Package

By Rahul Barve

Package

- Package is a collection of classes and interfaces.
- Used to keep class library isolated from other libraries.
- Can be used to reduce naming conflicts about classes and interfaces.

Creating a Package

By Rahul Barve

Creating a Package

- Packages are created using `package` statement.
- If used, it must be the first statement in the Java source file.

Creating a Package

- Syntax:

```
package <package-name>;
   //class definition
```

- E.g.

```
package test;
   public class Test {
   ….
   }
```

Accessing Classes

- If two or multiple classes are belonging to same package, one class can directly access other classes irrespective of whether they are declared as `public` or not.

Accessing Classes

- E.g.

```
package business;
public class Address {….}

package business;
public class Customer {
   Address commAddress;
   …
}
```

Accessing Classes

- If classes belong to different packages, then one class has to import classes from other packages provided they are declared as `public`.
- This is done using the `import` statement.

Accessing Classes

- E.g.

```
package residence;
public class Address {…}

package college;
import residence.Address;
public class Student {
    Address residentialAddress;
    …
}
```

Sub Packages

Sub Packages

- A package within another package is called as a sub package.
- To import classes from a sub package, the name of the super package is mandatory.
- E.g.

```
import p1.p2.*;
```

Default Package

By Rahul Barve

Default Package

- Whenever a class is declared without `package` statement, then that class is said to be a part of a `default package`.

Default Package

- Such classes cannot be imported by classes coming from other packages; and hence the use of default package is discouraged.

Access Modifiers Revisited

By Rahul Barve

Access Modifiers Revisited

- Java provides 4 access modifiers: `private`, `public`, `protected` and `default`.
- Except `private`, remaining behave same unless different packages are used.

Access Modifiers Revisited

- If different packages are used then:
  - `public` makes the member accessible from anywhere.
  - `protected` makes the member accessible throughout the entire package as well as outside the package if the class is a subclass.
  - `default` makes the member accessible throughout the entire package but not outside the package. Hence it is also known as package level access modifier.