Head First Java: Chapter 3 Notes

August 27, 2019

Variable Types

Variables are containers that hold or reference to data. Variables in Java must be declared along with a type. There are two main categories of types:

- Primitive — these hold fundamental (simple bit patterns) values:
  - `boolean` — true or false
  - `char` — a single character. E.g. 'a', 'T', '3', '@'.
  - Numerics/Integers — these are positive or negative whole numbers and can be various in size and must be declared thusly:
    - `byte` — values from -128 to 127
    - `short` — values from -32768 to 32767
    - `int` — values from -2147483648 to 2147483647
    - `long` — values with VERY large magnitudes
  - Floating points — these are essentially decimals/fractions, two different size:
    - `float` — 32 bits in memory
    - `double` — 64 bits in memory
  - Declare primitive variables by: `int x = 4;`, `char atSign = '@';`, or without assigning values: `double y;`, `boolean isTom;`
- Object Reference — these reference to objects, few things of note:
  - They **DO NOT** hold actual objects, but rather memory address to an object.
  - The dot operator (`.`) allows access to instance variables and methods within an object the variable is referencing: `object.variable`, `object.method()`
  - Declare object reference variables by: `className objectName = new className();`

An array is a "list" or variables. The individual element of the array can be any kind of variables (primitive or object reference). For example, to declare an `int` array of length 7:

```
int[] nums;
nums = new int[7];
nums[0] = 6;
nums[1] = 19;
```

Like python lists; arrays are ordered, 0-indexed, and mutable. In Java, arrays are also objects and behave like any other objects, no matter if they contain primitives or object references. An object reference array can be declared as:

```
className[] names;
names = new className[7];
names[0] = new className();
names[1] = new className();
```