**[ 2 ] 자바에서의 변수**

목표 : 변수에 대한 개념들을 알고 실제 자바에서 변수를 쓰는 규칙으로 사용할 수 있다.

자바에서 활용되는 여러가지 데이터 형태를 알고 적절하게 사용할 수 있다.

기본 자료형과 참조 자료형(reference variable)에 대하여 개념을 안다.

자바에서 활용되는 메모리 구조를 알고, 활용한다.

1. 변수(variable)란?
   1. 프로그램 작업을 처리하기 위해 하나의 값을 저장할 수 있는 메모리 공간을 말한다
   2. 임의의 메모리 공간에 이름을 붙여 관리하는 것이다.
   3. 자바에서는 다양한 타입을 저장할 수 없고, 한가지 타입만 값으로 저장될 수 있다.
   4. 식별자(identifier)라고도 한다 : 자바코드에서 변수로 입력시킨 이름을 의미한다.
   5. 변수 선언 방법은 다음과 같다.

|  |
| --- |
| int i = 10; //선언과 할당  데이터 타입 변수명 할당연산자 변수내용물  (주머니성질) (주머니이름) (주머니내용물)  int j; // 선언  j = 10; // 값할당 |

(ex) String greeting = “안녕하세요”;

**public** **class** VarEx0 {

**public** **static** **void** main(String[] args){

**int** age = 23; // 우항의 23 값을 좌항의 age라는 곳에 담음

String name = "설현";

System.***out***.println("안녕하세요? " + age +"살 " + name+"씨");

System.***out***.println("안녕하세요? " + age +"살 " + name+"씨");

System.***out***.println("안녕하세요? " + age +"살 " + name+"씨");

System.***out***.println("안녕하세요? " + age +"살 " + name+"씨");

System.***out***.println("안녕하세요? " + age +"살 " + name+"씨");

}

}

1. 기본 자료형

|  |  |  |  |
| --- | --- | --- | --- |
| **자료형** | **키워드** | **크기** | **표현범위** |
| 논리형 | boolean | 1 byte | true, false |
| 문자형 | char | 2 byte | 0~65,535 |
| 정수형 | byte | 1 byte | -128 ~ 127 (-27~ 27-1) |
| short | 2 byte | -32,768 ~ 32,767 (-215~ 215-1) |
| **int** | 4 byte | -2,147,483,648 ~ 2,147,483,647 |
| long | 8 byte | -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 |
| 실수형 | float | 4 byte | -3.4E38 ~ +3.4E38 |
| **double** | 8 byte | 1.7E308 ~ + 1.7E308 |

**package** com.tj.ex;

// 클래스명은 대문자로시작하여 알파벳과 숫자, \_를 쓸 수 있다.

// 변수명은 소문자로 시작하여 알파벳과 숫자, \_를 쓸 수 있다.

**public** **class** Ex02 {

**public** **static** **void** main(String[] args) {

// 프로그램 작업을 처리하기 위해 하나의 값을 저장하는 주머니 : 변수

// 변수 선언 문법 : 데이터타입 변수명 =(할당연산자) 데이터값

**int** i = 10; // 4byte짜리 i라는 이름의 주머니에 10을 넣는다.

**byte** j = 20; // 초기화

**double** h = 0.1;

i=5; // 할당

**char** c1 = 'A';

// 2byte짜리 c1이라는 이름의 주머니에 'A'코드값(65:1000 0001)을 넣는다

**char** c2 = 'B';

**char** c3 = '^';

// "\t"는 탭. "\n"은 다음줄 맨 앞으로

System.***out***.println(i+"\t"+j+"\t"+h);//개행포함

System.***out***.print(c1+"\t"+c2+"\t"+c3+"\n");

System.***out***.printf("%d\t %d \t %d\n",(**int**)c1,(**int**)c2,(**int**)c3);

// %c-문자. %s-문자열. %d-10진수. %x-16진수 %f-실수출력

}

}

**public** **class** Ex3 {

**public** **static** **void** main(String[] args) {

// 변수 선언 위치 : 제약 없음. 어디에서나 변수의 선언이 가능

System.***out***.println("프로그램 시작");

**int** num1 = 10;

// 변수 선언 후 사용할 수 있다.

System.***out***.printf("num1 = %d\n",num1);

// num2선언전이므로 사용불가

//System.out.printf("num2 = %d\n",num2);

**int** num2 = 20;

// 모든 변수는 초기화 이후에만 사용가능!!

// 초기화 : 변수의 선언 이후 값을 할당하는 것.

// 변수는 생성과 동시에 쓰레기값(사용할 수 없는 임의의 값)을 가지므로 값을 할당해야만 사용할 수 있다.

**int** num3;

//System.out.printf("num3 = %d\n", num3);

num3 = 30;

System.***out***.printf("num3 = %d\n", num3);

}

}

**public** **class** Ex4 {

**public** **static** **void** main(String[] args) {

// 변수의 선언 방법

// 1. 자료형 변수명;

// 변수의 선언과 값의 대입을 분리하는 방법

**int** num1;

num1 = 10;

// 2. 자료형 변수명 = 값;

// 변수의 선언과 동시에 값을 대입하는 방법

**int** num2 = 20;

// 3. 자료형 변수명1, 변수명2;

// 동일한 자료형 타입의 변수를 다수개 선언하는 방법

**int** num3, num4;

// 4. 자료형 변수명1 = 값1, 변수명2 = 값2;

// 동일한 자료형 타입의 변수를 다수개 선언하면서 값을 대입하는 방법

**int** num5 = 50, num6 = 60;

**int** num7, num8 = 80, num9;

System.***out***.println("num1="+num1+"\t num2="+num2);

System.***out***.println("num3="+num3+"\t num4="+num4);

System.***out***.println("num5="+num5+"\t num6="+num6);

}

}

**public** **class** Ex5 {

**public** **static** **void** main(String[] args){

**char** c = '씨';

**long** i = 2200000000;

**double** d = 3.14; // **double** d = 3.14d;

**boolean** b = **true**;

System.***out***.println("C는 "+c);

System.***out***.println("i는 "+i);

System.***out***.println("d는 "+d);

System.***out***.println("b는 "+b);

}

}

**public** **class** Ex6 {

**public** **static** **void** main(String[] args) {

// float 타입의 선언과 초기화 방법

**float** f;

// 실수의 기본형은 double 타입이므로, 10.1 은 double 타입의 값이 됩니다.

// float 자료형은 4 Byte 의 크기를 갖기 때문에 8 Byte 크기의

// double 형을 저장할 수 없습니다.

// f = 10.1;

// 실수의 기본형은 double 타입이므로, float 타입의 실수를 사용하기 위해

// 값f / 값F 형식을 사용합니다.

f = 10.1f;

f = 10.1F;

System.***out***.printf("f = %.1f\n", f);

// 형변환 예제

// (변환할 자료형)값;

f=(**float**)10.0000017;

System.***out***.println("f="+f);

**double** d = 10.0000017;

System.***out***.println("d="+d);

System.***out***.printf("f = %.2f\n", f);

System.***out***.printf("d= %.2f\n", d);

**if**(f==d)

System.***out***.println("같다");

**else**

System.***out***.println("다르다");

}

}

**public** **class** Ex6 {

**public** **static** **void** main(String[] args) {

**int** n1 = 97;

**int** n2 = 79;

**int** n3 = 85;

// 위의 n1, n2, n3 은 각 과목의 성적을 의미합니다.

// 성적 점수를 활용해 총점과 평균을 출력하세요.

**int** tot; // 총점 저장 변수

**int** avg; // 평균 저장 변수

tot = n1 + n2 + n3;

avg = tot / 3;

System.***out***.printf("총점 %d 점, 평균 %d 점 입니다.\n", tot, avg);

}

}

**public** **class** Ex6 {

**public** **static** **void** main(String[] args) {

**int** n1 = 99;

**int** n2 = 79;

**int** n3 = 85;

// 위의 n1, n2, n3 은 각 과목의 성적을 의미합니다.

// 성적 점수를 활용해 총점과 평균을 출력하세요.

**int** tot; // 총점 저장 변수

**double** avg; // 평균 저장 변수

tot = n1 + n2 + n3;

avg = tot / 3.0D;

System.***out***.printf("총점 %d 점, 평균 %5.2f 점 입니다.\n", tot, avg);

}

}

**public** **class** Ex6 {

**public** **static** **void** main(String[] args) {

**int** n1 = 99;

**int** n2 = 79;

**int** n3 = 85;

// 위의 n1, n2, n3 은 각 과목의 성적을 의미합니다.

// 성적 점수를 활용해 총점과 평균을 출력하세요.

**int** tot; // 총점 저장 변수

**float** avg; // 평균 저장 변수. 평균을 소수점계산을 위해 float타입으로

tot = n1 + n2 + n3;

avg = tot / 3.0f;

System.***out***.printf("총점 %d 점, 평균 %5.2f 점 입니다.\n", tot, avg);

}

}

1. 기본 데이터 type과 참조 데이터 type의 이해
   1. 기본 데이터 type (primitive 기본 자료형)

메모리에 있는 실제값 = 변수 데이터 값

Java 언어에 이미 존재하고 있는 데이터 타입. 주로 간단한 데이터들(예:int double, boolean, char 등..)

ex. int i = 10;

10

i

* 1. 참조 데이터 type (Object, 객체 자료형)

메모리에 있는 실제값(stack영역) = 변수가 저장된 주소(실제 변수 데이터는 heap 영역)

여러 가지 데이터들이 모여 있는 복잡한 데이터로 기본 자료형에 비해 크기가 크다(예:String, System 등의 앞으로 나올 모든 객체, 배열 등등)

ex. String greeting = “Good Morning”;

“Good Morning”

greeting

1. 형변환
   1. 묵시적 형변환 : 작은 주머니의 데이터를 큰 주머니로 옮기는 것

**public** **class** VarEx3 {

**public** **static** **void** main(String[] args){

**int** i1 = 10;

**long** l1 = 220000000L;

**double** d1 = i1; // 묵시적형변환

System.***out***.println("double형 d1의 데이터는 "+d1);

**double** d2 = 10.91d;

**int** i2 = (**int**)d2;

System.***out***.println("int형 i2의 데이터는 "+i2);

}

}

* 1. 명시적 형변환 : 큰 주머니의 데이터를 작은 주머니로 옮기는 것. 데이터 손실이 있을 수 있으며 코드에 명시해 주어야 한다

**double** d2 = 10;

**int** i2 = (**int**)d2; // 명시적 형변환

System.***out***.println("int형 i2의 데이터는 "+i2);