

# 6. Les Packages

Mr IGHIL Mohamed

INSIM Boumerdes

2025

# 1. Présentation

- Un **package** regroupe un ensemble de **classes** qui ont des **fonctionnalités similaires**.
- Par exemple, on peut regrouper toutes les classes qui traitent des objets **graphiques**.

- C'est un package déjà défini dans Java - le package **awt**.
- L'utilité principale de ce regroupement est de permettre à importer très rapidement un ensemble d'outils, ainsi d'établir un bon environnement pour un programme.

- Un package contient seulement des classes.
- Les classes qui sont à inclure dans un package doit être mises dans un fichier dont la première ligne spécifie le package comme suit:

```
package <nom_package>;
```

```
public class C1 {...}
```

Après **compilation**, la classe **C1**  
sera mise dans le package  
<nom\_package>.

```
graph TD; Packages --> UserDefined[User Defined Packages]; Packages --> InBuilt[In Built Packages];
```

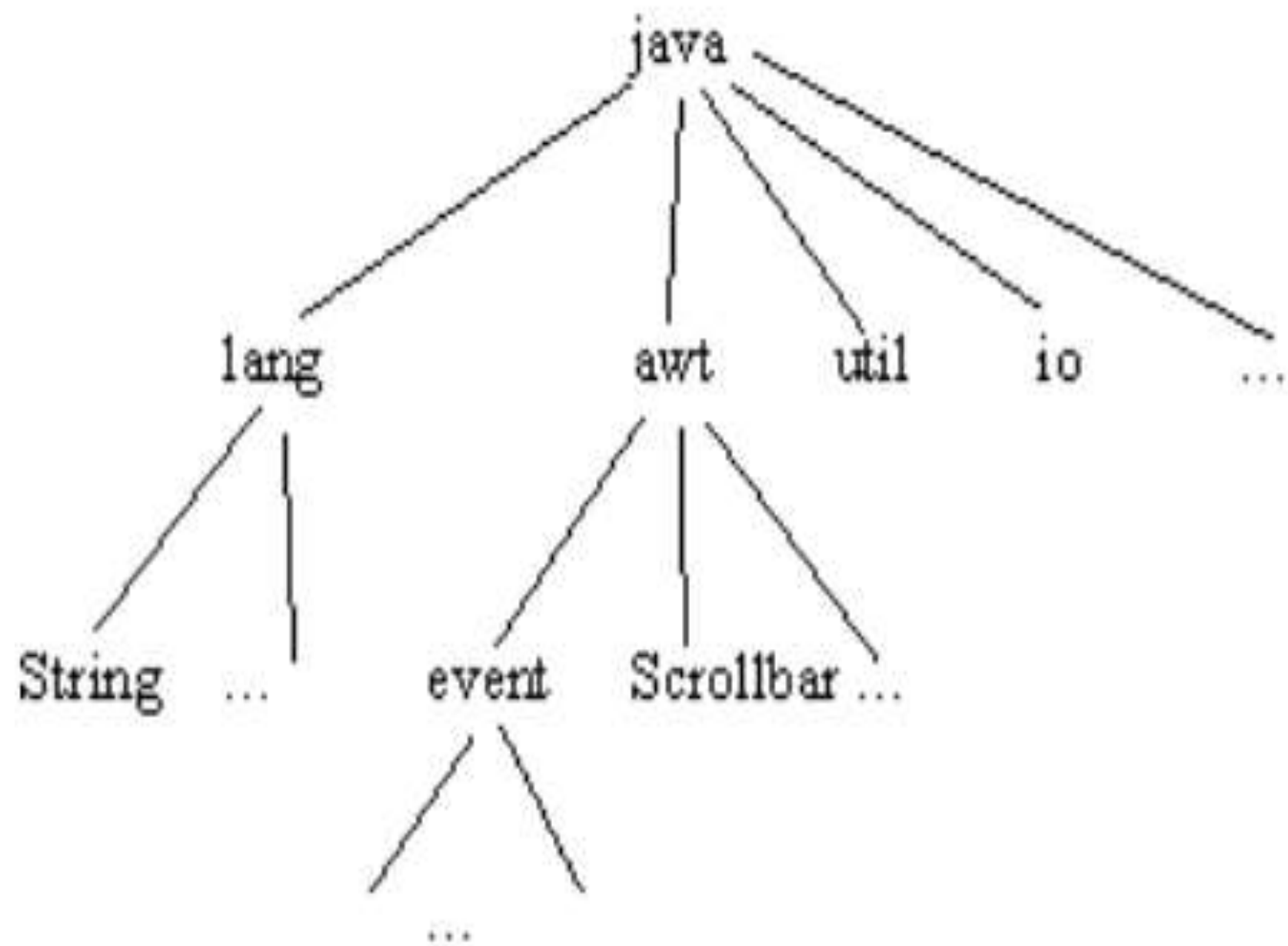
Packages

User Defined  
Packages

In Built  
Packages

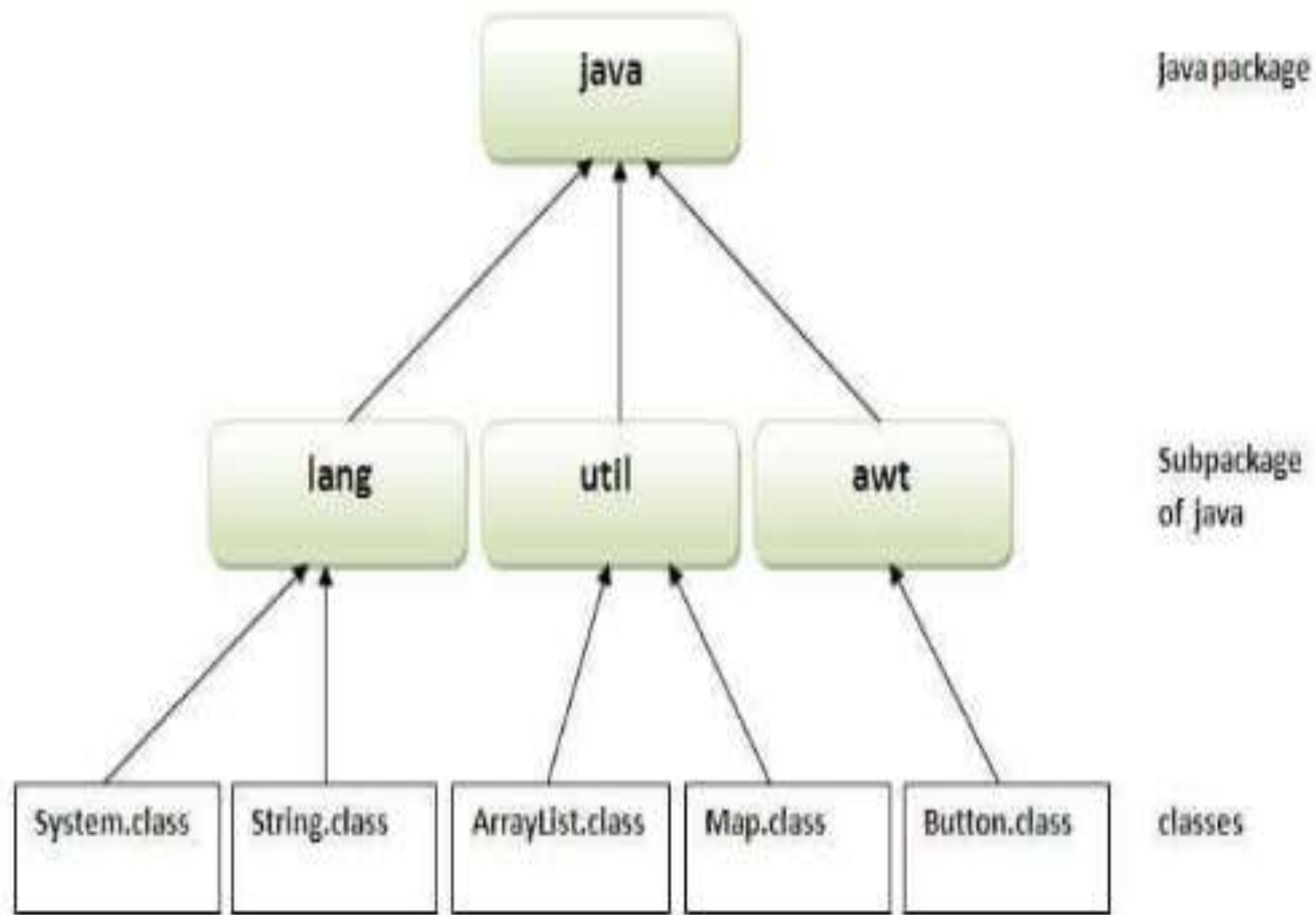
## 2. Organisation des packages

Il existe aussi une **hiérarchie** dans les packages. La racine des packages prédéfinis en Java est **"java"**. Quelques **sous-packages** dans **"java"** sont **"lang"**, **"awt"**, **"util"**, etc. On peut voir ceci comme un **arbre**:





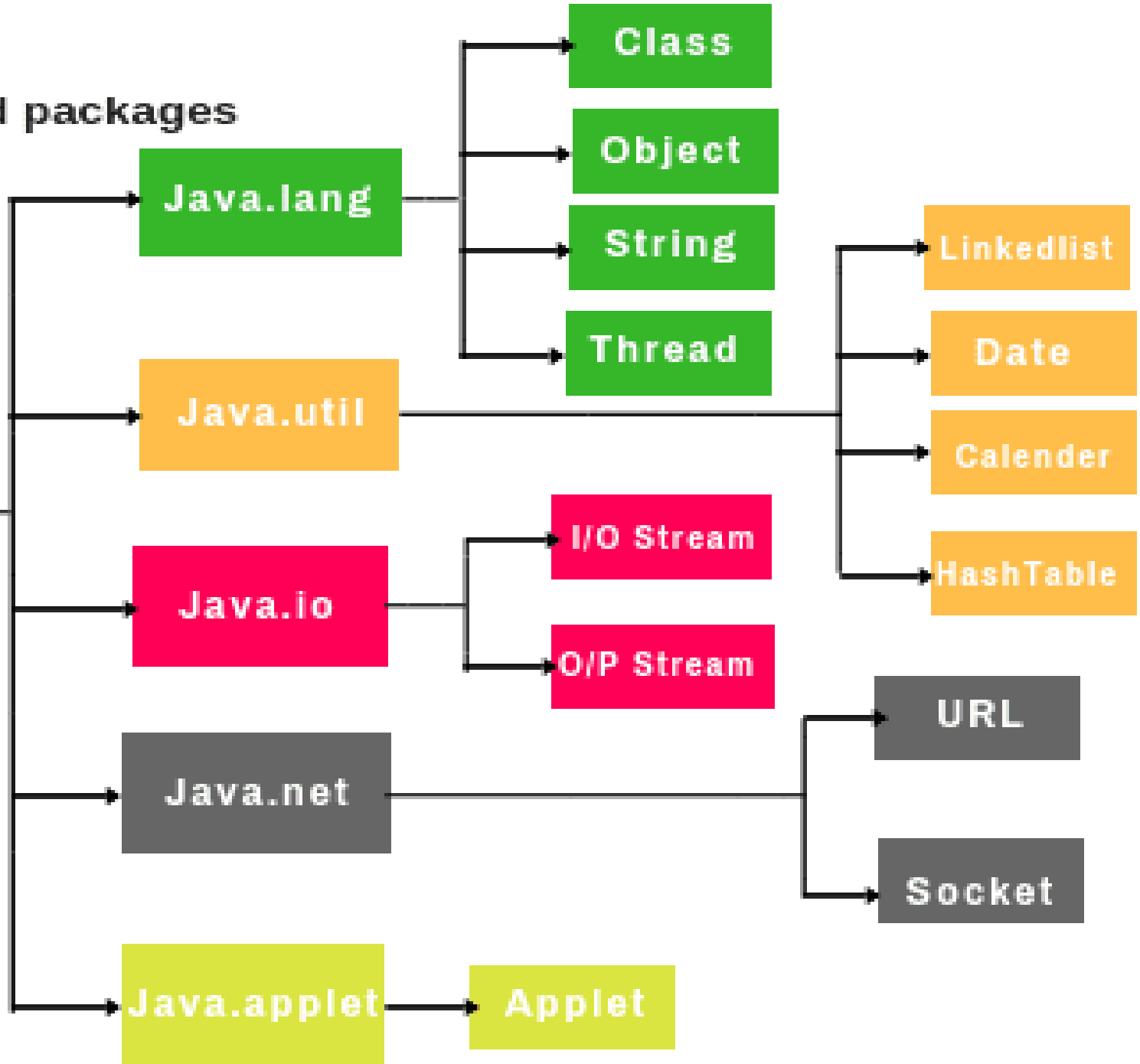
Dans cette figure, un **nom** en **Majuscule** correspond à une **classe**, et en **minuscule** à un **package**. Ainsi, on peut voir qu'un package peut contenir des sous-packages, et des classes.



## 5 Predefined packages

Java 1.8

Java



- **java.lang** : classes de base de Java
- **java.util** : utilitaires
- **java.io** : entrées-sorties
- **java.awt** : interface graphique
- **javax.swing** : interface graphique avancée
- **java.applet** : applets
- **java.net** : réseau

### 3. Utilisation d'un package

Dans un programme, on peut importer une classe ou toutes les classes d'un package comme suit:

```
import <package>.C1;
```

```
import <package>*;
```

La première ligne importe **juste**  
la classe **C1** du package, tandis  
que la seconde importe **toutes**  
les **classes** du package.

```
package vehicule.motorise ;
```

```
import java.util.* ; // importation de tout le package java.util, dont la classe Date
```

```
public class Voiture {
```

```
    private Date dateAchat ;
```

```
}
```

```
package vehicule.motorise ;
```

```
import java.util.Date ; // importation d'une classe unique
```

```
public class Voiture {
```

```
    private Date dateAchat ;
```

```
}
```

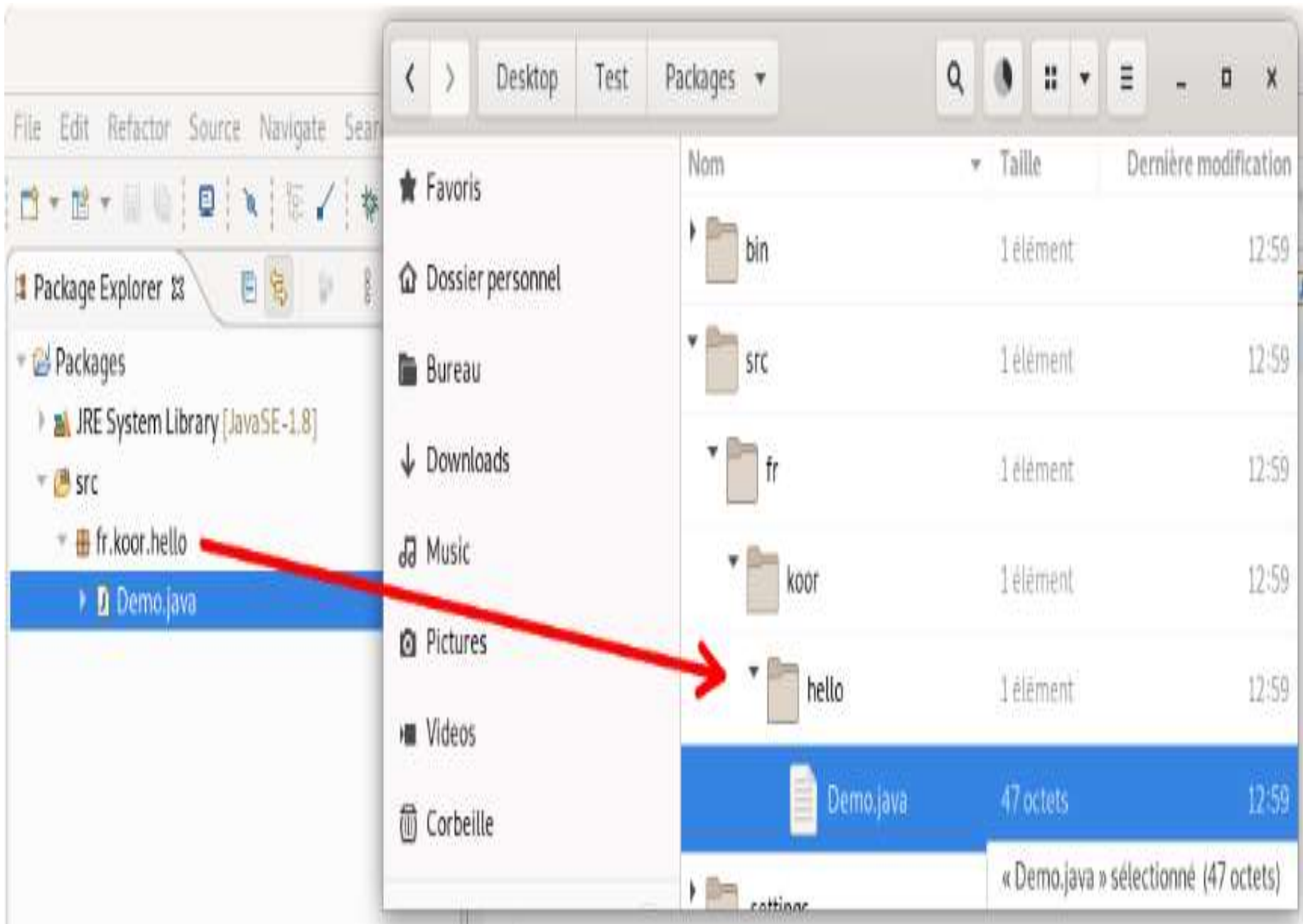


```
package transport ; // la classe Voiture appartient au paquet transport

public class Voiture { // rangée dans un fichier transport/Voiture.java
    // corps de la classe
}
```

```
package transport.motorise ; // la classe Voiture appartient au paquet transport.motorise

public class Voiture { // rangée dans un fichier transport/motorise/Voiture.java
    // corps de la classe
}
```



## 3.1 Importer une classe d'un paquetage

Pour pouvoir désigner une classe d'un autre paquetage par son nom terminal, il faut l'importer

```
import java.util.ArrayList;
```

```
public class Classe {
```

```
...
```

```
ArrayList liste = new ArrayList();
```

```
...
```

On peut utiliser une classe sans l'importer l'importation permet seulement de raccourcir le nom d'une classe dans le code :

dans ce cas, il faut indiquer le **nom total**

```
java.util.ArrayList Liste = new java.util.ArrayList();
```

```
package com.scientecheasy.calculate;

public class Sum
{
    int a = 20;
    int b = 30;
    public void cal()
    {
        int s = a + b;
        System.out.println("Sum: " +s);
    }
}
```

```
package com.maths.calculator;
import com.scientecheasy.calculate.*;
class SumTest
{
    public static void main(String[] args)
    {
        Sum s = new Sum();
        s.cal();
    }
}
```