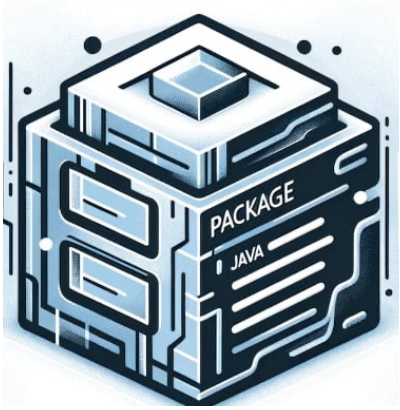
PACKAGES

Packages- Introduction

Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. Packages are used for:

- Preventing naming conflicts. For example there can be two classes with name Employee in two packages, college.staff.cse.Employee and college.staff.ee.Employee
- Making searching/locating and usage of classes, interfaces, enumerations and annotations easier.
- Providing controlled access: protected and default have package level access control. A protected member is accessible by classes in the same package and its subclasses. A default member (without any access specifier) is accessible by classes in the same package only.
- Packages can be considered as data encapsulation (or data-hiding).



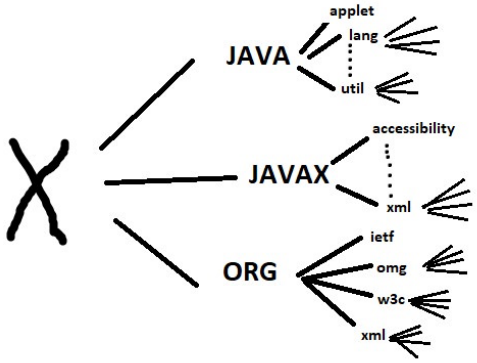
All we need to do is put related classes into packages. After that, we can simply write an import class from existing packages and use it in our program. A package is a container of a group of related classes where some of the classes are accessible are exposed and others are kept for internal purpose. We can reuse existing classes from the packages as many time as we need it in our program.

Every class is part of some package. If no package is specified, the classes in the file goes into a special unnamed package (the same unnamed package for all files). All classes/interfaces in a file are part of the same package. Multiple files can specify the same package name. If package name is specified, the file must be in a subdirectory called name (i.e., the directory name must match the package name). We can access public classes in another package.

Package naming conventions : Packages are named in reverse order of domain names, i.e., com.company For example, in a college, the recommended convention is college.tech.cse, college.tech.ee, college.art.history, etc.

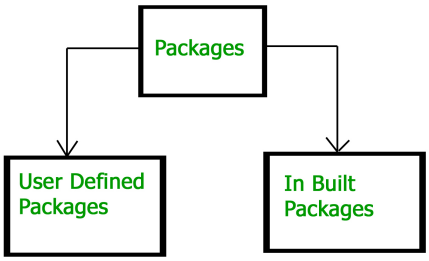
Adding a class to a Package : We can add more classes to a created package by using package name at the top of the program and saving it in the package directory.

Subpackages: Packages that are inside another package are the subpackages. These are not imported by default, they have to imported explicitly. Also, members of a subpackage have no access privileges, i.e., they are considered as different package for protected and default access specifiers. util is a subpackage created inside java package.



Accessing classes inside a Package: Refer //LINK

Types of Packages



Built-in Packages: These packages consist of a large number of classes which are a part of Java. Some of the commonly used built-in packages are:

- java.lang: Contains language support classes(e.g. classes which defines primitive data types, math operations). This package is automatically imported.
- java.io: Contains classes 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). 6)
- java.net: Contain classes for supporting networking operations.

User-defined packages: These are the packages that are defined by the user. First we create a directory myPackage (name should be same as the name of the package). Then create the MyClass inside the directory with the first statement being the package names.

Static Import

Static import is a feature introduced in Java programming language (versions 5 and above) that allows members (fields and methods) defined in a class as public static to be used in Java code without specifying the class in which the field is defined.