**변수와 자료형**

|  |  |
| --- | --- |
| 1 | 변수에 대한 설명 중 틀린 것은?  (1) 변수의 첫 문자는 숫자로 시작할 수 없다.  (2) 변수의 첫 문자는 원칙상 특수문자로 시작할 수 없으나 예외적으로 \_(언더바) 또는 $ 는 사용이 가능하다.  (3) 예약어는 변수명으로 사용할 수 없다.  (4) ~~변수명을 대문자로 사용할 경우 에러가 발생한다.~~  (5) 자바 언어에서 클래스명의 첫 철자는 대문자로 함이 관례이다. |
| 2 | 다음 설명 중 틀린 것은?  (1) Java에서 사용 가능한 기본적 데이터의 종류는 문자,숫자,논리값이다.  (2) 숫자의 경우 세부적으로 정수와 실수로 분류된다  (3) 자료형을 나누는 이유는 메모리에 생성되는 데이터의 크기를 결정짓기 위함이다.  (4)문자형 데이터를 메모리에 올릴때는 char 형을 사용하고, 홑따옴표를 사용해서 데이터를 감싸야 한다.  (5)~~논리값은 boolean 으로 선언하며, 다른 언어에서와 마찬가지로 true, false 대신 1과 0으로 대입할 수 도 있다.~~  **C 계열만 가능** |
| 3 | 다음 설명 중 틀린 것은?  (1) 원화의 최소 단위가 1원이지만, 실생활 에서는 보다 효율적인 경제 활동을 위해 1원을 100개 모아놓은 100원을 기본단위로 사용하고 있다.  (2) ~~컴퓨터 분야에서 데이터의 최소 단위는 byte이다.~~  Byte는 기본단위  (3) 데이터의 최소단위는 1bit 이고 1bit에 저장할 수 있는 데이터의 종류는 0과 1 이라는 유형만을 저장할 수 있다.  (4) 1bit 로는 다양한 처리가 불가능하므로 1bit 를 8개 묶어놓은 단위인 byte 가 기본 단위로써 사용된다. |
| 4 | 다음 빈칸을 채우세요   |  |  |  | | --- | --- | --- | | **자료형** | **byte 수** | **데이터 표현 범위** | | byte | 1 | -128 ~128 | | short | 2 |  | | char | 2 |  | | int | 4 |  | | long | 8 |  | | float | 4 |  | | double | 8 |  | | boolean | 1 |  | |
| 5 | 다음 설명 중 틀린 것을 고르면?  (1) short a = 3; 이라는 코드를 작성할 경우 메모리에 확보되는 데이터 용량은 2byte 이다.  (2) ~~byte 자료형으로 표현할 수 있는 데이터의 경우의 수는 2의 8승인 256 가지인데, sun에서는 이 256가지 종류의 숫자를 -255 부터 0까지로 정했다.~~  (3) 자바언어는 한글 등 다국어를 지원하므로 아래와 같이 char형 변수 a에 한글인 ‘가’ 등을 담을 수 있다.  **char a = ‘가’**  (4) ~~char형이 메모리에서 차지하는 용량은 1byte 이다.~~  (5) 아래의 코드는 에러가 날 것이다.  char k=true;  (6) ~~아래의 코드는 에러가 날 것이다.~~  ~~char c=65;~~  (7) ~~아래의 코드는 에러가 나지 않을 것이다.~~  ~~char y = -10;~~ |
| 6 | 다음은 무엇에 대한 설명인가?  컴퓨터는 1과 0만을 이해하므로 , 프로그래밍 언어에서 사용되는 문자,숫자,논리값은 사실상 0과 1으로 변환되  어질 대상이 된다. 하지만 숫자나 논리값은 0과 1로 변환이 가능하나 문자는 직접적으로 0과 1로 변환하는 것은  불가능 하므로, 키보드 자판에 존재하는 모든 영문자와 특수문자에 대해 1:1로 대응하는 정수를 미국의 표준협회인 ANSI가 제시했는데, 이 코드 체계를 무엇이라 하는가?  ( ASCII 코드 ) |
| 7 | 다음 설명 중 틀린 것은?  (1) ASCII코드는 영미권 문자인 알파벳을 위주로 한 코드 체계이므로, 영문자판과 1:1 매칭되어질 데이터의 개수가  256가지 이하이므로, 1byte 기반 코드체계이다.  (2) 한글, 중국어 ,아랍어 등 비영어권 문자를 1byte 용량으로 표현하기엔 부족하므로 ASCII코드보다 넓은 범위의 문자 코드 체계로 확장시킨 코드표를 유니코드라 한다.  (3) 유니코드 기반의 언어는 문자를 메모리에 올릴때 2 byte 용량이 확보되므로, 전세계 모든 문자를 처리할 수 있다.  (4) ~~자바언어는~~ **~~char c=’대’~~** ~~와 같이 프로그램 작성시 한글 문자 사용이 가능하므로 유니코드를 지원하는 언어로 볼 수~~  ~~있으며 , 이 char 형에 의해 메모리에 확보되는 용량은 1byte로 볼 수 있다.~~ |
| 8 | 문자 자료형인 char에 대한 설명 중 맞는 것은?  (1) ~~char a=’A’ 가 실행될 경우 메모리에 실제 ‘A’라는 문자 자체가 올라간다~~  (2) char x=65; 를 입력하면 컴파일러는 아스키코드에서 65에 매칭되는 정수를 찾는다.  (3) ~~char c=-3 은 정상적으로 컴파일 된다.~~  (4) 컴퓨터의 메모리에는 문자 자체가 올라갈 수 없으므로 사실상 자바 언에서 사용되는 char형은 숫자 자료형  으로 봐도 무관하다. |
| 9 | 괄호안을 채우세요.  자바는 유니코드 기반이므로, char 형으로 선언된 데이터는 메모리에 ( 2 )byte 를 차지하게 된다.  따라서 유니코드가 표현할 수 있는 문자의 개수는 ( 2^16 ) 가지 이므로 현존하는 전세계 문자를 모두 표현할 수 있다. |
| 10 | 다음 설명 중 맞는 것은?  **class MyClass{**  **public static void main(String[] args){**  **int a=7; (가)**  **int a=3; (나)**  **{**  **int a=3; (다)**  **}**  **{**  **a=5; (라)**  **}**  **System.out.println(a); (마)**  **}**  **}**  (1) (나) 는 중복된 변수 선언으로서 컴파일 에러가 발생한다. O  (2) ~~(다)의 경우 (가) 와 중복된 선언이기는 하지만 지역변수 영역안에 선언하였으므로, (가)와 충돌되지 않기 때문에 컴파일 에러는 발생하지 않는다.~~  (3) ~~만일 (나)의 코드가 없었다면, (라)에서 컴파일에러가 발생할 것이다.~~ **다**  (4) ~~(나)의 코드가 없었다고 가정한다면 (마)의 출력결과는 7이다.~~  **5** |