[[ 기초다지기 ]]

1. 정수형 byte 타입의 범위는?

(1) 27 ~ 27-1 (2) 0 ~ 28 (3) -28 ~ 28-1

(4) -27 ~ 27-1 (5) -27-1 ~ 27

2. 다음 중에서 잘못된 것은?

(1) int i = 12345678; (2) float f = 3.5;

(3) double d = 12345678.0; (4) String s = “”;

3. 다음 프로그램의 결과는?

|  |
| --- |
| class Test {  public static void main ( String [] args ) {  byte a = 64;  byte b = 64;  byte result = a + b;  System.out.println(“result = “ + result );  }  } |

(1) 127 (2) 128 (3) -128 (4) 컴파일 에러

4. 다음 프로그램의 결과는?

|  |
| --- |
| class Test {  public static void main ( String [] args ) {  byte b = 36;  int i = ( int ) b;  System.out.println( “b = “ + b );  System.out.println( “i = “ + i );  }  } |

5. 다음 프로그램의 결과는?

|  |
| --- |
| class Test {  public static void main ( String [] args ) {  int i = 360;  byte b = ( byte ) i;  System.out.println( “i = “ + i );  System.out.println( “b = “ + b );  }  } |

6. 다음 중 형변환을 생략이 가능한 것은?

byte b = 127;

char ch = ‘글’;

int i = 20000000;

long l = 1L;

(1) b = ( byte ) i;

(2) i = ( int ) ch;

(3) int var = ( int ) b;

(4) float f = (float) l;

(5) l = (long) i;

답:

1. (4)

byte형은 1byte로 총 8bit 중에서 부호비트 1를 포함하기에 숫자를 표현하는 비트는 7비트이다. 이에 표현하는 숫자의 범위는 -27에서 +27-1까지 이다. 0이 양수측 값으로 인식되기에 양수를 표현하는 범위에서 1를 뺀 범위이다.

2. (2)

자바의 기본 실수형은 double으로 float형으로 실수를 표현할 때는 실수뒤에 f나 F를 붙여야 한다. 즉, float f = 3.5f; 나 float f = 3.5F; 여야 한다.

7. (4)

자바에서 산술연산자 + 연산한 후의 결과는 int형이기에 결과값을 byte에 저장하고자 한다면 형변환이 필요하다. byte result = (byte) ( a + c ); 라고 byte 형변환을 해야 한다. 형변환을 한 후 결과를 확인하면 -128이 된다. 이는 byte형은 -128~+127까지의 숫자를 저장할 수 있는데 64+64=128로 byte형이 표현하는 범위를 벗어나기에 정확한 값이 출력되지 않는다.

8. (3)

main() 메소드 안에 있는 지역변수임으로 초기화 작업이 필요하다.

9. true , true

a == b는 두 String 변수의 레퍼런스(주소)값을 비교하는데, String 클래스를 new를 이용하기 않고 기본 데이터 타입인 것처럼 객체를 생성하면 자바 컴파일러는 동일한 문자열에 대해 새롭게 메모리를 확보하기 않고 기존의 a의 주소값을 b에 복사하기에 a == b는 true이다. String 클래스에서 문자열이 동일한 지 확인하는 메소드로 equals() 메소드가 있는데, a와 b 모두 “hello”의 값을 가지고 있기에 a.equals(b) 역시 true 이다.

10. b = 36

i = 36

11. i = 360

b = 104

byte형 변수에는 -27~+27-1의 숫자를 표현할 수 있는데, 이는 -128~+127까지만 표현이 가능하다. int 형 변수에 저장된 360을 byte형으로 형변환하면서 값을 표현할 수 있는 범위가 넘어가 값이 짤리는 현상(overflow)이 발생하여 다른 값으로 저장되는 것이다.

int형 변수 i 에 360이 저장된 메모리 구조

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

이 4바이트 int형을 1바이트 byte 형으로 변환하면서 하위 1바이트(8비트)만 저장된다.

byte형 변수 b에 메모리 구조

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

이 변수의 값을 읽으면 104가 되는 것이다.

12. (2), (3), (4)

범위가 큰 변수에 형변환하여 지정할 때는 형변환을 생략이 가능하다.

값을 표현하는 범위는 byte < short < char < int < long < float < double 순이다.