Classes & Objects-indepth

CCVT-2018

Ravi Tomar

Assistant Professor(Senior Scale)
Department of Analytics
School of Computer Science
University of Petroleum & Energy Studies

January 16, 2018

class declaration

class declaration can include these components:

- Modifiers such as public, private, and a number of others that you will encounter later.
- The class name, with the initial letter capitalized by convention.
- The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
- A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
- The class body, surrounded by braces, .

class variables

There are several kinds of variables:

- Member variables in a class—these are called fields.
- Variables in a method or block of code—these are called local variables.
- Variables in method declarations—these are called parameters.

Methods

method declarations have six components, in order:

- Modifiers—such as public, private, and others such as static(discuss later)
- The return type—the data type of the value returned by the method, or void if the method does not return a value.
- The method name—the rules for field names apply to method names as well, but the convention is a little different(verb).
- The parameter list in parenthesis—a comma-delimited list of input parameters, preceded by their data types, enclosed by parentheses, (). If there are no parameters, you must use empty parentheses.
- An exception list—to be discussed later.
- The method body, enclosed between braces—the method's code, including the declaration of local variables, goes here.

Overloading Methods

Constructor

A class contains constructors that are invoked to create objects from the class blueprint. Constructor declarations look like method declarations—except that they use the name of the class and have no return type.

- no args constructor.
- object class constructor.
- construtor overloading.
- You can use access modifiers in a constructor's declaration to control which other classes can call the constructor.(will discuss with inheritence)

Passing Information to a Method or a Constructor

The declaration for a method or a constructor declares the number and the type of the arguments for that method or constructor. Listed below:

- Parameter Types-Any including primitives
- Arbitrary Number of Arguments-an **ellipsis** (three dots, ...)
- Parameter Names-The name of a parameter must be unique in its scope but can be same as a feild name(resolved by this).
- Everything in Java is passed by Value, still a big difference between **Primitives** and **Objects**

Creating Object

Lifecycle is divided in four parts:

- Declaration: associate a variable name with an object type.
- Instantiation: The new keyword is a Java operator that creates the object.
- Initialization: The new operator is followed by a call to a constructor, which initializes the new object.
- Garbage Collection-Automatic

More on Object

Once created object can be used to:

- Referencing an Object's Fields
- Calling an Object's Methods
- Returning values from methods.
  > completes all the statements in the method,
  > reaches a return statement, or
  > throws an exception (covered later),
- Returning a Class or Interface-**Co-variant Return type**

this keyword

Within an instance method or a constructor, this is a reference to the current object — the object whose method or constructor is being called. You can refer to any member of the current object from within an instance method or a constructor by using this.
Also used in construtors to call other construtors.

static keyword

Can be used for:

- Class Variables.
- Class Methods.
- Define Constant along-with final keyword.
- static **initialization block.**
  >Objects can be initialized via constructors but what about static feilds?
  > Initializing Fields Problem? Simple but we recommend construtor, Why?
  >Is there any Alternative to this problem thru functions?

Conclusion

- A class declaration names the class and encloses the class body between braces.
- The class name can be preceded by modifiers.
- The class body contains fields, methods, and constructors for the class.
- A class uses fields to contain state information and uses methods to implement behavior.
- Constructors that initialize a new instance of a class use the name of the class and look like methods without a return type.
- We specify a class variable or a class method by using the static keyword in the member's declaration.
- A member that is not declared as static is implicitly an instance member.
- Class variables are shared by all instances of a class and can be accessed through the class name as well as an instance reference.
- The garbage collector automatically cleans up unused objects.
- An object is unused if the program holds no more references to it.
- We can explicitly drop a reference by setting the variable holding the reference to null.