Module-II (Part-II)

by:

Dr. Soumya Priyadarsini Panda

Sr. Assistant Professor

Dept. of CSE

Silicon Institute of Technology, Bhubaneswar

Contents

- Interface Cont…

- Packages

Example:

```
//design an interface calculator to perform arithmetic calculations

interface calculator
          {
          int a=20,b=10;
          void add();
          void sub();
          void mul();
          void div();
          }
```

Cont…

```
class A implements calculator{
    public void add(){
            System.out.println(a+b);
            }
    public void sub(){
            System.out.println(a-b);
            }
    public void mul(){
            System.out.println(a*b);
            }
    public void div(){
            System.out.println(a/b);
            }
}
```

Cont…

```
class B{
    public static void main(String args[])
            {
            A a=new A();
            a.add();
            a.sub();
            a.mul();
            a.div();
            }
}
```

Multiple Inheritance using Interfaces

```
class MyClass implements Interface1, Interface2, Interface3
{
    // class body
}
```

Example

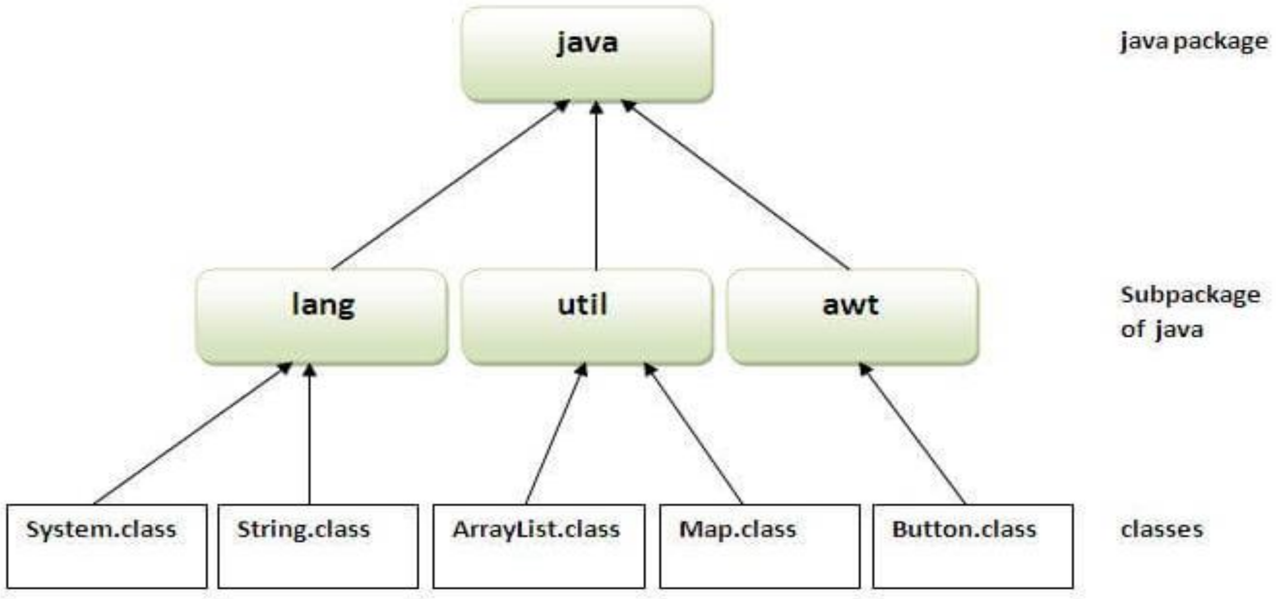
```
interface A{
    void m1();
    }
interface B {
void m2();
}
class C implements A,B{
        public void m1(){
            System.out.println(“Hello”);
            }
        public void m2(){
            System.out.println(“Hi..”);
        }
    }
```

```
public class Test
  {
  public static void main(String[] args)
      {
      C  c1 = new C();
      c1.m1();
      c1.m2();
    }
  }
```

Packages

Packages

- A java package is a group of similar types of classes, interfaces and sub-packages.



Types of Packages

- **Built-in Packages:**
  - These packages consist of a large number of classes which are a part of Java **API**.

- **User Defined Packages:**
  - These are the packages that are defined by the user.

Examples of Built-in Packages:

- **java.lang:** Contains language support classes (e.g classed which defines primitive data types, math operations).
  - This package is automatically imported.

- **java.io:** Contains classed for supporting input / output operations.

- **java.util:** Contains utility classes which implement data structures like Linked List, Dictionary and support for Date / Time operations.

- **java.applet:** Contains classes for creating Applets.

- **java.awt:** Contain classes for implementing the components for graphical user interfaces (like button, menus etc).

Creating User-defined Packages:

- These are the packages that are defined by the user.
- A package can be created by include a ***package*** command as the first statement in a Java source file.
- Any classes declared within that file will belong to the specified package.
- The package statement defines a name space in which classes are stored

Syntax: package *pkgName*;

Example: package MyPackage;

Steps:

1. Create a directory **myPackage**

2. Create a file **MyClass.java** inside the directory

3. Write the first line inside **MyClass.java** as-

    package myPackage;

    //**myPackage** is the name of the package

4. Write the methods and other details in the MyClass.java file

5. Create another file **PrintMath.java** outside the folder to use the user defined package

Example: step 3, 4

```
package myPackage;    //step-3
public class MyClass
{
    public void add(int a, int b)
    {
        System.out.println(“sum=“ +(a+b));
    }
}
```

NOTE: package name and directory name should be same

Example: step 5

```
//file outside the directory to use the package

import myPackage.MyClass;    //or import myPackage.*;
public class PrintMath
{
   public static void main(String args[])
   {
       int a=5,b=2;
       MyClass ob1 = new MyClass();
      ob1.add(a,b);
   }
}
```

Notes

- Packages act as containers for classes and other subordinate packages.

- Classes act as containers for data and code.