변수(variable)

변수란 ?

수학에서 ‘변수’를 ‘변하는 수’라고 정의하지만 프로그래밍언어에서는 변수를 값을 저장할 수 있는 메모리상의 공간을 의미

이 공간에 저장된 값은 변경될 수 있기 때문에 ‘변수’라는 수학용어와 상통하는 면이 있어서 이렇게 이름을 붙임.

변수 초기화란 ?

변수를 사용하기 전에 처음으로 값을 저장하는 것

변수의 명명규칙.

‘변수의 이름’ 처럼 프로그래밍에서 사용하는 모든 이름을 ‘식별자’라고 하며. 식별자는 같은 영역 내에서 서로 구분될 수 있어야 한다.

* 규칙 (필수)

1. 대소문자가 구분되며 길이에 제한이 없다.
2. 예약어(자바의 경우 총 53개)를 사용해서는 안 된다.
3. 숫자로 시작해서는 안 된다.
4. 특수문자는 ‘\_’와 ‘$’만을 허용한다.

* 규칙 (프로그래머의 약속)

1. 클래스 이름의 처 글자는 항상 대문자로 한다.
2. 여러 단어로 이루어진 이름은 단어의 첫 글자를 대문자로 한다.
3. 상수의 이름은 모두 대문자로 한다. 여러 단어로 이루어진 경우 ‘\_’ 로 구분한다.

Data types

* Type
* Size (Peimitive data types)
* Range (Reference data types)
* Properties

Primitive datatypes

character

boolean

Number  
(whole number)

Number  
(real number)

char

boolean

byte

float

short

double

int

long

\*\* 자바는 모두 Signd 데이터 타입

(unsigned type 없음) => 걍 –(마이너스) 쓰면 됨 ex) int a = -5;

변수 default value

byte : 0

short : 0

int : 0

long : 0

float : 0.0

double : 0.0

char : ‘ ‘ (space)

boolean : false

상수와 리터럴

‘상수(constant)’는 변수와 마찬가지로 ‘값을 저장할 수 있는 공간’ 이지만, 변수와 달리 한번 값을 저장하면 다른 값으로 변경할 수 없다

상수를 선언하는 방법은 변수와 동일하며, 단지 변수의 타입 앞에 키워드 ‘final’을 붙여주기만 하면 됨

final int MAX\_SPEED = 10;

그리고 상수는 반드시 선언과 동시에 초기화해야 하며, 그 후 부터는 상수의 값을 변경하는 것이 허용되지 않는다.

리터럴(literal)

프로그래밍에서는 상수를 ‘값을 한 번 저장하면 변경할 수 없는 저장공간’으로 정의하였기 때문에 이와 구분하기 위해 상수를 다른 이름으로 불러야만 했다. 그래서 상수 대신 리터럴이라는 용어를 사용한다.

Ex)

int year = 2015;

final int MAX\_VALUE = 100;

리터럴의 타입과 접미사

정수형과 실수형에는 여러 타입이 존재하므로, 리터럴에 접미사를 붙여서 타입을 구분한다.

정수형의 경우, long타입의 리터럴에 접미사 ‘I’ 또는 ‘L’ 을 붙이고, 접미사가 없으면 int 타입의

리터럴이다. Byte와 short타입의 리터럴은 별도로 존재하지 않으며 byte와 short타입의 변수에 값값 저장할 때는 int타입의 리터럴을 사용한다.

10진수 외에도 2,8,16진수로 표현된 리터럴을 변수에 저장할 수 있다

Ex)

int octNum = 010; // 8진수 10, 10진수로는 8

int hexNum = 0x10; // 16진수로 10, 10진수로는 16

int binNum = 0b10; // 2진수 10, 10진수로는 2

형변환(casting).

같은 타입뿐만 아니라 서로 다른 타입간의 연산을 수행해야 하는 경우도 있다.

이럴때, 연산을 수행하기 전에 타입을 일치시켜야 한다.

형변환이란, 변수 또는 상수의 타입을 다른 타입으로 변환하는 것

형변환 방법 : (타입)피연산자

Ex)

double d = 85.4;

int score = (int)d;

System.out.println(“score = ” + score);

System.out.println(“d = ” + d);

자동형변환.

float f = 1234; // 형변환의 생략. float f = (float)1234; 와 같음.

자동형변환은 기존의 값을 최대한 보존할 수 있는 타입으로 자동 형변환한다.

1. Boolean 을 제외한 나머지 7개의 기본형은 서로 형변환이 가능하다.
2. 기본형과 참조형은 서로 형변환할 수 없다.
3. 서로 다른 타입의 변수간의 연산은 형변환을 하는 것이 원칙이지만, 값의 범위가 작은 타입에서 큰 타입으로 형변환은 생략할 수 있다.