를 식별하고 수정할 수 있다.