Variables and Modifiers in Java

Prof. Siddharth Shah
Department of Computer Engineering
Dharmsinh Desai University

Outline

- Variable Types
- Local Variables
- Instance Variables
- Class/Static Variables
- Modifiers
- Access modifiers
- Non-Access Modifiers
- Overloading (Method and Constructor)
- Passing and Returning Objects/References (Call-By-Value and not Call-By-Reference)

Variable Types

- There are three kinds of variables in Java –
  - Local variables
  - Instance variables
  - Class/Static variables

Instance Variables

- Instance variables are declared in a class, but outside a method, constructor or any block.
- When a space is allocated for an object in the heap, a slot for each instance variable value is created.
- Instance variables are created when an object is created with the use of the keyword 'new' and destroyed when the object is destroyed.
- Access modifiers can be given for instance variables. Normally, it is recommended to make these variables private.
- The instance variables are visible for all methods, constructors and block in the class.
- Instance variables have default values. Values can be assigned during the declaration or within the constructor.
- Instance variables can be accessed directly by calling the variable name inside the class. However, within static methods (when instance variables are given accessibility), they should be called using the fully qualified name. ObjectReference.VariableName.

Local Variables

- Local variables are declared in methods, constructors, or blocks.
- Local variables are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method, constructor, or block.
- Local variables are visible only within the declared method, constructor, or block.
- Access modifiers cannot be used for local variables.
- Local variables are implemented at stack level internally.
- There is no default value for local variables, so local variables should be declared and an initial value should be assigned before the first use.

Class Variables

- Class variables also known as static variables, are declared with the static keyword in a class, but outside a method, constructor or a block.
- There would only be one copy of each class variable per class, regardless of how many objects are created from it.
- Static variables are rarely used other than being declared as constants.
- Static variables are stored in the static/class memory.
- Static variables are created when the class is loaded and destroyed when the class is unloaded.
- Visibility is similar to instance variables. However, most static variables are declared public since they must be available for users of the class.
- Default values are same as instance variables. Values can be assigned during the declaration or within the constructor. Additionally, values can be assigned in special static initializer blocks.
- Static variables can be accessed by calling with the class name ClassName.VariableName.
- When declaring class variables as public static final, then variable names (constants) are all in upper case. If the static variables are not public and final, the naming syntax is the same as instance and local variables.

Modifiers

- There are two types of modifiers in Java – Access and Non-Access Modifiers
- Access Modifiers to set access levels for classes, variables, methods and constructors. The four access levels are
- private: Visible to the class only.
- default: Visible to the package. No modifiers are needed.
- protected: Visible to the package and all subclasses.
- public: Visible to the world.

Modifiers (Cont..)

- Non-Access Modifiers
- static modifier for creating static blocks, class (nested), methods and variables.
- final modifier for finalizing the implementations of classes, methods, and variables.
- abstract modifier for creating abstract classes and methods.
- synchronized and volatile modifiers, which are used for threads.