Università di Torino – Facoltà di Scienze MFN
Corso di Studi in Informatica

Programmazione I - corso B
a.a. 2009-10

prof. Viviana Bono

Blocco 10 – Import di classi e di metodi statici

Dichiarazioni di import: Introduzione

(l'argomento verrà ripreso in Prog II, si tratta dei package)

Java viene fornito con una ricca libreria standard di classi, ripartite in "pacchi" (packages) e "sottopacchi" (subpackages), che corrispondono a cartelle e sottocartelle.

In qualunque programma Java ci si può riferire a tali classi per mezzo del loro nome "lungo", cioè avente come prefisso anche il nome del package e del subpackage. Ad esempio, l'istruzione:

```
java.util.Scanner tastiera = new java.util.Scanner(System.in);
```

può essere inserita correttamente in qualunque programma java. Tuttavia nomi così "lunghi" sono scomodissimi!

Le dichiarazioni import, che possono essere messe solo all'inizio di un file .java, prima di ogni dichiarazione di classe, permettono di usare in quel file i nomi "corti" delle classi importate, senza il prefisso del package.

Esempio: un programma senza import

```
public class CiaoNome {

  public static void main(String[] args) {
    java.util.Scanner tastiera = new java.util.Scanner(System.in);
    System.out.print("Come ti chiami? ");
    String nome = tastiera.nextLine();
    javax.swing.JOptionPane.showMessageDialog( ... );
   ...
  }
}
```

"Importiamo" le classi che ci servono

```
import java.util.Scanner;
import javax.swing.JOptionPane;
// ora possiamo usare i nomi corti

public class CiaoNome {

  public static void main(String[] args) {
    Scanner tastiera = new Scanner(System.in);
    System.out.print("Come ti chiami? ");
    String nome = tastiera.nextLine();
    JOptionPane.showMessageDialog( ... );
   ...
  }
}
```


"Importiamo" le classi che ci servono

Invece di importare una sola classe, si possono importare (cioè rendere indicabili coi nomi corti) tutte le classi di un package, ad esempio:

```
import java.util.*; // * e` detto wild card
```

Ciò è utile quando si devono usare più classi di uno stesso package (ad esempio diverse classi di java.util).

NB Le dichiarazioni di import NON copiano programmi né in formato sorgente né in formato compilato; rendono semplicemente utilizzabili nomi corti invece di nomi lunghi.

Pertanto, importare tutte le classi di un package non è penalizzante rispetto a importarne una sola.

Import automatico delle classi di java.lang

Il package java.lang (il nome *lang* è stato inventato come abbreviazione della parola *language*) è privilegiato: le sue classi, che sono quelle di uso più comune (come String, System, Math), vengono importate automaticamente senza bisogno di alcuna dichiarazione. Esempio:

```
class Trigono {
  public static void main(String[] args) {
    String s = "sin(pigreco/2) = ";
    System.out.println(s + Math.sin(Math.PI/2));
  }
}
```

(out è un campo statico della classe System, il quale contiene il riferimento a un oggetto dotato di un metodo println)

Java 1.5: import static

In Java 1.5 è stato introdotto il nuovo costrutto import static, che permette di usare i campi e metodi statici di una classe senza il nome della classe come prefisso.

Esempio:

```
import static java.lang.Math.PI;
import static java.lang.Math.sin;
import static java.lang.System.out;
```

ora la costante PI e il metodo statico sin della classe Math, e il campo statico out della classe System possono essere indicati con i loro nomi "corti"

```
class Trigono {
  public static void main(String[] args) {
    String s = "sin(pigreco/2) = ";
    out.println(s + sin(PI/2));
  }
}
```

NB Import non nessario, ma utile

Java 1.5: import static

Invece di importare un singolo metodo statico o campo statico di una classe, si possono importare tutti gl elementi (campi e metodi) statici di quella classe:

```
import static java.lang.Math.*;
import static java.lang.System.*;

class Trigono {
  public static void main(String[] args) {
    String s = "sin(pigreco/2) = ";
    out.println(s + sin(PI/2));
  }
}
```


Altro Esempio

La classe JOptionPane, del pacco javax.swing, possiede due metodi statici per fare input-output che usiamo.

Prima versione – nessun import

classe JOptionPane e suoi metodi: con nome "lungo"

```
public class CiaoNomeG {

  public static void main(String[] args) {
    String nome = javax.swing.JOptionPane.showInputDialog("Come ti chiami?");
    javax.swing.JOptionPane.showMessageDialog(null, "Ciao, " + nome);
    System.exit(0);
  }
}
```

Stesso Es. con import della classe JOptionPane

Seconda versione – import della classe
classe JOptionPane: con nome corto

```
import javax.swing.JOptionPane;

public class CiaoNomeG {

  public static void main(String[] args) {
    String nome = JOptionPane.showInputDialog("Come ti chiami?");
    JOptionPane.showMessageDialog(null, "Ciao, " + nome);
    System.exit(0);
  }
}
```


Stesso Es. con import dei suoi metodi statici

Terza versione - import dei metodi statici della classe JOptionPane
metodi statici di JOptionPane: con nome corto

```
import static javax.swing.JOptionPane.*;

public class CiaoNomeG {

  public static void main(String[] args) {
    String nome = showInputDialog("Come ti chiami?");
    showMessageDialog(null, "Ciao, " + nome);
    System.exit(0);
  }
}
```