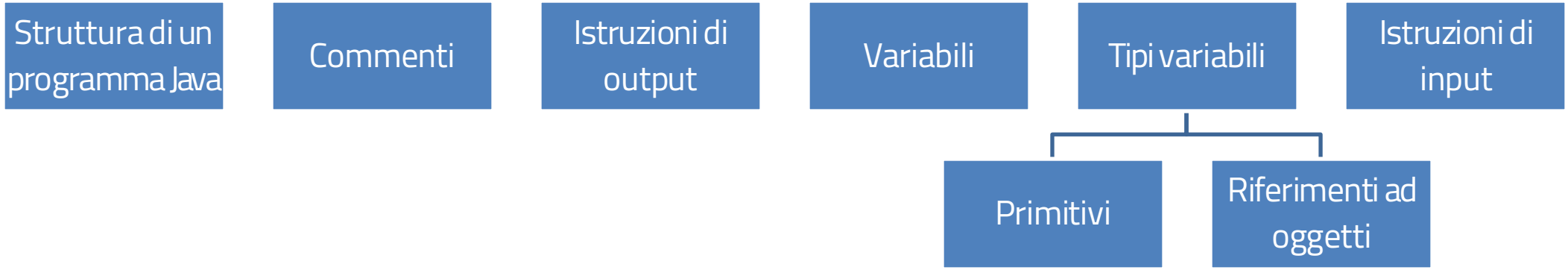
Struttura di un programma, operazioni di I/O e variabili in Java

Docente
Mario Perna
prof.perna.mario@darzo.net

A.S.
2025/2026
Materia
Informatica

Indice


Struttura di un programma Java
Commenti
Istruzioni di output
Variabili
Tipi variabili
Istruzioni di input
Primitivi
Riferimenti ad oggetti

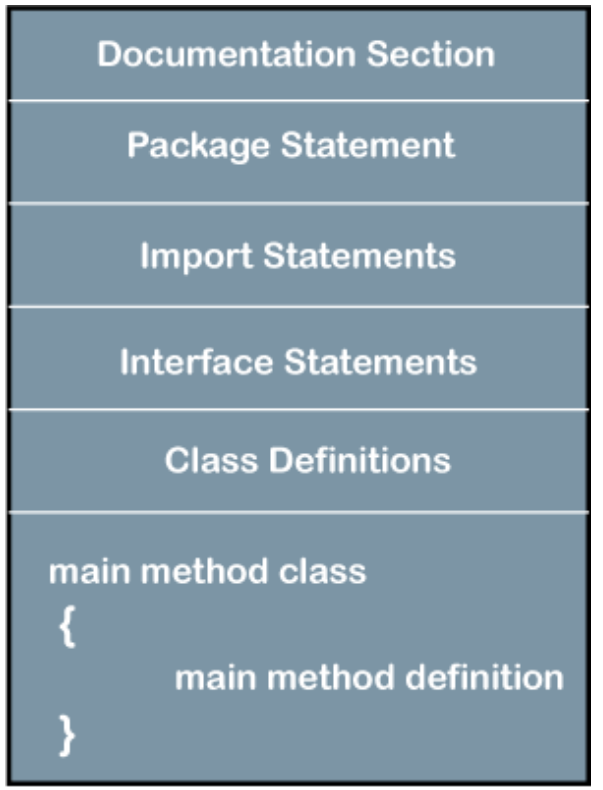
Struttura di un programma Java

Java è un linguaggio di programmazione orientato agli oggetti, indipendente dalla piattaforma e sicuro, che lo rende popolare.
Utilizzando il linguaggio di programmazione Java, possiamo sviluppare un'ampia varietà di applicazioni. Quindi, prima di immergerci nei dettagli, è necessario comprendere in dettaglio la struttura di base del programma Java.

```java
package scuola.testambiente;

/**
 *
 * @author Andre
 */
public class TestAmbiente {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```



Structure of Java Program

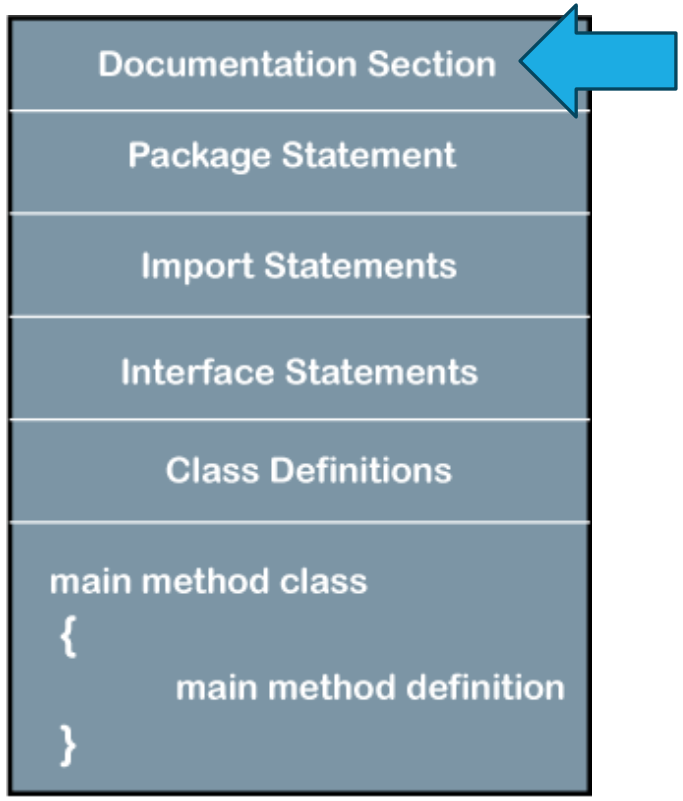
Struttura di un programma Java

Sezione documentazione

La sezione documentazione è una sezione importante ma facoltativa per un programma Java. Include informazioni di base su un programma Java.

Le informazioni includono il nome dell'autore, la data di creazione, la versione, il nome del programma, il nome dell'azienda e la descrizione del programma.

Migliora la leggibilità del programma. Qualunque cosa scriviamo nella sezione documentazione, il compilatore Java ignora le istruzioni durante l'esecuzione del programma. Per scrivere le istruzioni nella sezione documentazione, utilizziamo i commenti.



Structure of Java Program

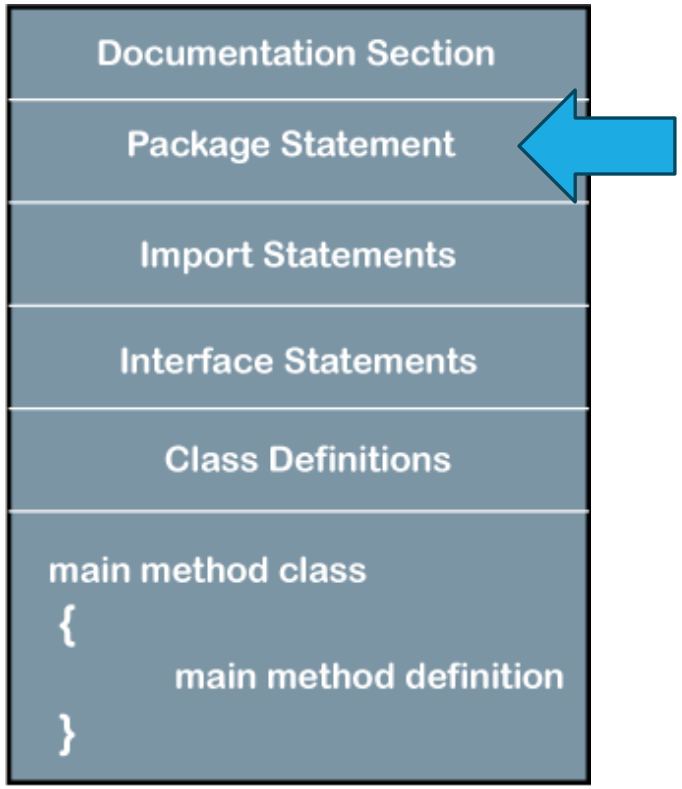
Struttura di un programma Java

Dichiarazione dei pacchetti
La dichiarazione dei pacchetti è facoltativa. In questa sezione, dichiariamo il nome del pacchetto in cui è posizionata la classe.

Deve essere definita prima di qualsiasi dichiarazione di classe e interfaccia. È necessaria perché una classe Java può essere posizionata in pacchetti diversi.

Utilizziamo la parola chiave packageper dichiarare il nome del pacchetto. Ad esempio:

```
package scuola.es4;
```



Structure of Java Program

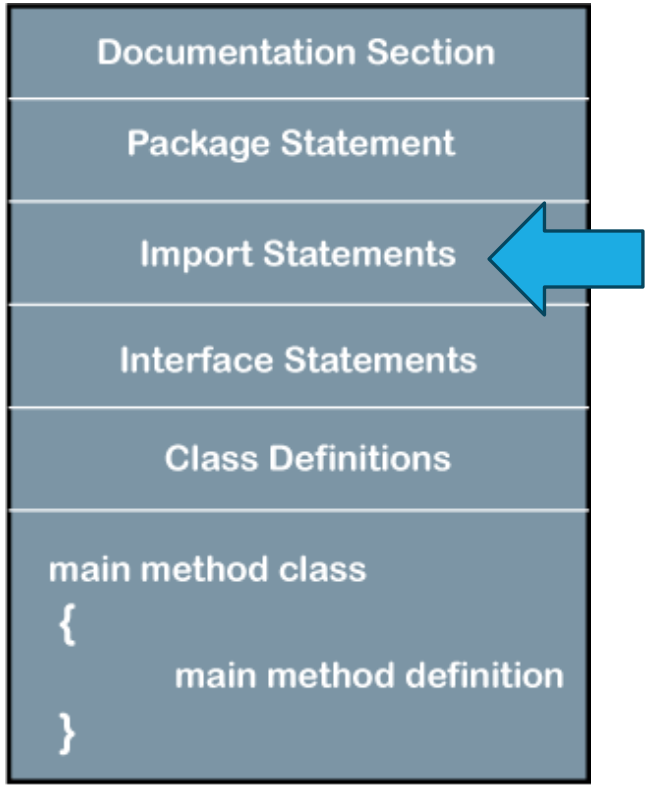
Struttura di un programma Java

Istruzioni di importazione
I pacchetti contengono le numerose classi e interfacce predefinite. Se vogliamo usare una qualsiasi classe di un pacchetto particolare, dobbiamo importare quella classe.

Utilizziamo la parola chiave import per importare la classe. È scritta prima della dichiarazione della classe e dopo l'istruzione del pacchetto.

Utilizziamo l'istruzione di importazione in due modi, importando una classe specifica o importando tutte le classi di un pacchetto particolare. Ad esempio:

```
import java.util.Scanner;
import java.io.*;
```



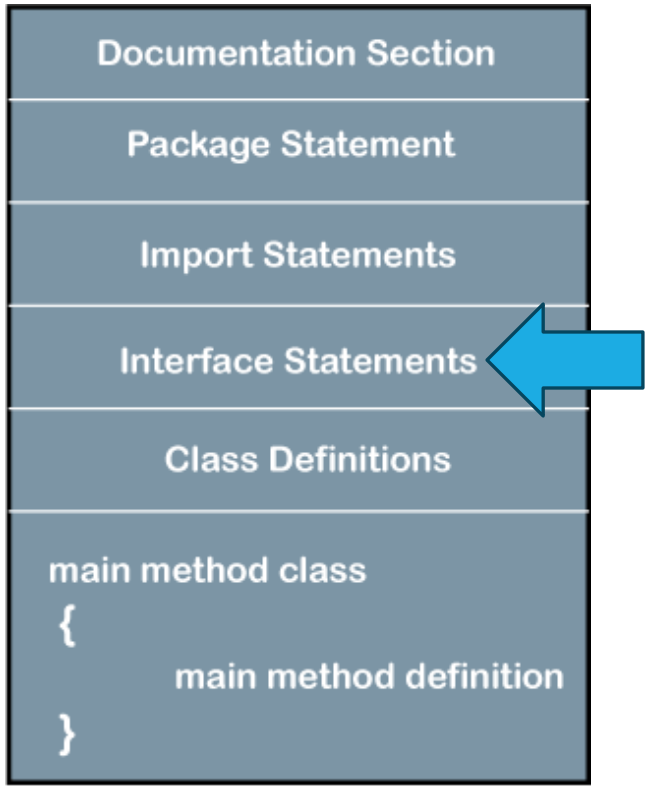
Structure of Java Program

Struttura di un programma Java

Sezione interfaccia

È una sezione facoltativa. Possiamo creare un'interfaccia in questa sezione se necessario. Utilizziamo la parola chiave interface per creare un'interfaccia.

Un'interfaccia è leggermente diversa dalla classe. Contiene solo costanti e dichiarazioni di metodi. Vedremo i dettagli molto più avanti.



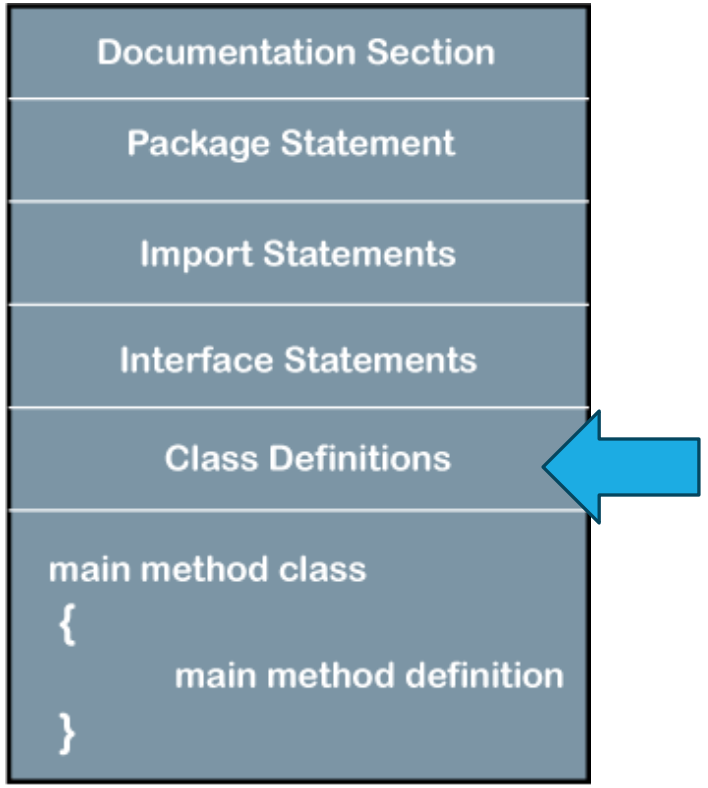
Structure of Java Program

Struttura di un programma Java

Definizione di classe
In questa sezione, definiamo la classe. È una parte fondamentale di un programma Java. Senza la classe, non possiamo creare alcun programma Java.

Un programma Java può contenere più di una definizione di classe. Utilizziamo la parola chiave dass per definire la classe. La classe è un modello di un programma Java. Contiene informazioni su metodi, variabili e costanti definiti dall'utente.

Ogni programma Java ha almeno una classe che contiene il metodo main()



Structure of Java Program

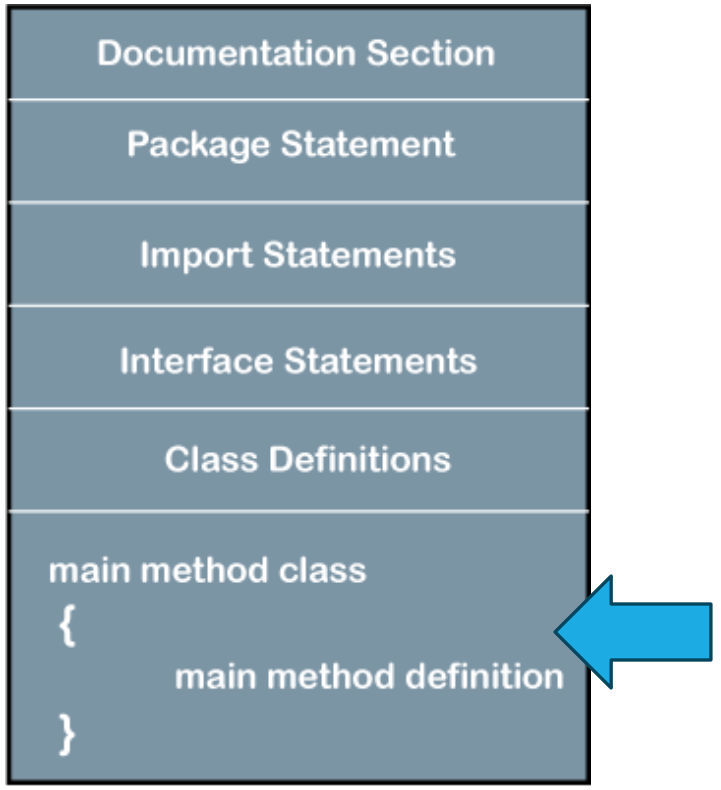
Struttura di un programma Java

Definizione del metodo main
In questa sezione, definiamo il metodo main(). È essenziale per tutti i programmi Java. Poiché l'esecuzione di tutti i programmi Java inizia dal metodo main().

In altre parole, è un punto di ingresso della classe. Deve essere all'interno della classe.

Utilizziamo la seguente istruzione per definire il metodo main():

```
public static void main(String[] args) {
    // istruzioni da eseguire
}
```



Structure of Java Program

Tipi di commenti

Commento su una sola riga: inizia con una coppia di barre oblique (//). Ad esempio:

```
// Primo programma Java
```

Commento su più righe: inizia con /*e termina con */. Scriviamo tra questi due simboli. Ad esempio:

```
/* È un esempio di commento su più righe */
```

Commento di documentazione: inizia con il delimitatore (/**) e termina con */.
Ad esempio:

```
/** È un esempio di commento di documentazione */
```

Istruzioni di output

Per stampare in output un argomento, esiste l'istruzione:

System.out.println(<argomento>);

Dove l'argomento può essere una variabile o un valore. Notare che dopo avere eseguito la stampa il cursore si sposta sulla riga successiva.

```
System.out.println("Ciao da main -- prima riga");
System.out.println("Ciao da main -- seconda riga");
System.out.println();
```

```
Ciao da main -- prima riga
Ciao da main -- seconda riga
```

Esiste anche l'istruzione che non sposta il cursore a capo:

System.out.print(<argomento>);

```
System.out.print("Ciao da main --");
System.out.print(" terza riga");
```

```
Ciao da main -- terza riga
```

Variabili

Come in C++ una variabile è una locazione di memoria identificata da un nome univoco a cui è associato un tipo ed un valore.

La sintassi (uguale a quella in C++) è:

<tipo> <nome>
<tipo> <nome> = <valore>

Ad esempio

```
int x; // dichiarazione di variabile
x = 50; // assegnazione di variabile

float vFloat = 3.3F; // dichiarazione + assegnazione
double vDouble = 5.5;
```

```
boolean vBoolean = true;
byte vByte = 70;
short vShort = 2000;
int vInt = 10045;
long vLong = 445_324_324_564L;
float vFloat = 5.24234F;
double vDouble = 65.8342424554;
char vChar = 'A';
String vString = "Sono una stringa";
```

Tipi di variabili

In Java esistono due tipologie di tipi: primitivi e riferimenti ad oggetti.

Nei tipi primitivi la variabile è direttamente accessibile ed è possibile manipolare direttamente il suo valore. In Java il nome di questi tipi inizia con una lettera minuscola.

Nei tipi con riferimenti ad oggetti la variabile non è direttamente accessibile e per modificarla bisogna utilizzare delle funzioni particolari (denominate anche metodi). In Java il nome di questi tipi inizia con una lettera maiuscola.

Tipo primitivo	Tipo riferito ad un oggetto
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
N.D.	String

Tipi di variabili

	Valori	Dimensione	Tipologia
boolean	true | false	1 bit	Primitivo
byte	-128 a 127	1 byte	Primitivo
short	-32768 a 32767	2 byte	Primitivo
int	-2 miliardi a 2 miliardi	4 byte	Primitivo
long	-9 quintilioni a 9 quintilioni	8 byte	Primitivo
float	numeri con 6-7 cifre decimali	4 byte	Primitivo
double	numeri con 14-15 cifre decimali	8 byte	Primitivo
char	singolo carattere	2 byte	Primitivo
String	sequenza di caratteri	variabile	Riferimento

Costanti

Come suggerisce il nome, una costante è un'entità nella programmazione che è immutabile. In altre parole, il valore non può essere cambiato.

Per dichiarare una variabile come costante, utilizziamo il modificatore final, noto come modificatore di non-accesso. Secondo la convenzione di denominazione Java, il nome dell'identificatore deve essere in lettere maiuscole.

```
final int N = 10;
final String PAROLA = "Java";
final float PI = 3.14f;
```

Istruzioni di input

Per leggere in input il valore di una variabile è necessario includere la libreria che contiene la definizione della classe Scanner

Per importare una libreria è sufficiente adottare la seguente sintassi:

import <percorso della libreria come pacchetti>

```
12  import java.util.Scanner;   // importo librerie, un po' come #include di C++
```

Istruzioni di input

Dopo avere importato la libreria necessaria è necessario definire l'oggetto Scannernel seguente modo:

```
Scanner sc = new Scanner(System.in);     // al momento fidatevi della seguente sintassi
                                         // più avanti sarà tutto più chiaro
```

Ed iniziare ad utilizzare i metodi di tale oggetto mediante le istruzioni:

- nextLine() per leggere una riga intera
- nextInt() per leggere un intero
- nextDouble() per leggere un double
- next().charAt(0) per leggere un carattere
- etc.

Istruzioni di input

Nel seguente esempio chiediamo all'utente tre informazioni: nome, età e cognome.
Mostriamo in output i valori immessi.

```
12  import java.util.Scanner;   // importo librerie, un po' come #include di C++
13
14  public class Es4 {
15
16      public static void main(String[] args) {
17          Scanner sc = new Scanner(System.in);
18
19          System.out.print("Inserire il tuo nome: ");
20          String nome = sc.nextLine();    // leggi la prossima riga come stringa
21
22          System.out.print("Inserire la tua eta': ");
23          int eta = sc.nextInt();     // leggi un numero intero
24          sc.nextLine();              // sc.nextInt() non manda il cursore a capo
25
26          System.out.print("Inserire il tuo cognome: ");
27          String cognome = sc.nextLine(); // leggi la prossima riga come stringa
28
29          System.out.println("Hai inserito il nome " + nome);
30          System.out.println("Hai inserito il cognome " + cognome);
31          System.out.println("Hai " + eta + " anni");
32
33      }
34  }
```

Notare l'istruzione sc.nextLine() dopo sc.nextInt()

Informazioni sull'utilizzo dei materiali didattici

Queste slides si basano su materiali originariamente elaborati dal **Prof. Andrea Melioli**, opportunamente modificati e integrati secondo specifiche esigenze riguardanti la programmazione disciplinare.

L'autore autorizza al **prof. Mario Perna** l'utilizzo, la modifica, la pubblicazione e qualsiasi altra forma di operazione sui materiali a scopo didattico e formativo.

L'uso o la diffusione di questi materiali è **vietato** senza il preventivo contatto con il proprietario del documento, Prof. Mario Perna (prof.mario.perna@darzo.net).