

Concetti base di Java

- tipi di dato in Java
- la memoria
- uso di classi di libreria (I/O e GUI)
- aritmetica
- decisioni

Tipi di dato

- Nome che indica l'insieme di valori che una variabile, o il risultato di un'espressione, possono assumere, e le operazioni che su tali valori si possono effettuare
- Dire per esempio che la variabile X è di tipo "numero intero" significa affermare che X può assumere come valori solo numeri interi (appartenenti ad un certo intervallo) e che su tali valori sono ammesse solo certe operazioni (es. le operazioni aritmetiche elementari)
- Ogni linguaggio di programmazione:
 - consente di usare un certo numero di **tipi di dati predefiniti** di uso generale
 - fornisce di solito un certo insieme di strumenti per definire **nuovi tipi** sulla base delle necessità specifiche di un programma

Tipi di dato primitivi

- **valori di verità, risultati di test**
 - **boolean** $x = \text{true}$;
 - **boolean** $x = (5 > 7)$;
- **caratteri**
 - **char** $x = \text{'d'}$, $y = \text{'5'}$, $z = \text{'!'}$;
- **numeri interi**
 - **short** $x = 120$; (16 bit: da -2^{15} a $2^{15} - 1$)
 - **int** $x = -12000000$ (32 bit: da -2^{31} a $2^{31} - 1$)
 - **long** $x = 12300000000$ (64 bit: da -2^{63} a $2^{63} - 1$)
- **numeri reali**
 - **float** $x = -4.67$; (32 bit: 1 segno, 8 esponente, 23 mantissa, min pos = 1.17×10^{-38} , max pos = 3.40×10^{38})
 - **double** $x = +1.3e12$ (64 bit: ...)

Tipi di dato non primitivi

Chiamati anche “tipi Riferimento”

- Ogni classe definisce un nuovo tipo
 - classi di libreria
 - classi definite dall'utente
- Valore speciale: **null**

(chiariremo il concetto più avanti)

Somma di interi

(1)

- Il secondo programma Java
- Progettiamo un programma Java che:
 - Prende in input due numeri interi inseriti dall'utente attraverso la tastiera
 - Calcola la somma dei due numeri interi
 - Visualizza il risultato della somma

Somma di interi

(2)

Addition.java

```
1 // Addition.java: programma per la somma di due numeri interi
2
3 import java.util.Scanner; // importa la classe Scanner
4
5 public class Addition {
6     public static void main( String args[] )
7     {
8         Scanner input = new Scanner(System.in);
9
10        int number1; // primo numero da sommare
11        int number2; // secondo numero da sommare
12        int sum; // somma
13
14        System.out.print("Inserisci il primo numero intero: ")
15        number1 = input.nextInt();
16
17        System.out.print("Inserisci il secondo numero intero: ")
18        number2 = input.nextInt();
19
20        sum = number1 + number2;
21
22        System.out.printf("La somma e' %d\n", sum);
23    }
24 }
25
```

Importa la classe Scanner dal package java.util

Inizia la classe Addition

Dichiara e inizializza la variabile **input**, di tipo Scanner

Dichiara le variabili **number1**, **number2** e **sum** di tipo int

Legge un intero da tastiera e lo assegna a **number1**

Legge un intero da tastiera e lo assegna a **number2**

Calcola la somma di **number1** e **number2** e assegna il risultato a **sum**

Visualizza la somma usando output formattato (printf)

Somma di interi

(3)

```
1 // Addition.java: programma per la somma di due numeri interi
2
3 import java.util.Scanner; // importa la classe Scanner
4
5 public class Addition {
6     public static void main( String args[] )
7     {
8         Scanner input = new Scanner(System.in);
9
10        int number1; // primo numero da sommare
11        int number2; // secondo numero da sommare
12        int sum; // somma
13
14        System.out.print("Inserisci il primo numero intero: ")
15        number1 = input.nextInt();
16
17        System.out.print("Inserisci il secondo numero intero: ")
18        number2 = input.nextInt();
19
20        sum = number1 + number2;
21
22        System.out.printf("La somma e' %d\n", sum);
23    }
24 }
25
```

Usa il metodo **System.out.print()** per visualizzare un messaggio. **System.out** è un oggetto speciale chiamato *output standard* che permette di visualizzare stringhe di caratteri nella finestra dei comandi

Usa il metodo **System.out.printf()** per la stampa del risultato in modo formattato. Il metodo ha 2 parametri separati da virgole:
• "La somma e' %d\n", dove %d è uno specificatore di formato che indica che il metodo si aspetta un intero (d sta per decimal integer), e \n è il carattere di escape che indica al cursore di posizionarsi all'inizio della linea successiva
• **sum** è il numero intero richiesto

Somma di interi

(4)

```
1 // Addition.java: programma per la somma di due numeri interi
2
3 import java.util.Scanner; // importa la classe Scanner
4
5 public class Addition {
6     public static void main( String args[] )
7     {
8         Scanner input = new Scanner(System.in);
9
10        int number1; // primo numero da sommare
11        int number2; // secondo numero da sommare
12        int sum; // somma
13
14        System.out.print("Inserisci il primo numero intero: ")
15        number1 = input.nextInt();
16
17        System.out.print("Inserisci il secondo numero intero: ")
18        number2 = input.nextInt();
19
20        sum = number1 + number2;
21
22        System.out.printf("La somma e' %d\n", sum);
23    }
24 }
25
```

Istruzioni per la dichiarazione di variabili che specificano il **nome** e il **tipo** di variabili usate nel programma

Dichiara e *inizializza* una variabile **input**, che è di tipo Scanner. Un oggetto di tipo Scanner permette ad un programma di leggere dati che possono provenire da diverse sorgenti (nel caso specifico dalla tastiera)

Questa espressione crea un nuovo (**new**) oggetto di tipo Scanner attraverso l'oggetto standard per l'input **System.in**. Il risultato dell'espressione viene assegnato alla variabile **input** tramite l'operatore di assegnazione **=**

Somma di interi

(5)

```
1 // Addition.java: programma per la somma di due numeri interi
2
3 import java.util.Scanner; // importa la classe Scanner
4
5 public class Addition {
6     public static void main( String args[] )
7     {
8         Scanner input = new Scanner(System.in);
9
10        int number1; // primo numero da sommare
11        int number2; // secondo numero da sommare
12        int sum; // somma
13
14        System.out.print("Inserisci il primo numero intero: ");
15        number1 = input.nextInt();
16
17        System.out.print("Inserisci il secondo numero intero: ");
18        number2 = input.nextInt();
19
20        sum = number1 + number2;
21
22        System.out.printf("La somma e' %d\n", sum);
23    }
24 }
25
```

Istruzione di assegnazione:
usa il metodo **nextInt()** di Scanner che restituisce il numero intero digitato dall'utente. Il risultato della chiamata al metodo **nextInt()** viene memorizzato nella variabile **number1** tramite l'operatore di assegnazione **=**

Istruzione di assegnazione:
calcola la somma delle variabile **number1** e **number2** e assegna il risultato alla variabile **sum** tramite l'operatore di assegnazione **=**

La memoria

(1)

- Variabili
 - Ogni variabile all'interno di un programma ha un **nome**, un **tipo**, una **dimensione** e un **valore**
 - Il nome della variabile corrisponde ad una locazione di memoria del computer
 - Quando un valore viene posizionato in una variabile, esso rimpiazza (e distrugge) il valore precedente
 - Al contrario la lettura di variabili dalla memoria non altera il loro contenuto

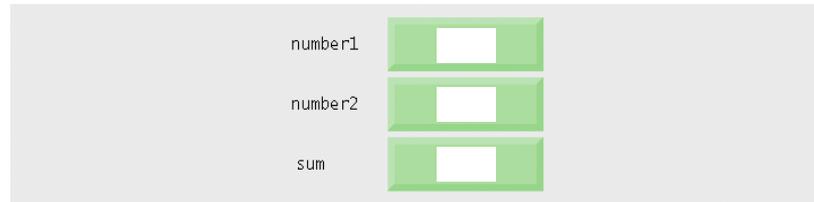
La memoria

(2)

Nel programma Addition.java, dopo la dichiarazione delle variabili

```
int number1;  
int number2;  
int sum;
```

una parte della memoria (RAM) può essere convenientemente rappresentata come segue:



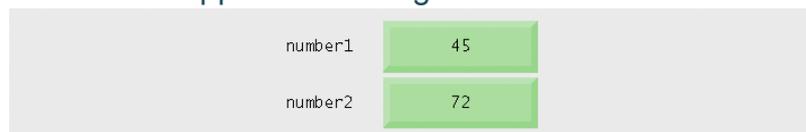
La memoria

(3)

Quando viene eseguita l'istruzione `number1 = input.nextInt();` il numero digitato dall'utente viene inserito nella locazione di memoria associata a `number1`



Quando viene eseguita l'istruzione `number2 = input.nextInt();` la memoria appare come segue



La memoria

(4)

Quando viene eseguita l'istruzione `sum = number1 + number2`; il valore 117 ($45 + 72$) viene memorizzato nella locazione di memoria associata alla variabile `sum` e la memoria appare come segue

| | |
|---------|-----|
| number1 | 45 |
| number2 | 72 |
| sum | 117 |

Aritmetica

(1)

La maggior parte dei programmi esegue calcoli aritmetici usando i seguenti operatori

| Java operation | Arithmetic operator | Algebraic expression | Java expression |
|----------------|---------------------|--|--|
| Addition | + | $f + 7$ | <code>f + 7</code> |
| Subtraction | - | $p - c$ | <code>p - c</code> |
| Multiplication | * | Bm | <code>b * m</code> |
| Division | / | x / y or $\frac{x}{y}$ or $x \div y$ | <code>x / y</code> → <code>7 / 4 = 1</code> |
| Remainder | % | $r \text{ mod } s$ | <code>r % s</code> → <code>17 % 5 = 2</code> |

Aritmetica

(2)

Java prende in considerazione gli operatori presenti in una espressione algebrica secondo le seguenti *regole di precedenza per gli operatori*

| Operator(s) | Operation(s) | Order of evaluation (precedence) |
|-------------|----------------|--|
| * | Multiplication | Evaluated first. If there are several operators of this type, they are evaluated from left to right. |
| / | Division | |
| % | Remainder | |
| + | Addition | Evaluated next. If there are several operators of this type, they are evaluated from left to right. |
| - | Subtraction | |

Per alterare l'ordine basta usare le parentesi come nelle normali espressioni algebriche

Nell'espressione $(a + b) * c$ la parte tra parentesi viene valutata per prima

Aritmetica

(3)

■ Esempi

- Trovare la media di tre variabili a , b e c

- Non usare: $a + b + c / 3$

- Usa: $(a + b + c) / 3$

- Algebra: $z = pr\%q+w/x-y$

Java: $z = p * r \% q + w / x - y$
⑥ ① ② ④ ③ ⑤

- Algebra: $y = ax^2 + bx + c$

Java: $y = a * x * x + b * x + c$
⑥ ① ② ④ ③ ⑤

Aritmetica

(4)

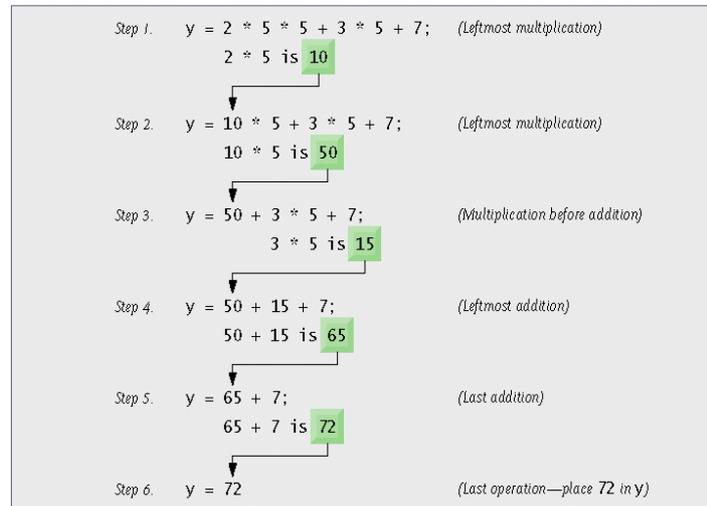
$$y = ax^2 + bx + c$$

$$a = 2$$

$$x = 5$$

$$b = 3$$

$$c = 7$$



String

(1)

- Classe di libreria che modella sequenze di caratteri
- Automaticamente disponibile in ogni programma Java
- Molti metodi a disposizione per la manipolazione di tali sequenze

String

(2)

```
TestString.java
1  /* Test della classe String */
2
3  public class TestString
4  {
5      public static void main( String args[] )
6      {
7          String s1 = new String( "ciao " );
8          String s2 = "mondo!";
9
10         String s3 = s1.concat( s2 );
11         System.out.println( s3.length( ) );
12         System.out.println( s3.endsWith( "o!" ) );
13         System.out.println( s3.replace( "o", "x" ) );
14         System.out.println( s3.substring( 3, 6 ) );
15         System.out.println( s3.toUpperCase( ) );
16
17     } // fine main
18 } // fine classe TestString
```

equivalente a:
String s3 = s1 + s2;

questi metodi restituiscono nuove stringhe

Visualizzare testo in Dialog Box

- Finestre e dialog box
 - Molte applicazioni Java usano questi box per visualizzare messaggi di input e output
 - **JOptionPane** è la classe di libreria che fornisce tali box chiamati "message dialogs"
 - **JOptionPane** è contenuta nel package **javax.swing**

Esempio

```
Dialog.java
1 // Dialog.java
2 // Printing multiple lines in dialog box.
3
4 import javax.swing.JOptionPane; // import classe JOptionPane
5
6 public class Dialog
7 {
8     public static void main( String args[] )
9     {
10        // visualizza un messaggio in una finestra dialog
11        JOptionPane.showMessageDialog( null, "Welcome\nto\nJava" );
12    }
13 } // fine classe Dialog
14
15
```

Import della classe
JOptionPane

Visualizza un messaggio in
una finestra dialog



Prendere decisioni

(1)

- Una **condizione** è una espressione che può assumere due possibili valori, **true** oppure **false**
- Ad esempio: "Se il voto è maggiore o uguale a 18 lo studente passa l'esame"
- Introduciamo una versione semplificata dell'istruzione **if** del linguaggio Java che consente di prendere una decisione in base alla verità o falsità di una condizione
- Se una condizione è true, allora il corpo dell'istruzione **if** viene eseguito, altrimenti viene eseguita l'istruzione successiva
- La condizione dell'istruzione **if** può essere formata usando **operatori di uguaglianza** e **relazionali**

Prendere decisioni

(2)

| Standard algebraic equality or relational operator | Java equality or relational operator | Sample Java condition | Meaning of Java condition |
|--|--------------------------------------|-----------------------|---------------------------------|
| <i>Equality operators</i> | | | |
| = | == | X == Y | X is equal to Y |
| ≠ | != | X != Y | X is not equal to Y |
| <i>Relational operators</i> | | | |
| > | > | X > Y | X is greater than Y |
| < | < | X < Y | X is less than Y |
| ≥ | >= | X >= Y | X is greater than or equal to Y |
| ≤ | <= | X <= Y | X is less than or equal to Y |

Operatori di uguaglianza e relazionali

Prendere decisioni

(3)

```
Comparison.java
1 // Comparison.java: usa if, operatori relazionali e di uguaglianza
2
3 import javax.swing.JOptionPane;
4 public class Comparison {
5     public static void main( String args[] ) {
6         String number1, number2, message = "";
7         int n1, n2;
8
9         number1 = JOptionPane.showInputDialog("Inserisci primo numero: ");
10        n1 = Integer.parseInt(number1);
11
12        number2 = JOptionPane.showInputDialog("Inserisci secondo numero: ");
13        n2 = Integer.parseInt(number2);
14
15        if ( n1 == n2 ) ←
16            message = message + String.format("%d == %d\n", n1, n2);
17
18        if ( n1 != n2 )
19            message = message + String.format("%d != %d\n", n1, n2);
20
21        if ( n1 < n2 ) ←
22            message = message + String.format("%d < %d\n", n1, n2);
23
24        if ( n1 > n2 )
25            message = message + String.format("%d > %d\n", n1, n2);
26
27        if ( n1 <= n2 )
28            message = message + String.format("%d <= %d\n", n1, n2);
29
30        if ( n1 >= n2 )
31            message = message + String.format("%d >= %d\n", n1, n2);
32
33        JOptionPane.showMessageDialog(null, message);
34    }
35 }
```

Test di uguaglianza tra due numeri

Confronto di due numeri tramite operatori relazionali

Prendere decisioni

(4)

```
Inserisci il primo numero intero: 300
Inserisci il primo numero intero: 300
300 == 300
300 <= 300
300 >= 300
```

```
Inserisci il primo numero intero: 500
Inserisci il primo numero intero: 300
500 != 300
500 > 300
500 >= 300
```

```
Inserisci il primo numero intero: 300
Inserisci il primo numero intero: 500
300 != 500
300 < 500
300 <= 500
```

Precedenza e associatività

| Operators | Associativity | Type |
|-----------|---------------|----------------|
| * / % | left to right | multiplicative |
| + - | left to right | additive |
| < <= > >= | left to right | relational |
| == != | left to right | equality |
| = | right to left | assignment |

IDE

- Esistono ambienti di sviluppo integrati (Integrated Development Environment, IDE)
- Mettono a disposizione strumenti per lo sviluppo di programmi e la loro messa a punto
 - editor
 - compilatore e/o interprete
 - debugger

IDE: JCreator

The screenshot shows the JCreator IDE interface. The main window displays a Java file named `GradeBook.java` with the following code:

```
1 | comment  
8 |  
9 |  
10 | public class GradeBook {  
11 | comment  
15 | private String courseName;  
16 |  
17 |  
18 | /**  
19 | * <p> Costruttore del GradeBook </p>  
20 | *  
21 | * @param name Stringa che definisce il nome del corso  
22 | */  
23 | public GradeBook( String name )  
24 | {  
25 |     courseName = name  
26 | }  
27 |  
28 | comment  
29 | public void setCourseName( String name )  
30 | {  
31 |     courseName = name;  
32 | }  
33 |  
34 | comment  
35 | public String getCourseName()  
36 | {  
37 |     return courseName;  
38 | }  
39 |  
40 | comment  
41 | public String getCourseName()  
42 | {  
43 |     return courseName;  
44 | }
```

The interface includes several panels and toolbars:

- comando esecuzione**: A toolbar button for running the program.
- comandi compilazione**: A toolbar button for compiling the code.
- vista File**: A file explorer on the left showing the project structure.
- vista Data**: A data view on the left showing the class structure and methods.
- editor**: The central area where the code is written.
- vista Output**: A panel at the bottom showing the build output, including the message "Process completed."

Fondam
For Help, press F1

28