Packages

Lecture 26

Learning objective

After the completion of today’s session you will be able to:

- Illustrate the use of packages.
- Import packages and use it in your code.
- Select appropriate predefined packages in your code.

Packages – introduction:

- *Packages* are containers for classes. A **java package** is a group of similar types of classes, interfaces and sub-packages.
- They are used to keep the class name space compartmentalized. For example, a package allows you to create a class named **List**, which you can store in your own package without concern that it will collide with some other class named **List** stored elsewhere.
- Packages are stored in a hierarchical manner and are explicitly imported into new class definitions.
- Package in java can be categorized in two form, built-in package and user-defined package.
- There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.
- Our main discussion will be on user defined package.

Predefined Java Packages

PACKAGE NAME	EXAMPLE CLASSES	FUNCTIONALITY (PURPOSE)
java.lang	System, String, Object, Thread, Exception etc.	These classes are indispensable for every Java program. For this reason, even if this package is not imported, JVM automatically imports.
java.util	These are called as utility (service) classes and are used very frequently in coding.	
java.io	FileInputStream, FileOutputStream, FileReader, FileWriter, RandomAccessFile, BufferedReader, BufferedWriter etc.	These classes are used in all I/O operations including keyboard input.
java.net	URL, ServerSocket, Socket, DatagramPacket, DatagramSocket etc.	Useful for writing socket programming (LAN communication).

Predefined Java Packages

java.applet	AppletContext, Applet, AudioStub, AudioClip etc	Required for developing applets that participate on client-side in Internet (Web) programming.
java.awt	Button, Choice, TextField, Frame, List, Checkbox etc.	Essential for developing GUI applications.
java.awt.event	MouseListener, ActionListener, ActionEvent, WindowAdapter etc.	Without these classes, it is impossible to handle events generated by GUI components
java.sql	DriverManager, Statement, Connection, ResultSet etc	Required for database access in JDBC applications.

Example

```
// A simple package
package mypack;

class Balance {
  String name;
  double bal;

  Balance(String n, double b) {
    name = n;
    bal = b;
  }

  void show() {
    if(bal<0)
      System.out.print("--> ");
    System.out.println(name + ": $" + bal);
  }
}

class AccountBalance {
  public static void main(String[] args) {
    Balance[] current = new Balance[3];
    current[0] = new Balance("K. J. Fielding", 123.23);
    current[1] = new Balance("Will Tell", 157.02);
    current[2] = new Balance("Tom Jackson", -12.33);

    for(int i=0; i<3; i++) current[i].show();
  }
}
```

Compiling and Running the packages

- First save the .java file in the respective working directory.
- For compiling use the following command:

javac -d directory javafilename

The -d switch specifies the destination where to put the generated class file.

Eg.

C:\Users\kumar-pc\Desktop\JavaCodes>javac -d C:\Users\kumar-pc\Desktop\JavaCodes AccountBalance.java

Or

C:\Users\kumar-pc\Desktop\JavaCodes>javac -d . AccountBalance.java

Dot(.) represents the current folder

The above will create a package named mypack in the folder JavaCodes

- For running use the following command:

java mypack.Classfilename

Eg.

C:\Users\kumar-pc\Desktop\JavaCodes>java MyPack.AccountBalance

Importing Packages

- All of the standard core java classes are stored in some named package.
- Since classes within packages must be fully qualified with their package name or names, it could become tedious to type in the long dot-separated package path name for every class you want to use.
- For this reason, Java includes the **import** statement to bring certain classes, or entire packages, into visibility.

Running imported packages

- First create the package
- Then before compiling the file which is importing the package set the class path.
- C:\Users\kumar-pc\Desktop\JavaCodes>set classpath=C:\Users\kumar-pc\Desktop\JavaCodes\P1;.;

Thanks