

Packages

Packages

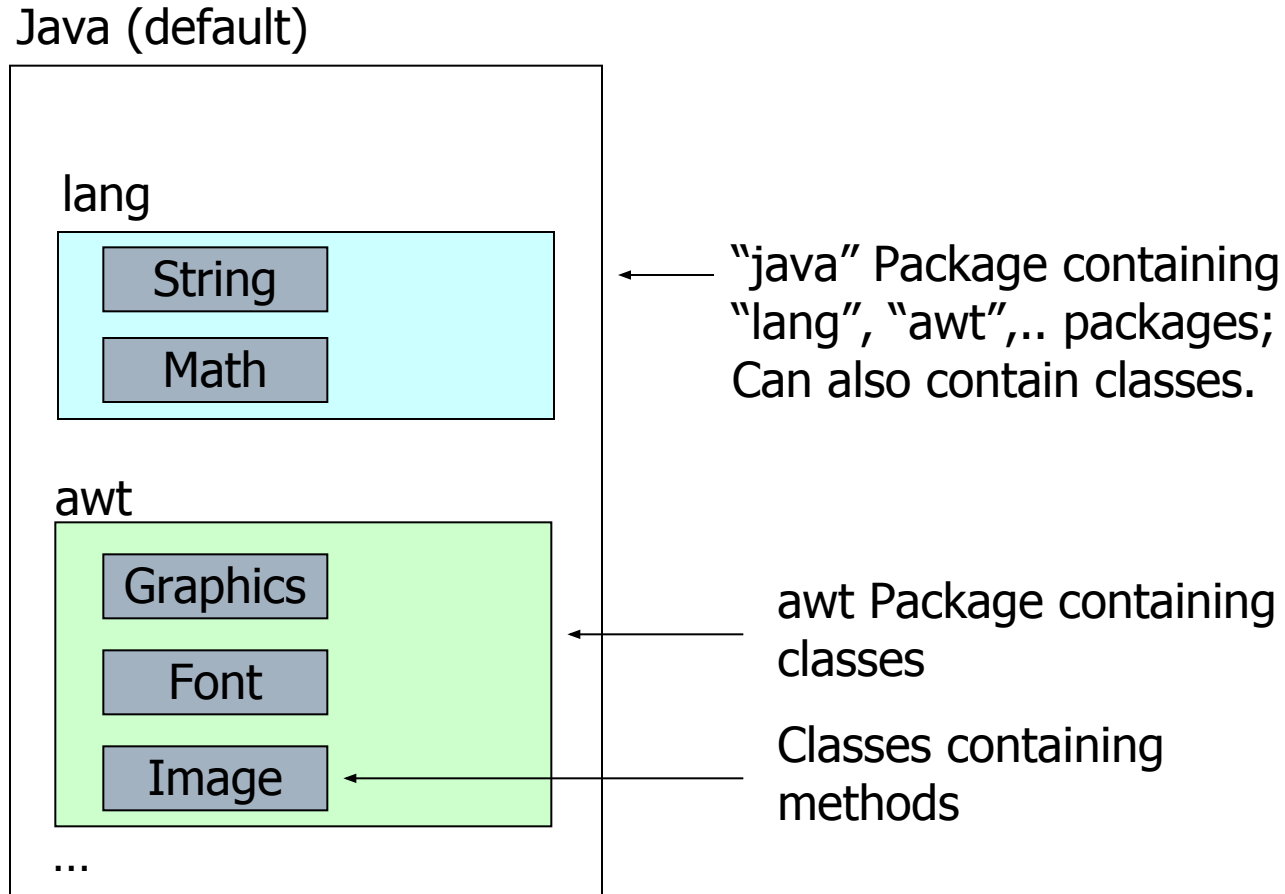
- Packages are Java's way of grouping a number of related classes and/or interfaces together into a single unit. That means, packages act as "containers" for classes.
- The benefits of organising classes into packages are:
 - The classes contained in the packages of other programs/applications can be reused.
 - In packages classes can be unique compared with classes in other packages. That two classes in two different packages can have the same name. If there is a naming clash, then classes can be accessed with their fully qualified name.
 - Classes in packages can be hidden if we don't want other packages to access them.
 - Packages also provide a way for separating "design" from coding.

Java API Packages

- The Java API provides a large number of classes grouped into different packages based on their functionality.
- The six foundation Java packages are:
 - `java.lang`
 - Contains classes for primitive types, strings, math functions, threads, and exception
 - `java.util`
 - Contains classes such as vectors, hash tables, date and the Scanner class
 - `java.io`
 - Stream classes for I/O
 - `java.awt`
 - Classes for implementing GUI – windows, buttons, menus etc.
 - `java.net`
 - Classes for networking
 - `java.applet`
 - Classes for creating and implementing applets

Using System Packages

- The packages are organised in a hierarchical structure. For example, a package named "java" contains the package "awt", which in turn contains various classes required for implementing GUI (graphical user interface).



Accessing Classes from Packages

- There are two ways of accessing the classes stored in packages:
 - Using fully qualified class name
 - `java.util.Scanner`
 - Import package and use class name directly.
 - `import java.util.Scanner;`
- Selected or all classes in packages can be imported:

```
import package.class;  
import package.*;
```

- Implicit in all programs: `import java.lang.*;`
- package statement(s) must appear first