**[ 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** VarEx01 {

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

**int** age = 27; // 우항의 27 정수값을 좌항의 age라는 곳에 담음

String fullName = "제니"; // String 문자열 : 여러글자

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

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

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

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

System.***out***.println("안녕하세요? "+age+"살 "+fullName+"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.lec.ex;

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

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

**public** **class** VarEx02 {

**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 = 10.15;

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 다음 줄 맨 앞으로

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

System.***out***.print("변수c1\t변수c2\t변수c3\n");

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

System.***out***.printf("h=%5.1f\n",h);

System.***out***.println("프로그램 끝");

}

}

**public** **class** VarEx03 {

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

// 1. 자료형 변수명; - 변수 선언과 할당을 분리

**int** num1; // 변수 선언

num1 = 10; // 할당

// 2. 자료형 변수명 = 값; - 변수 선언과 동시에 초기화(할당)

// System.out.printf("num1 = %d, num2=%d\n", num1, num2);

**int** num2 = 20;

System.***out***.printf("num1 = %d, num2=%d\n", num1, num2);

// 3. 자료형 변수1, 변수2,...; - 동일 자료형의 변수를 다수 선언

**int** num3, num4;

num3 = 30; num4=40;

System.***out***.printf("num3 = %d, num4=%d\n", num3, num4);

// 4. 자료형 변수1=값1, 변수2=값2,...; -- 동일자료형 변수를 다수 선언과 초기화

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

}

}

**public** **class** VarEx04 {

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

**char** c = '씨';

**int** i = 20;

**long** l = 2200000000L;

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

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

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

**float** f = 3.14159265359F;

**double** d = 3.14159265359;

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

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

**if**(f==d) {

System.***out***.println("f와 d는 같다");

}**else** {

System.***out***.println("f와 d는 다르다");

}

f = 10.1F;

d = 10.1;

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

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

**if**(f == d) {

System.***out***.println("f와 d값은 같다");

}**else** {

System.***out***.println("f와 d값은 다르다");

}

// 정수(4byte)랑 실수(8byte)랑 연산한결과 : i+d(8byte)

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

// 정수(4byte)랑 정수(4byte)랑 연산한 결과 : 4byte

System.***out***.println("i / 7 = " + (i/7) );

}

}

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

**public** **class** VarEx07 {

**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 = 10.1;

**int** i2 = (**int**)d2; // 명시적 형변환

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

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”;

1000

“Good Morning”

greeting 1000번지

// 기본 데이터 타입(기초데이터타입;소문자로시작) vs 참조 데이터 타입(객체데이터타입)

**public** **class** VarEx05 {

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

// 기본 데이터 타입

**int** i;

i = 10;

// 참조데이터타입(객체데이터타입)

String name;

name = "홍길동";

name ="김수한무거북이";

}

}

실습 예제 : 국어, 영어, 수학 점수를 변수에 할당하고 각 점수를 출력하고 총점을 출력하는 프로그램을 구현 하시오(com.lec.quiz패키지내에 Example.java을 구현한다).