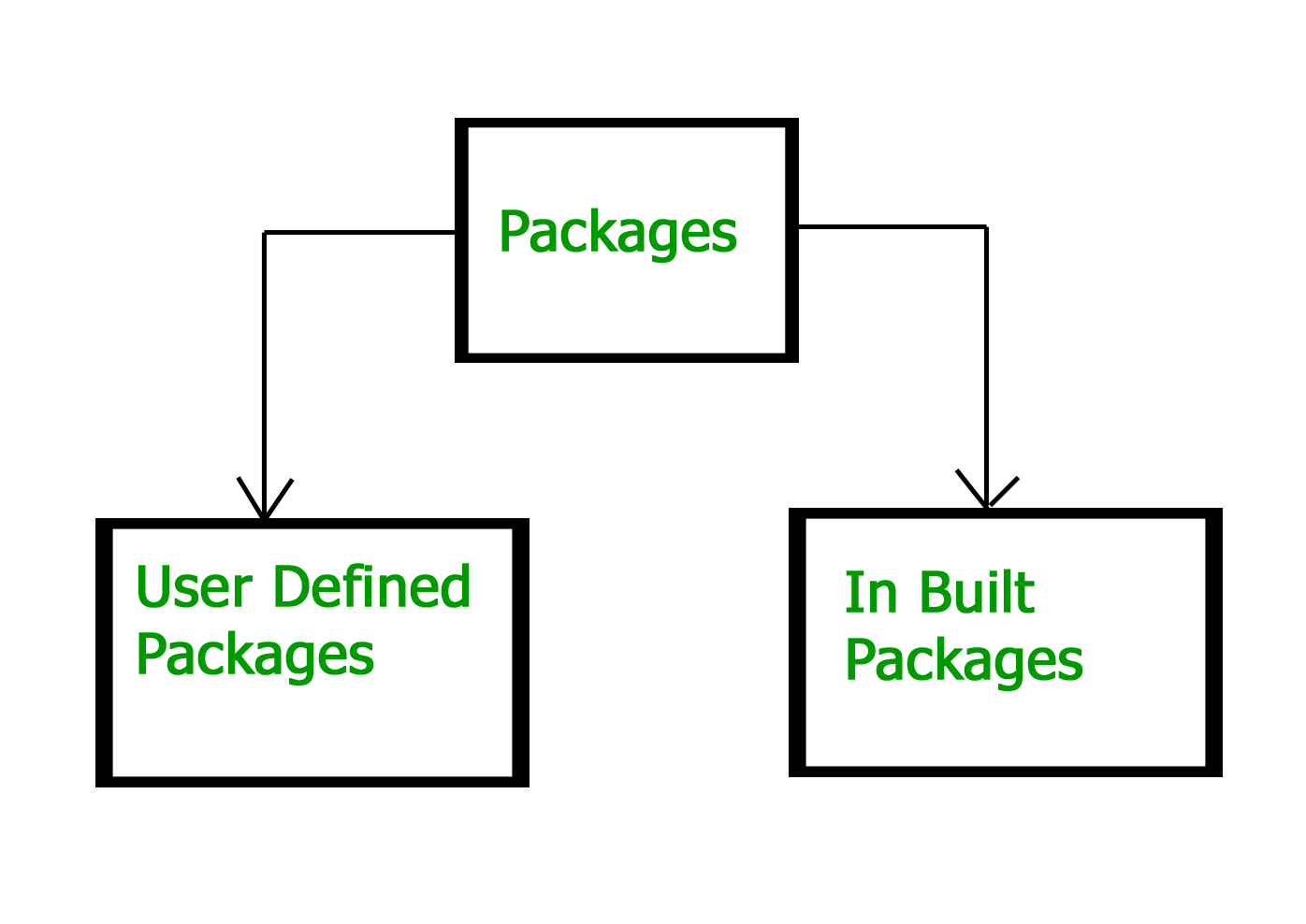
**Java Packages**

* A package in Java is similar to a folder in a file system.
* A package is a collection of sub-packages, classes, and interfaces, much like a folder can contain sub-folders and files.

**Package keyword**

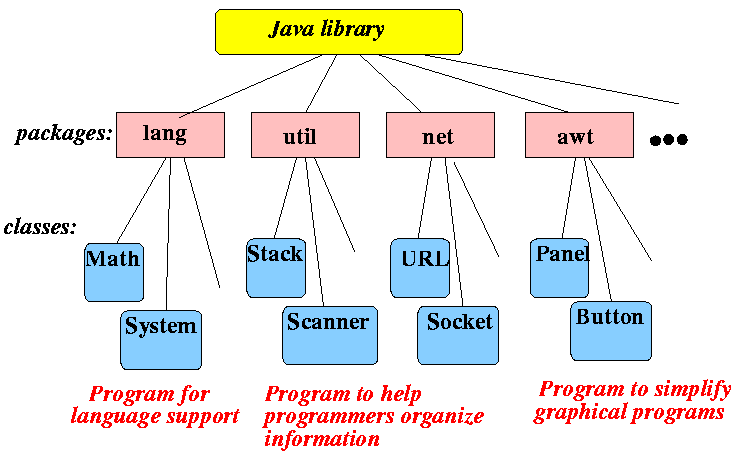
* It is used to collect similar types of classes, interfaces and sub-packages.

**Types of Packages**



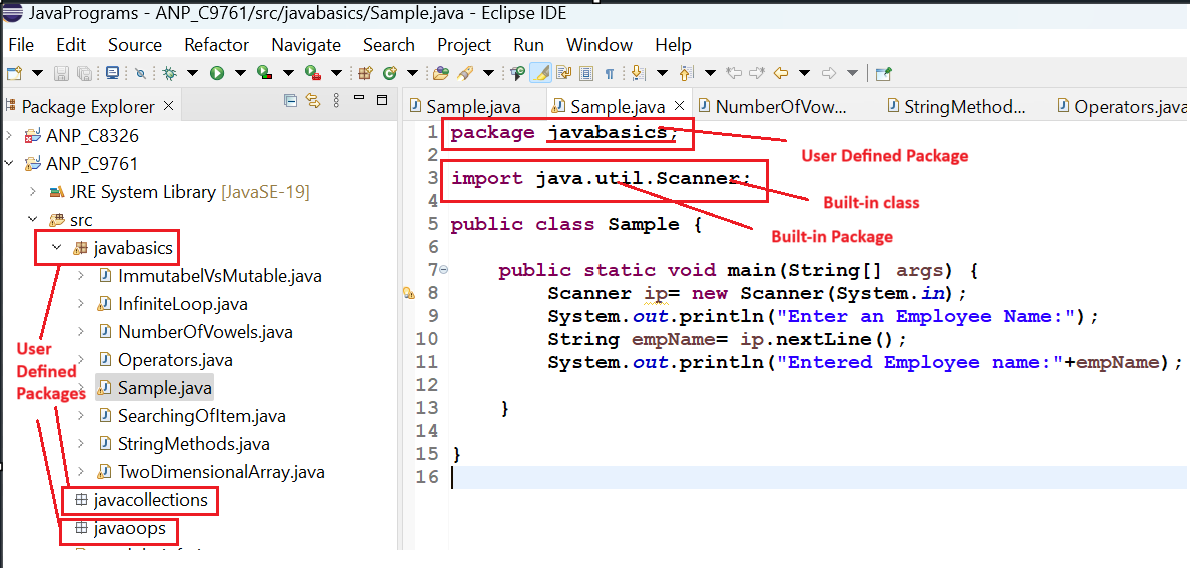
Built- in or Predefined Packages

* Already defined within java itself



User Defined Package

* Java developers(users) are defining this package.



**Note:**

1. Default packages are assigned for enumerations, interfaces, classes and annotation types if no package statement is specified.

2. One source file can only have a single package statement.

**import keyword**

* It is used to give access to all the classes and interfaces of one package to another package
* If we import a package then the classes and interfaces of that package will be imported but cannot import the sub-package.



**System Packages**

1. lang

🡪contains all the fundamental language support classes of Java. i.e

* Math Class
* String Class
* StringBuilder Class
* etc.
* The classes can be imported as follows:
* import java.lang.Math;
  + This will import Math class from java library -->lang package.
  + Math class has many methods like .pow(), .max(), .min(), avg(), pow(), sin(), cos(),.....
* import java.lang.String;
  + String class has many methods like .toUpper(), .toLower(), .indexOf(), .charAt(), .concat(), etc.

2. util

🡪contains Java legacy collection classes and utility classes i.e

* Scanner Class
* Arrays Class
* Collections Class
* Stack Class
* LinkedList Class etc.

The classes can be imported as follows:

* import java.util.Scanner;
  + //This will import Scanner class from java library -->util package.
  + Scanner class has many methods like nextInt(), nextFloat(),......
* import java.util.\*

// This will import the entire util package (all the classes from java library-- >util package).

3. awt(Abstract Windowing Toolkit)

-->contains the graphics classes, including Graphics and Graphics 2D.

**Real-time uses of Java packages**

1. Organizing Code in Large Applications: For example, in a banking application, you could have separate packages like com.bank.accounts, com.bank.transactions, and com.bank.customers to keep classes related to these functionalities organized.

2. Reusing Libraries: Java has predefined packages like java.util, java.io, and java.sql that provide reusable utility classes. For instance, java.util contains collections (e.g., ArrayList, HashMap), which are essential for managing data structures efficiently.

3. Security and Access Control: By grouping classes in packages, you can set different access levels for classes and methods, controlling their visibility and securing sensitive parts of the code. For example, you might place sensitive classes in a private package accessible only within the module.

4. Web Applications: In web development frameworks like Spring, packages are used to separate components like controllers, services, and repositories, following the Model-View-Controller (MVC) design pattern. This enhances maintainability and code readability.

5. Namespace Management: Packages help avoid naming conflicts. For instance, if two developers create a Date class, placing them in separate packages (e.g., java.util.Date and com.myapp.utils.Date) prevents conflicts.