Programming 1

Lesson 05 – Identifier and Data Type

Identifier

Identifier

- Identifiers are the name for the variable or methods and so on.
- In general you have to follow the following rules while giving an identifier:
  - Java is case-sensitive.
  - Right Syntax
  - Self-documenting
  - Stable style

Identifier - syntax

- You can only use the following symbols in an identifier:
  - letters,
  - digits,
  - the underscore character (_),
  - the dollar sign ($)
- An identifier must begin with a letter, underscore, or the dollar sign (but not digits)
- Mostly use letters and digits in identifiers: num1, num2, avgNum, str1, idx, etc.

Identifier: Syntax

- Quiz
- Which of the following identifiers are legal?
  - counter1
  - $Amount
  - employee Salary
  - Hello!
  - counter1
  - conversion
  - one+two
  - payRate
  - 2nd
  - first
  - class

- Answer
- Which of the following identifiers are legal?
  - T
  - T
  - F
  - F
  - T
  - T
  - F
  - T
  - F
  - T
  - F

Identifier – Self-documenting

- We write documentations and comments to make sure our code is easy to understand.
- Identifiers should be meaningful, so people can understand what you is the variable or what does the method do when they read the identifier.
- Meaning less identifiers such as: x, y, a, b, m, n should not be used.
- It will come with practice to learn how to name variables and methods appropriately.
- It’s the same as naming folders, you don’t want to name your folders as: new folder, new folder(1), new folder(2), etc….

Identifier – Self-documenting

- Do not be afraid if the identifier is too long, the most important thing is to keep your code understandable.
- Identifiers with more than one words should follow the camel-case:
  - Class: XxxxXxxxXxx
  - Variable or Method: xxxxXxxxXxxx
  - For example: classAvgScore, calcFinalScore, isUserStatusValid
- Short abbreviations are also widely used for identifiers:
  - str: string
  - num: number
  - avg: average
  - idx: index
  - calc: calculate

Identifier – Stable Style

- Each programmer may have different programming styles (preferences)
- It is OK to have style different from others
- But it is NOT OK to have a random style
- It is better to follow a fixed pattern for you to give identifiers

Identifier – Stable Style

- For me, I follow the strategy that:
  - For a class: noun
  - For a method: verb + noun
  - For a variable: adj + noun, or noun + adj
- Then the identifiers in the previous example can be re-written as:
  - strIn, strOut,
  - calcResult (or even calcSum, calcArea, calcPerimeter etc. would be better)
  - convertLetters
  - processOper


Questions?

Primitive Data Type

Primitive Data Type

- The objective of a Java program is to manipulate data.
- A Data should has a specific data type.
- Java is a static type programing language, that means it is very strict about data types. (Python or JavaScript are dynamic type programming language, which is not that strict about data types)
- In Java, putting a value of a data type into another data type variable may (or may not) create a problem.

Primitive Data Type

- In Java, there are some Primitive Data Types which is pre-defined by Java, or user defined class (can be understood as data type)
- In general, there are three groups of Primitive Data Types in Java.
- Integral: a data type that deals with integers and characters
- Floating-point: a data type that deals with decimal numbers
- Boolean: a data type that deals with logical values

Primitive Data Type - integral

- Integral data types are further classified into five:
  - char
  - byte
  - short
  - int
  - long

Primitive Data Type - integral

- All the data are stored in the memory (RAM). So it is important to declare that how much memory you would like to give for storing the data.
- Especially in the early days, the memory is relatively small, you really have to calculate it carefully.
- All the integral data type store integers, the only difference is that they have different amount of space to store a value.

Primitive Data Type - integral

Data Type	Values	Storage (in bytes)
`char`	0 to 65535 ($= 2^{16} - 1$)	2 (16 bits)
`byte`	-128 ($= -2^7$) to 127 ($= 2^7 - 1$)	1 (8 bits)
`short`	-32768 ($= -2^{15}$) to 32767 ($= 2^{15} - 1$)	2 (16 bits)
`int`	-2147483648 ($= -2^{31}$) to 2147483647 ($= 2^{31} - 1$)	4 (32 bits)
`long`	-9223372036854775808 ($= -2^{63}$) to 9223372036854775807 ($= 2^{63} - 1$)	8 (64 bits)

Primitive Data Type - integral

- int is the most frequently used data type. Most of the time, when we want to store an integral number, we use int data type.

- It support positive numbers (e.g. 5), negative numbers (e.g. -3) and 0.

Primitive Data Type - integral

- The main purpose of char data type is to represent single characters.
- A character in Java is defined with ‘ ’. (This is different from a string, which is a group of several characters, defined with “ ”).
- Examples: 'A', 'a', '0', '*', '+', '$', '&', ' '
- The char data type can only store one symbol.

Primitive Data Type - integral

- As I just said, each character has a specific representation in computer memory, but the computer can only store 0 and 1. So when we store ‘a’, it is a long string of ‘0’s and ‘1’s stored in the computer to represent it.
- Java uses the Unicode character set, you can understand it as a dictionary which defines what number should be stored for each character (English, French, Arabic, Korean, Japanese, Chinees, etc.).
- Since Unicode character set is a big “dictionary”, it contains 65536 values numbered 0 to 65535. When we store a character in the memory, the computer will check the dictionary and find the appropriate number for it, and then convert it to binary number and store it in the computer.

Primitive Data Type - integral

- Unicode character set is a big “dictionary”, in most of the time we do not need to use the entire Unicode character set, but a much smaller set names “ASCII”, which stands for American Standard Code for Information Interchange. ASCII set is a subset of Unicode character set. It covers all the most frequently used keys on the English keyboard.
- For example, in ASCII, 65 represents 'A', 43 represents '+’.
- There is no need for you to memorize the ASCII table, you can find the it online easily.
- That’s why char belongs to the integral group

Primitive Data Type - integral

- It is a good idea to remember some general pattern in the ASCII code table:
  - Numbers, upper-case letters, lower-case letters are group together, sorted.
  - Upper-case letters has smaller values than lower-case letters
  - The ASCII value of a number does not equal to the number itself, e.g.: ‘0’ is 48
  - Space is also a character, with ASC II value 32
  - The difference between an Upper-case letter and its lower-case letter is also 32


Questions?

Primitive Data Type - Boolean

- Java also support Boolean (or logical) data type
- There are only two values (both lower case) in Boolean data type
  - true
  - false
- Boolean data type is widely used in selection and looping structure.

Primitive Data Type - Floating

- Beyond Integral numbers, we also need floating numbers
- Java support two different kinds of floating number
  - float: 4 bytes, accuracy 6~7 digits after the decimal point (single accuracy)
  - double: 8 bytes, accuracy 15 digits after the decimal point (double accuracy)
- In Java, a number with decimal part is double by default.
- If you insist to use the float data type, you should add a “f” behind the number, e.g.: 3.14f. But most of the time using double is fine.

Primitive Data Type - Floating

- Attention: If you write 3, it is an int number; while if you write 3.0, it is a double number.
- For double number, you can also write
  - 3e2, which equals to 3 * 100 = 300
  - 3e-2, which equals to 3 * 0.01 = 0.03
  - When you have a very small or very large number, you can use this way to store the number.


Questions?