02

02-1. 변수와 타입

Java Basic Syntax

소프트웨어 언어 학습 흐름

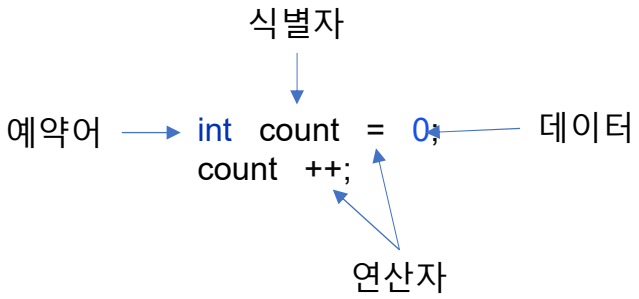
- 모든 소프트웨어 언어는 작성규칙이 있습니다. 이 작성규칙을 그 언어의 문법이라고 합니다.
- 소프트웨어 문법을 익히고 그 문법에 맞추어 프로그램을 작성해 주면 됩니다.

코드는 예약어, 식별자, 연산자, 데이터의 조합이다.

- 예약어는 소프트웨어 언어에서 지정한 단어
- 식별자는 개발자가 구분을 위해 지정한 단어
- 연산자는 특별한 연산을 위해 지정된 기호 혹은 단어
- 데이터는 값

소프트웨어 언어 학습 흐름

- 소프트웨어 언어 문법을 학습한다는 것은 그 언어에서 제공하는 예약어가 무엇이 있으며 식별자 정의 규칙은 어떻게 되고, 연산자는 무엇이 있는지등을 학습하는 것입니다.



소프트웨어 언어 학습 흐름

코드는 선언구문과 실행구문으로 구분된다.

- 프로그래밍 코드는 선언구문과 실행구문으로 구분할 수 있습니다.

- 선언 구문이라고 하면 "무언가 이런 것이 있다" 라고 선언하기 위해 작성되는 구문입니다.
- 변수, 함수, 클래스를 선언하기 위해서 작성되는 구문이며 흔히들 "변수를 선언해서, 함수를 선언해서, 클래스를 선언해서" 라는 표현을 많이 합니다.
- 선언구문에는 각 선언한 것을 식별하기 위해서 식별자를 지정해 줍니다.
- 변수를 선언하면서 그 변수 명을 지정하거나 함수를 선언하면서 그 함수명을 지정하게 됩니다.

소프트웨어 언어 학습 흐름

코드는 선언구문과 실행구문으로 구분된다.

- 실행구문은 이미 선언되어 있는 변수, 함수, 클래스를 이용하는 구문을 의미합니다.
- 선언된 변수에 값을 판단하거나 변경하거나, 선언된 함수를 호출하거나 아니면 선언된 클래스를 생성하는 등 무언가 선언된 것 가지고 행위가 이루어지는 구문을 실행구문이라고 합니다.

```
class User { }

public class Exam1 {

   //선언
      int sum = 0;

   //선언
      void calSum() {
      //실행구문
            for(int i = 1; 1 <=10; i++){
         sum += i;
      }
   }
}
```

소프트웨어 언어 학습 흐름

소프트웨어 언어의 문법을 먼저 학습하고 플랫폼에서 제공하는 API 를 학습한다.

- 문법을 학습한 다음 플랫폼에서 제공되는 API 를 학습해야 합니다.
- API 라고 하면 프로그램 코드에서 이용할 수 있는 변수, 함수, 클래스등을 의미하며 이미 만들어져 제공되는 것들입니다.

소프트웨어 언어 학습 흐름

애플리케이션의 궁국적인 목적은 데이터이다. 이 데이터를 이용하기 위해 변수, 타입등이 학습되어야 한다.

- 데이터는 애플리케이션에서 이용하는 값이라고 생각하면 되는데 이 값이 애플리케이션이 실행되는 동안 컴퓨터 메모리에 저장되어 있어야 합니다.
- 메모리에 저장된 데이터를 쉽게 식별해서 이용하기 위해서 변수를 선언하고 그 변수로 데이터를 이용합니다.

- 변수로 데이터가 이용됨으로 그 데이터가 어떤 형식의 데이터인지를 구분하기 위한 타입 이야기가 따라 나옵니다.
- 타입이란 변수에 저장된 데이터가 숫자 타입인지, 문자타입인지를 구분하는 문법이라고 보시면 됩니다.

소프트웨어 언어 학습 흐름

데이터를 적절하게 핸들링하기 위한 연산자, 제어문에 대한 학습이 필요하다.

- 애플리케이션이 실행되면서 각 데이터를 우리가 원하는 대로 조작해 주어야 합니다.
- 단순하게 사칙 연산을 하거나 어느 데이터가 큰지를 판단하는 등 다양한 작업이 진행되어야 합니다.
- 이를 위해서는 연산자가 필요하며 자바에서 어떤 연산자를 제공하는지 학습이 되어야 합니다.

- 어떤 코드 부분이 반복적으로 실행되거나 조건에 만족하는 경우에만 실행되게 제어를 해주어야 합니다.
- 이런 부분을 제어문이라고 하고 자바에서 제어문을 작성하는 방법에 대한 학습이 있어야 합니다.

소프트웨어 언어 학습 흐름

하나의 업무처리를 위한 여러 라인의 코드를 묶어서 이용하기 위한 함수의 학습이 필요하다.

- 하나의 업무 처리를 위해서는 여러 라인의 코드가 필요할 수도 있습니다.
- 하나의 업무를 위한 여러 라인의 코드를 묶어서 개발하는데 그 역할을 하는 것이 함수입니다.

관련된 데이터 및 함수를 묶어서 표현하기 위한 객체에 대한 학습이 필요하다.

- 프로그램 코드를 작성하다 보면 많은 변수, 함수들이 만들어지는데 이 변수, 함수들을 관련된 것끼리 다시 묶어서 개발하고 이용합니다.
- 이를 흔히 객체라고 하며 객체를 이용하는 프로그램을 흔히 객체지향 프로그램이라고 합니다.

식별자와 예약어

- 식별자(Identifier)
  - 첫 글자는 '문자', '_', '$' 중 하나로 시작해야 한다. 숫자로 시작할 수 없다.
  - 첫 글자는 소문자로 시작하는 것이 관례다.
  - 공백을 포함할 수 없다.
  - 대소문자를 구분한다.
  - 길이에 제한이 없다.
  - 예약어를 사용할 수 없다.

식별자와 예약어

- 예약어(Keyword)
  - 프로그래밍 언어에 미리 정의된 의미있는 단어
  - 예약어는 식별자로 사용하지 않음

abstract	double	int	strictfp
boolean	else	interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	package	throw
char	for	private	throws
class	goto *	protected	transient
const *	if	public	try
continue	implements	return	void
default	import	short	volatile
do	instanceof	static	while

변수(Variable)

- 값(Value)
  - 데이터(Data) 자체
  - 예: 1, 0, 20.5, true, 'a'

- 자료형(Type)
  - 데이터(Data)의 종류
  - 예: 정수(Integer), 실수(Float, Double), 문자(Character), 논리(Boolean – True/False), 문자열(String) 등
  - Java의 모든 데이터 값들은 타입을 가짐
    - 저장될 메모리와 저장 공간 결정
    - 해당 값에 적용될 연산을 결정
    - 프로그램의 에러 점검

변수(Variable)

- 변수(Variable)
  - 데이터를 저장할 메모리의 위치를 나타내는 이름
  - 변수의 선언 (Declaration)
    - 변수를 사용하기 위해서는 사용 전에 미리 선언해야 함
    - **Type Variable1, Variable2, ...;**
      - `int numberOfItems, numberOfTitles;`
      - `boolean isHoliday;`

변수(Variable)

- 초기화(Initialization)
  - 변수에 최초의 값을 할당하는 것

```
1. numberOfItems = 0; // 오른쪽 표현의 값을 왼쪽 변수에 할당
2. isHoliday = false;
```

  - 보통 변수의 선언과 동시에 초기화를 함

```
1. float averageOfScores = 65.4;
2. String companyName = "Samsung SDS";
```

- 같은 타입의 데이터만 저장할 수 있음
  - `isHoliday = 1; // COMPILE TIME ERROR`

자바의 자료형

- 기본 자료형(primitive data types)
  - 최소 단위의 자료형
  - 메서드가 없이 값만 가짐
  - int, float, double, char 등

- 참조 자료형(reference data types)
  - 여러 자료형들의 집합으로 구성된 클래스의 객체를 참조
  - 데이터와 메서드를 가짐
  - String, Vector, Integer 등

자바의 자료형 – 기본 자료형

자료형	키워드	크기	표현 범위	사용 예
논리형	boolean	1bit	true OR false(0과 1이 아니다)	boolean isFun = true;
문자형	char	2byte	0~65,535	char c = 'f';
정수형	byte	1byte	-128 ~ 127	byte b = 89;
	short	2byte	-32,768 ~ 32,767	short s = 32760;
	int	4byte	-2147483648:2147483647	int x = 59; int z = x;
	long	8byte	…	long big = 3456789;
실수형	float	4byte	-3.4E38 ~ 3.4E38	float f = 32.5f
	double	8byte	-1.7E308 ~ 1.7E308	double d = 23.34

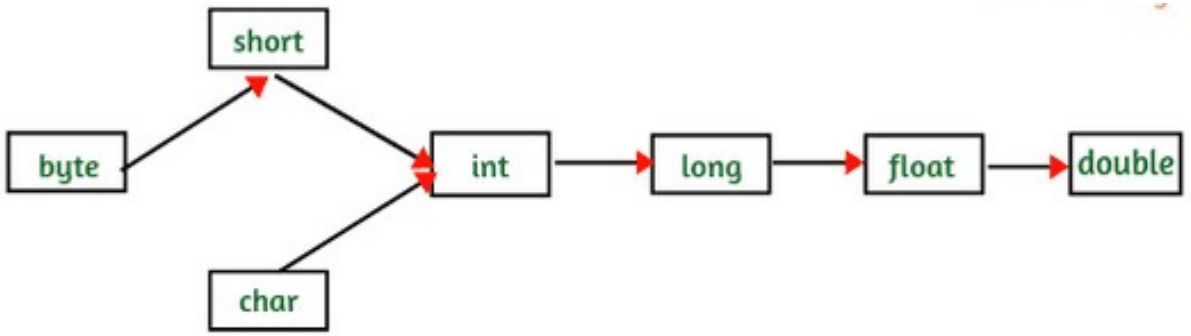
자료형 변환

- 자료형 변환(Casting)
  - 하나의 자료형을 다른 자료형으로 바꾸는 행위
    - **`boolean isHoliday = 1; // COMPILE TIME ERROR`**

- 변환 규칙
  - 암묵적 자료 유형 변환 (implicit casting): 자료의 범위가 좁은 자료형에서 넓은 자료형으로의 변환은 시스템이 자동으로 행함
  - 명시적 자료 유형 변환(explicit casting): 자료의 범위가 넓은 자료형에서 좁은 자료형으로의 변환은 시스템에서 자동으로 수행할 수 없으며 프로그래머가 강제로 변환해야 함

```
int i = 101;
long l = 101L;
l = i; // No Problem
i = l; // Compiler ERROR.
i = (int) l; // Now, it's OK.
```

자료형 변환



상수(Constant)

- 상수(Constant)
  - 변경될 수 없는 고정된 데이터 → 常數
  - 코드의 이해와 변경이 쉬움
  - 정의
    - "final" 키워드를 사용하야 선언
    - 대문자로 상수변수명을 선언 할것을 권장
      - `final double PI = 3.1416;`
      - `final int MAXIMUM_SPEED = 240;`

참고 – 강형언어와 약형언어

- 강형 언어 (Strongly-typed language)
  - 모든 변수의 Type 검사가 컴파일 시에 이루어지는 언어
  - 신뢰성, 유지보수성, 가독성이 높음
  - C, Java, Ada 등

- 약형 언어 (Loosely-typed language)
  - 정적인 Type 체계를 갖추지 않은 언어
  - 실행시에 변수가 가질 자료의 유형이 결정됨
  - Javascript

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare