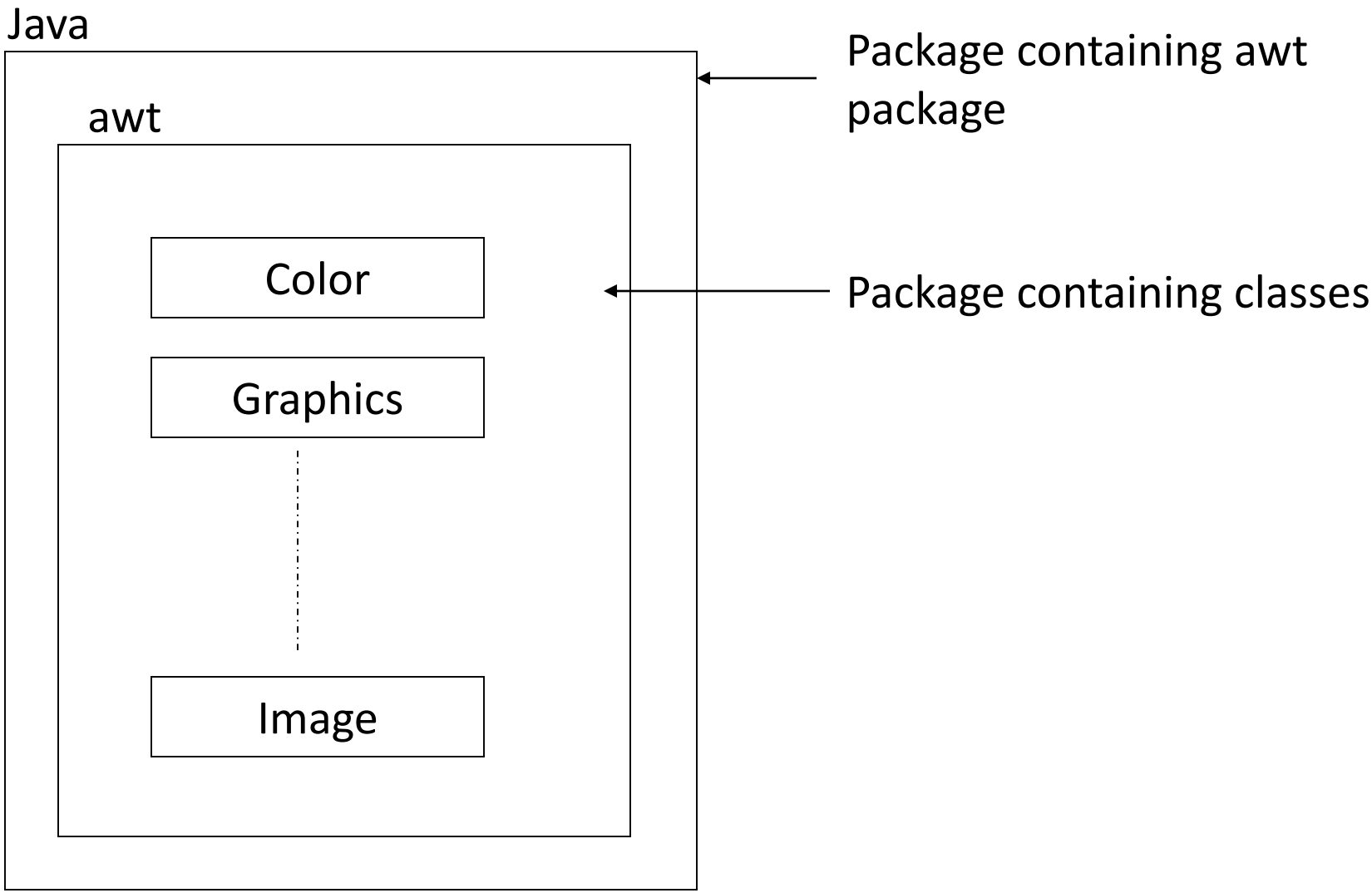
Package in Java

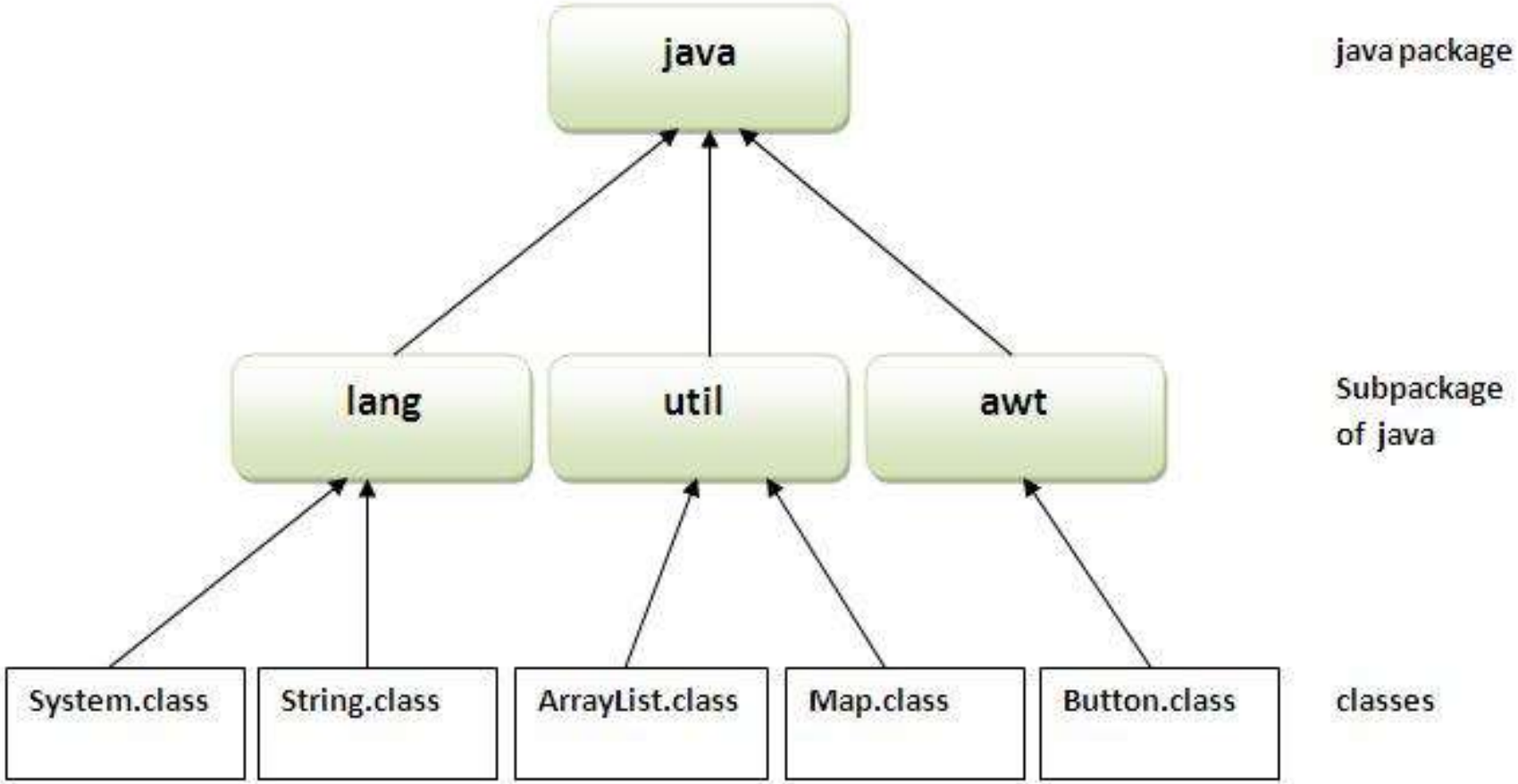
- Packages are used in Java in order to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations and annotations easier, etc.
- A package is a collection of related Java entities (such as classes, interfaces, exceptions, errors and enums).
- A **package** provides a mechanism for grouping a variety of similar types of classes, interfaces and sub-packages.
- Grouping is based on functionality.
- Java packages can be stored in compressed files called JAR files (Java Archieve)

- Resolving naming conflict of classes by prefixing the class name with a package name.
  - com.zzz.Circle and com.yyy.Circle are two distinct classes.
  - Fully-qualified class name - package name plus class name. This mechanism is called Namespace Management.
- **Benefits:**
  - The classes contained in the packages of other programs can be reused.
  - In packages, classes can be unique compared with classes in other packages.
  - Packages provides a way to hide classes.

- Two types of packages:
  - Java API packages
  - User defined packages
- **Java API Packages:**
  - A large number of classes grouped into different packages based on functionality. Examples:
    - java.lang - Contains classes for primitive types, strings, math functions, threads, and exception
    - java.util - Contains classes such as vectors, hash tables, date etc.
    - 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

Package


Java
awt
Color
Graphics
Image
Package containing awt package
Package containing classes


java
java package
lang
util
awt
Subpackage
of java
System.class
String.class
ArrayList.class
Map.class
Button.class
classes

- Package names are dot separated, e.g., java.lang.
- Packages Avoid name space collision. There can not be two classes with same name in a same Package But two packages can have a class with same name.
- Exact Name of the class is identifed by its package structure.

<< Fully Qualified Name>>

- java.lang.String ; java.util.Arrays; java.io.BufferedReader ; java.util.Date

Accessing Classes in a Package

- Fully Qualified class name:

  Example: java.awt.Color

- **import** packagename.classname;

  Example: import java.awt.Color;

  or

- **import** packagename.*;

  Example: import java.awt.*;

- Own package – package package_name;

  Example - package MyPackage;

Creating Your Own Package

- Declare the package at the beginning of a file using the form

  package *packagename;*

- Define the class that is to be put in the package and declare it **public**.
- Create a subdirectory under the directory where the main source files are stored.
- Store the listing as classname.java in the subdirectory created.
- Compile the file. This creates .class file in the subdirectory.

Example:

```
package firstPackage;


Public class FirstClass
{
   //Body of the class
}
```

Finding Packages

Two ways:

- By default, java runtime system uses current directory as starting point and search all the subdirectories for the package.
- Specify a directory path using CLASSPATH environmental variable.