Structure of Java Program

Thursday, August 27, 2020 10:08 PM

Import section

Package declaration section // if needed

Interface declaration & definition section // if needed

Public class class-name

{

static variable declaration; // if needed

public static void main(String s [ ])

{

body of main()

}

other static function definitions //if needed

} //end of main class

other class definitions. // if needed

Guidelines to design a class: -

- Always keep data as private.
- Always initialize data.
- Don't use too many basic types in a class.
- Use a standard form for class definition.
- Use descriptive names to the class and its members.

A simple class in Java:

```
public class demo
{
    public static void main( String s[ ] )
    {
            --------
            -------- // body of main( )
    }
}
```

Declaration of Objects

As Java is an object oriented language, any no. of classes can be defined in a program. If a class is defined, user has to create an object for that class to make use of the data. In fact it is a two step process.

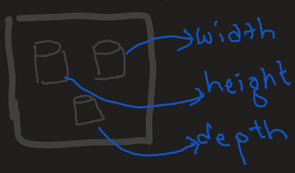
First you must declare a variable of the class type. This variable doesn't define an object. Instead, it is simply an object reference.

Second, we create a physical copy for the class and assign the address to the reference. To do this, use ‚new' keyword. The new keyword dynamically allocates memory for an object and returns the address.

Ex :

Box B; //creating a reference

B = new Box( ); //instantiating object



Example:

```
Public class demo
{
    public static void main(String s[ ])
    {
        test t = new test( ); // instance for class test is created t . display( );
    }
} // end of main class
class test
{
    int a ;
    public void display( )
    {
        a = 101;
        System.out.println( ,the value of a is <..'+a);
    }
} // end of other class.
```

Note:

1. If a program execution is over, the corresponding objects created will automatically be destroyed by the built-in garbage collector of JVM which will be invoked once the objects go beyond the scope of the programmer.
2. If we do not initialize the instance variables then they automatically assigned certain values.
    - For numeric data, a zero is assigned.
    - For objects & strings, a null is assigned.
    - For Boolean data, a false is assigned.