SNS COLLEGE OF ENGINEERING

Coimbatore-107
An Autonomous Institution

COURSE NAME : 23CSB201 & Object Oriented Programming

II YEAR/ III SEMESTER

UNIT – II INHERITANCE, PACKAGES, INTERFACE

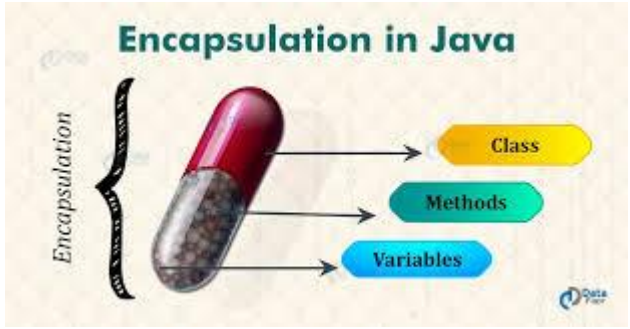
Topic: Packages

Dr.P.Poonkodi

Assistant Professor(SG)

Department of Computer Science and Technology

Real time example



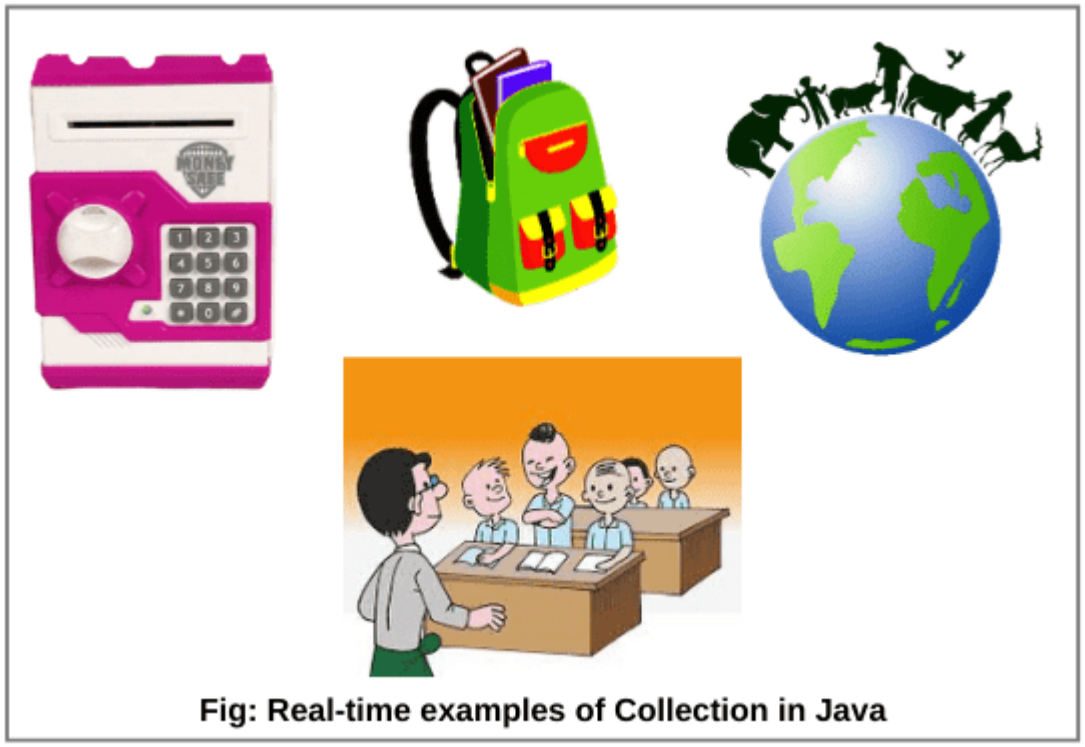
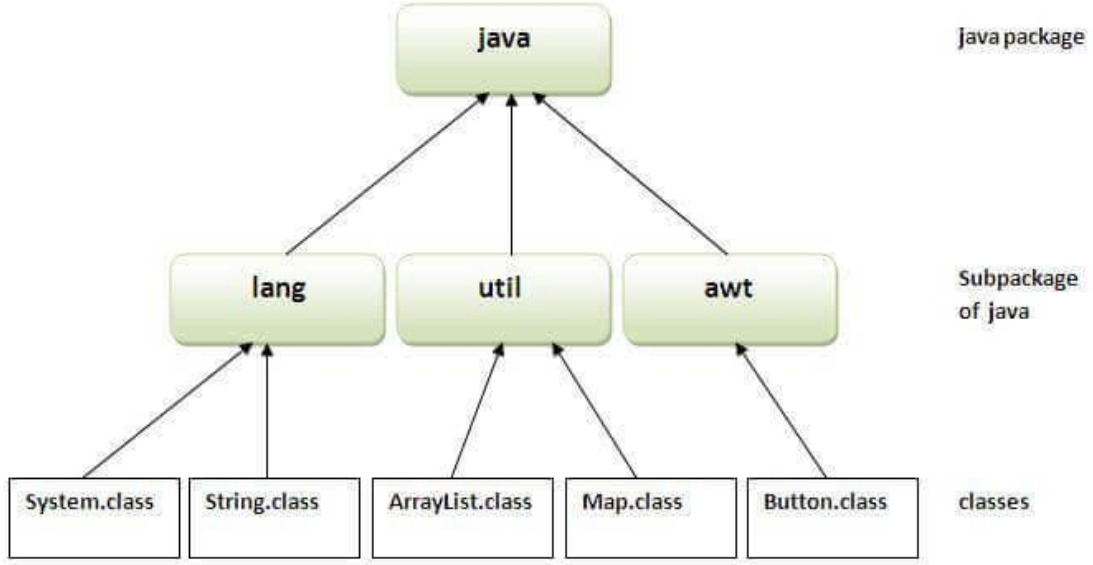


Fig: Real-time examples of Collection in Java

Introduction

- A **java package** is a collection of classes
- Used for reuse of classes
- Reduce the program complexity
- 2 Types
  - built-in package
  - user-defined package
- built-in package
  - Import java.io.*;
  - Import java.util.scanner;
- Super Package- java
- Sub Package-io, util
- Class- scanner, *



Introduction

- Package in Java is a mechanism to encapsulate a group of classes, sub-packages, and interfaces
- All we need to do put it in 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 classes are accessible or exposed and others are kept for internal purpose

example of java package

- **package keyword** is used to create a package in java

```
package mypack;
public class Simple
{
 public static void main(String args[])
{
    System.out.println("Welcome to package");
   }
}
```

example of java package

- How to compile java package
  - Syntax
    - javac -d javafilename.java
  - Example
    - javac -d . Simple.java

- How to run java package program

To Compile: javac -d . Simple.java

To Run: java mypack.Simple

Access package from another package

- three ways to access the package from outside the package
  - import package.*;
  - import package.classname;
  - fully qualified name

Using packagename.*

- If you use package.* then all the classes and interfaces of this package will be accessible but not sub-packages

```
package pack;
public class A
{
  public void msg()
{
System.out.println("Hello");
}
}
```

Compile: javac –d.package/classname.java

Using packagename.*

- The import keyword is used to make the classes and interface of another package accessible to the current package

```
package mypack;
import pack.*;

class B{
  public static void main(String args[]){
   A obj = new A();
   obj.msg();
  }
}
```

Compile: javac –d. package/classname.java

Run: java packagename.classname

Using packagename.classname

```
package pack;
public class A
{
  public void msg()
{
        System.out.println("Hello");
}
}
```

Compile: javac –d.package/classname.java

```
package mypack;
import pack.A;

class B
{
  public static void main(String args[])
{
   A obj = new A();
   obj.msg();
  }
}
```

Compile: javac –d.package/classname.java

Run: java packagename.classname

Using fully qualified name

```
package pack;
public class A
{
  public void msg()
{
        System.out.println("Hello");
}
}
```

```
package mypack;
import pack.A;

class B
{
  public static void main(String args[])
{
        pack.A obj = new pack.A();
        //using fully qualified name
         obj.msg();
  }
}
```

Advantages

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained
2) Java package provides access protection
3) Java package removes naming collision

References

- Java : the complete Reference (Eleventh Edition), Herbert Schildt, 2018.


Thank You!