Module 1 --1.2

Notebook:	Module 1_note		
Created:	7/7/2020 8:17 PM	Updated:	7/11/2020 9:54 AM
Author:	Runan Yang		
Tags:	Log		

Variable Defining

- Variable names should be defined using _, $ numbers and letters. $ is not recommenced
- Numbers could not be used as the first symbols of variables'.
- Do not use key words in Java, such as Static, Public...
- Java is case-sensitive.
- Length of variable name is not limited.
- Usually, capital letter is upper case for defining.

Variable input and output, Java API inventory

1. Scanner is used. import java.util.Scanner. And System.in represents what user inputs here.
2. Call scanner should add "Scanner sc = new Scanner(System.in)"
   - note: info imported by Scanner could only be saved as type String
3. never ever forget ; at the end of each line.

Data/variable types

- Normally, there are 8 basic types: byte, short, int,long; float, double, boolean and char.
- Noted, String is not a basic type in the 8 basics.
- And 5 types of input data.

Some bases used: binary, base 10, base 8, etc...

- Since in base 10, every digit represents 10^n (n= 0,1,2...), it is normal and natural that in binary, every digit could be treated as 2^n.
- A case study: 45 = 5 * 10^0 + 4 * 10^1; and 45 = 32 + 8 + 4 + 1 . Thus, in binary, 45 could be converted as 00101101

- Similarly, 00101101 as binary, could be converted to base 10 int.
- While, not all the numbers are non-negative integers. Thus, in binary, the first digit usually represents positive and negative. 0 represents non-negative; 1 is negative.
- A simple case (important and intermediately hard): -45 could be converted in this way
    - 45 converted to 00101101
    - all the digit to the negatives, which is 11010010
    - adding 1 to this number , 11010011

Thus, in this simple case, 45 is 00101101 in binary; while -45 is 11010011

- Similarly, the negative binary could be converted to base 10 in the procedure. For example, here is a binary 10101010
    - minus 1 at first. 10101001
    - and all digit to the negative 01010110
    - adding them together and add negative sign. 2^0 *0 + 2^1*1 + 2^2*1 + 2^3*0+ 2^4*1+2^5*0+2^6*1 = 86
    - so the answer is -86

Very important!!!

- The range a byte (8-digit in binary) could represent [-128, 127]
    - First, while the base 10 numbers are no less than 0. The first digit of byte is 0.
        - The range here in binary is 0000 0000 ~ 0111 1111
            - convert to base 10 could be 0 ~ 2^0 *1 +2^1 *1 + 2^2*1 +2^3*1 + 2^4*1 +2^5*1 + 2^6*1 = 127
        - No more than 0. the range is 0~127
    - Then, the other side. The first digit is 1. 1000 0000~ 1111 1111
        - The first step here is subtracting 1. the number goes to 0111 1111 ~ 1111 1110
        - Then, all digits go to the other side,1000 0000 ~ 0000 0001
        - Add the negative sign, -128 ~ -1

Integer type

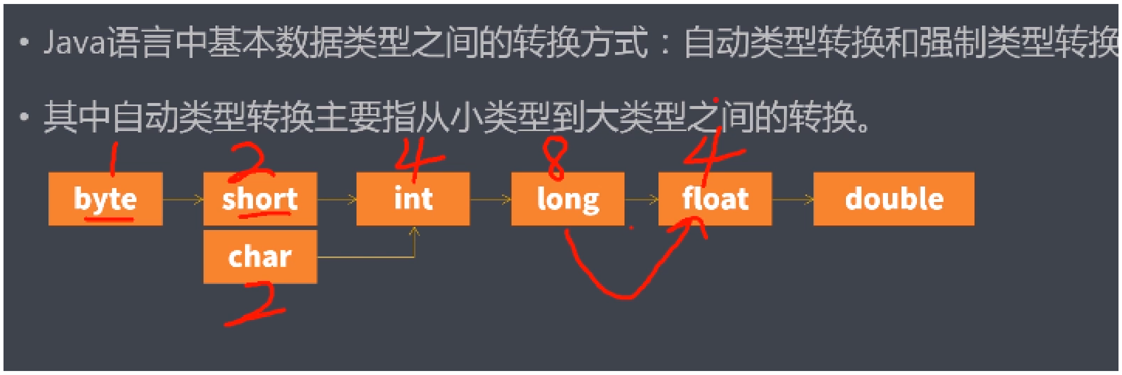
- Totally, there are 4 types of integer in java.
    - byte, 8-digit -128~ 127
    - short == 2 byte -2^15 ~ 2^15-1
    - int == 4 byte -2^31 ~ 2^31-1
    - long == 8 byte -2^63 ~ 2^63-1
- By default, in Java, javac will automatically identify the integer as type "int"

Non-integer type

- float -- 2 bytes since in java, the decimal will automatically be identified as double. Thus, a "F" or "f" should be added to the end of this input decimal.
- double -- 4 bytes

These two types might import some errors in processing. Thus, they could not be used for important financial data processing and calculations.

- boolean -- uncertain, less than 1 maybe
  - boolean type. Only two values, True and False, could be taken for a variable with type boolean.
- char -- 2 bytes
  - char is used for some symbols, such as letters, Chines characters, etc...
  - char, int, ASCII code. usually, when we define a variable char, it goes to the symbol. If we define it as a value, jdk would interpret it as the symbol.
  - basics: ASCII '\n' (line changing) -- 10 ' ' (space) ---32 '0' (number zero) -- 48 'A' (upper) -- 65 'a' (lower) -- 97



- Java语言中基本数据类型之间的转换方式：自动类型转换和强制类型转换
- 其中自动类型转换主要指从小类型到大类型之间的转换。

While doing the transfer from upper level to lower, one should use the format
new variable = (target type) source variable