Package

1. Concetto
2. Realizzazione in Java
3. I namespace del C++

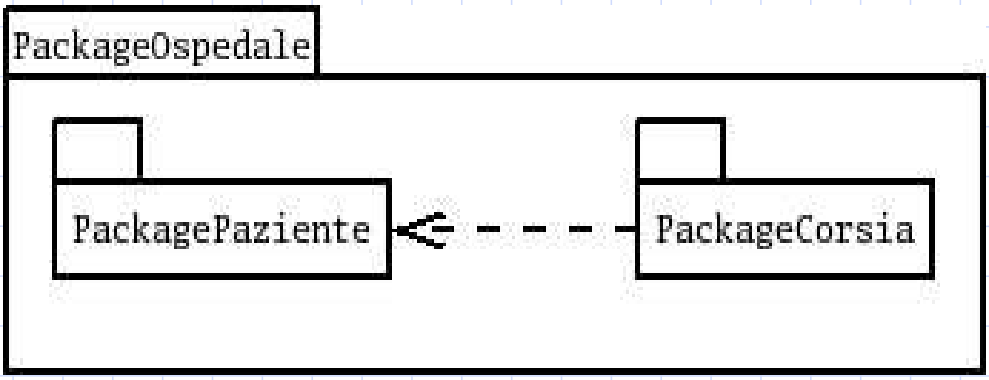
Package: *definizione*

- Non esiste una definizione universalmente accettata di package.
- In un sistema ad oggetti, un **package** è spesso una collezione di classi.
- Esempi di package sono le librerie di classi o le collezioni di classi di una applicazione.
- In generale un package può contenere anche altri package, un po' come una cartella può contenere altre cartelle nella gestione dei file.
- Il motivo per raccogliere classi in package risiede nella maggiore facilità di creare architetture software e nell'evitare il **conflitto di nomi,** cioè poter utilizzare nomi identici per classi diverse purché appartenenti a package distinti.

Esempio in UML

Il **PackageOspedale** contiene solo due package: **PackagePaziente** e **PackageCorsia**.

La linea tratteggiata indica che PackageCorsia è dipendente da PackagePaziente poiché, ad esempio, la classe Paziente (del PackagePaziente, non visualizzata) è usata da una classe (es. PostoLetto) del PackageCorsia.



I Package in Java

- In Java esistono dozzine di package che includono classi pronte per l'uso. Esempi sono:
  - **`java.lang`** per classi di supporto al linguaggio
  - **`java.util`** classi di utilità
  - **`java.io`** classi di input/output
  - **`java.awt`** classi per la gestione della grafica
  - **`java.swing`** classi per interfacce utenti
- Per includere un package si usa l'istruzione **`import`**. Es:

```
import java.awt.*;//awt e i sottopackage
```

Definizione di classi in package in Java

Per inserire una classe in un pacchetto, il file in cui è definita deve iniziare con l'istruzione **package**.

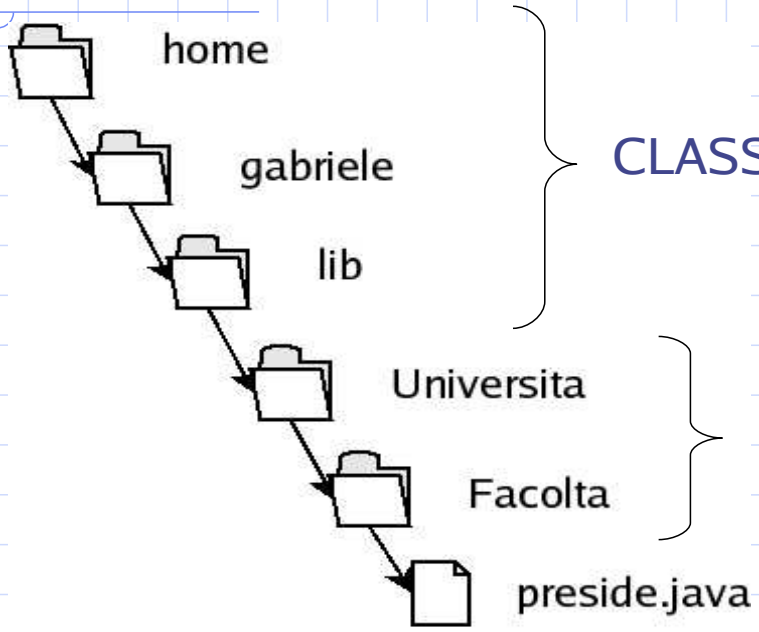
```
package Universita;

class Studente{
public  Studente(.....);     // costruttore e altri metodi
 ....
private string nome;        // dati dello studente
private string cognome;
....
};
```


Localizzazione dei package

- Il compilatore Java cerca i package predefiniti in cartelle del sistema.
- L'insieme dei file di un package risiede in una cartella il cui nome è identico a quello del package.
- Se si crea il file **preside.java** nel package **Facolta** a sua volta parte del package **Universita**, il file sarà posto in una **cartella Facolta** all'interno della **cartella Universita.**
- Tutti i package creati da un utente devono far parte di una ***cartella base*** di cui il compilatore deve essere messo a conoscenza, ad es. tramite la variabile di sistema `CLASSPATH`
- es. Linux: `export CLASSPATH=/home/gabriele/lib:.`
- es.Windows: `set CLASSPATH=c:\home\gabriele\lib;.`

Cartelle per i package




```
package Universita.Facolta;

class Preside{....
};
```

preside.java

Namespace in C++

- In C++ non esistono i package, ma i **namespace**.
- I namespace definiscono un ambito in cui possono essere dichiarati identificatori (di classe o altro) allo scopo di evitare conflitti di nomi.
- Tutti gli identificatori delle librerie standard del C++ sono definiti nel namespace **std**
- Al contrario dei package del Java, i namespace sono moduli logici, non fisici.
- Per includere una classe in un namespace si usa l'istruzione **namespace**.

Definizione di classi in namespace del C++

Per inserire una classe in un namespace, si procede come nel seguente esempio. I namespace sono "aperti": classi dichiarate in file diversi possono appartenere allo stesso namespace.

```
namespace Universita{

class Studente{
public:
   Studente(.....);     // costruttore e altri metodi
 ....
private:
   string nome, cognome;       // dati dello studente
....
 };
} //fine del namespace
```


Richiamo di elementi in un namespace

Per accedere ad elementi di un namespace si può usare l'operatore di scope **::**

```
void main() {
    Universita::preside p(...);
....
}
```

oppure l'istruzione **`using`.**

```
using namespace Universita;
void main() {
    preside p(...);
....
}
```