

Linguaggi di Programmazione I – Lezione 10

Prof. Marcello Sette
mailto://marcello.sette@gmail.com
http://sette.dnsalias.org

8 maggio 2008

Array	3
Dichiarazione	4
Creazione (1)	5
Creazione (2)	6
Inizializzazione (1)	7
Inizializzazione (2)	8
Esempio	9
Multidimensioni	10
Estremi	11
Assegnazione	12
Ridimensionamento	13
Copia	14
Aiutare il GC	15
Esercizi	16
Esercizi	16
Questionario	17
D 1	18
D 2	19
D 3	20
D 4	21
D 5	22
D 6	23
D 7	24
D 8	25
D 9	26
D 10	27
D 11	28
D 12	29
D 13	30
D 14	31
D 15	32
D 16	33
D 17	34
D 18	35

D 19	36
D 20	37
D 21	38
D 22	39
D 23	40

Panoramica della lezione

Array

Esercizi

Questionario

LP1 – Lezione 10

2 / 40

Array

3 / 40

Dichiarazione

- Si possono dichiarare array di tipi primitivi o di riferimenti ad oggetti:

```
char s[];  
Point p[];  
  
char [] s;  
Point [] p;
```

- Un array è un oggetto: le dichiarazioni precedenti creano solo il riferimento al rispettivo oggetto.

- NON viene qui generato l'oggetto: la generazione richiede ancora l'uso di `new`.
- Nella dichiarazione NON si deve specificare la dimensione dell'array.
- Se si dichiarano più array in uno stesso enunciato usando le parentesi quadre a sinistra, le parentesi sono applicate a tutte le variabili alla loro destra.
- Esempio:

```
int a[], b; // a e' riferimento ad array, b e' int  
int [] a, b; // a e b entrambi riferimenti ad array
```

LP1 – Lezione 10

4 / 40

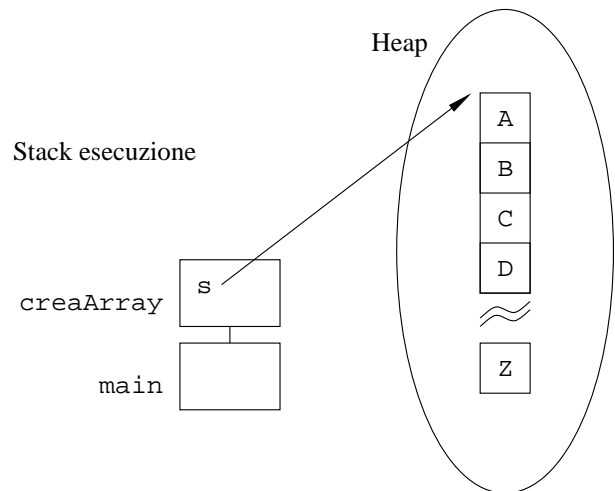
Creazione (1)

- Per creare l'array si deve usare new.
- Esempio:

```
public char[] creaArray() {
    char[] s;

    s = new char[26];
    for (int i=0; i<26; i++) {
        s[i] = (char) ('A' + i);
    }

    return s;
}
```



`s = new char[26];` crea un array di 26 caratteri: essi sono inizializzati al loro valore di default (`'\u0000'` per char). I valori sono accessibili nel range da 0 a 25: ogni tentativo di accesso oltre questo range causerà il lancio di una eccezione a runtime.

LP1 – Lezione 10

5 / 40

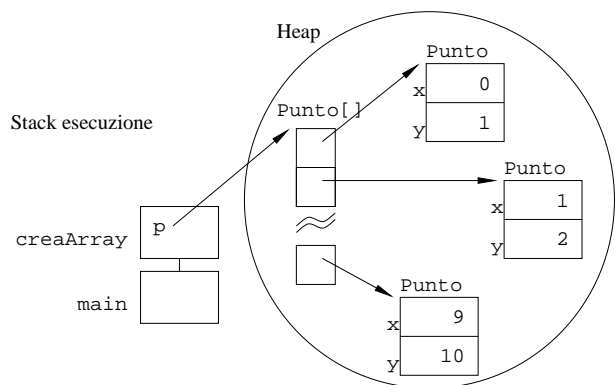
Creazione (2)

- Un altro esempio per array di oggetti:

```
public Punto[] creaArray() {
    Punto[] p;

    p = new Punto[10];
    for ( int i=0; i<10; i++ ) {
        p[i] = new Punto(i, i+1);
    }

    return p;
}
```



`p = new Punto[10];` anche questa volta crea un array di 10 **referimenti** ad oggetti Punto: essi sono inizializzati al loro valore di default null.

Questa volta è cruciale la creazione dei singoli oggetti, prima del loro uso.

Nota: una sintassi alternativa, ma forse meno leggibile, per il metodo precedente poteva essere:

```
public Punto creaArray() []
```

LP1 – Lezione 10

6 / 40

Inizializzazione (1)

Poiché l'inizializzazione delle variabili è cruciale, Java fornisce due metodi abbreviati per gli array (oltre il metodo diretto che utilizza un loop). Il primo consiste nella dichiarazione, costruzione ed inizializzazione in una riga:

```
String [] nomi = {  
    "Antonio",  
    "Marcello",  
    "Anna"  
};
```

equivalente a:

```
String [] nomi;  
nomi = new String[3];  
nomi[0] = "Antonio";  
nomi[1] = "Marcello";  
nomi[2] = "Anna";
```

LP1 – Lezione 10

7 / 40

Inizializzazione (2)

Il secondo metodo è la costruzione ed inizializzazione di un array anonimo. Esempio:

```
int [] esempiol;  
esempiol = new int [] {4,7,3};
```

ma, ancora meglio:

```
public class A {  
    void prendiArray (int [] unArray) {  
        // usa unArray  
    }  
    public static void main (String[] args) {  
        A a = new A();  
        a.prendiArray(new int [] {3,4,5,6,7});  
    }  
}
```

Attenzione: NON si deve specificare la dimensione in una creazione di un array anonimo.

```
new Object[2] {null, new Object()}; // illegale
```

LP1 – Lezione 10

8 / 40

Esempio

In Java tutti i parametri dei metodi sono di tipo IN (realizzati) per copia. Questo significa che il valore del parametro nel metodo chiamante non può essere modificato. Tuttavia, è possibile creare un riferimento a tale parametro e realizzare ugualmente la modifica nel metodo chiamante. Per esempio, per creare un riferimento ad un tipo primitivo:

```
public class PrimitiveReference {
    public static void main(String args[]) {
        int [] mioValore = { 1 };
        modifica(mioValore);
        System.out.println("mioValore contiene " +
            mioValore[0]);
    }
    public static void modifica(int [] valore) {
        valore[0]++;
    }
}
```

LP1 – Lezione 10

9 / 40

Multidimensioni

■ Array di array:

```
int dueDim [][] = new int [3] [];
dueDim[0] = new int [5];
dueDim[1] = new int [5];
dueDim[2] = new int [5];

int dueDim [][] = new int [] [3]; // illegale
```

■ Array non rettangolare di array:

```
int dueDim [][] = new int [3] [];
dueDim[0] = new int [2];
dueDim[1] = new int [5];
dueDim[2] = new int [8];
```

■ Array rettangolare di array:

```
int dueDim [][] = new int [4] [5];
```

■ Costruzione e inizializzazione:

```
int [][] multiD = {{5,4,3,2}, {9,8}, {7,6,5}};
```

LP1 – Lezione 10

10 / 40

Estremi

- Indice iniziale 0.
- Numero di elementi è parte dell'oggetto array, nell'attributo length.
- Esempio:

```
int list[] = new int [10];
for (int i=0; i<list.length; i++) {
    System.out.println(list[i]);
}
```

- Accesso oltre i limiti causa il lancio di una eccezione a runtime.
- Cosa stampa il brano seguente?

```
int m[][] = new int [10][5];
System.out.println("m.length vale " + m.length);
System.out.println("m[0].length vale " + m[0].length);
```

LP1 – Lezione 10

11 / 40

Assegnazione

- In una assegnazione l'array deve avere lo stesso numero di dimensioni della variabile di riferimento.
- Per esempio:

```
int [] vettore;
int [][] matrice = new int [3] [];

vettore = matrice; // illegale

int [] v = new int [6];
vettore = v; // OK
```

LP1 – Lezione 10

12 / 40

Ridimensionamento

- Gli array NON sono ridimensionabili.
- Si può usare la stessa variabile per riferirsi ad un nuovo array:

```
int elements[] = new int [6];
elements = new int [10];
```

- Che ne è del vecchio array?

LP1 – Lezione 10

13 / 40

Copia

Il metodo `System.arraycopy()`:

```
// array originale
int vecchi[] = {1, 2, 3, 4};

// nuovo array piu' lungo
int nuovo[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

// copia tutti i vecchi elementi nel nuovo array
System.arraycopy(vecchi, 0, nuovo, 0, vecchi.length);
```

Attenzione: il metodo copia i valori contenuti negli elementi dell'array. Nel caso di array di oggetti (o di array multidimensionali), ciò significa che vengono copiati i riferimenti agli oggetti, non vengono cioè create nuove copie di oggetti.

LP1 – Lezione 10

14 / 40

Aiutare il GC

Il garbage collector non è onnipotente. Alcune volte ha bisogno di un piccolo “aiuto”. Per esempio, supponiamo di dover operare su una pila usando il metodo:

```
public Object pop() {
    return pila[indice--];
}
```

Se l'utilizzatore del metodo abbandona il valore di ritorno ricevuto, esso non sarà eleggibile per GC, finché il riferimento ad esso nell'array `pila` non sarà sovrascritto. Questo potrebbe richiedere tempi lunghissimi. Più correttamente:

```
public Object pop() {
    Object valore = pila[indice];
    pila[indice--] = null;
    return valore;
}
```

LP1 – Lezione 10

15 / 40

Esercizi

16 / 40

Esercizi

1. Scrivere un programma che stampi i valori degli argomenti della linea di comando.
2. Estendere l'esercizio della lezione precedente, usando gli array per realizzare la molteplicità nella relazione (composizione) tra una banca e i propri clienti.

LP1 – Lezione 10

16 / 40

D 1

Un tipo di dato con segno ha un ugual numero di valori positivi e negativi.

- A. Vero
- B. Falso

LP1 – Lezione 10

18 / 40

D 2

Scegliere gli identificatori legali tra questi:

- A. StringaLunghissimaSenzaSignificato
- B. \$int
- C. bytes
- D. \$1
- E. finals

LP1 – Lezione 10

19 / 40

D 3

Quali delle seguenti segnature sono valide per il metodo main?

- A. `public static void main()`
- B. `public static void main(String arg[])`
- C. `public void main(String[] arg)`
- D. `public static void main(String[] args)`
- E. `public static int main(String[] arg)`

LP1 – Lezione 10

20 / 40

D 4

Se in un file sorgente sono presenti tutti e tre gli elementi top-level, in quale ordine devono apparire?

- A. `import`, `package`, `class`.
- B. `class`, `import`, `package`.
- C. `package` per primo, l'ordine degli altri non importa.
- D. `package`, `import`, `class`.
- E. `import` per primo, l'ordine degli altri non importa.

LP1 – Lezione 10

21 / 40

D 5

Si consideri la seguente linea di codice:

```
int[] x = new int[25];
```

Dopo l'esecuzione, quali delle seguenti affermazioni sono vere?

- A. x[0] è 0.
- B. x è indefinito.
- C. x è 0.
- D. x[0] è null.
- E. x.length è 25.

LP1 – Lezione 10

22 / 40

D 6

Qual è l'output della seguente applicazione:

```
class D6 {
    public static void main(String args[]) {
        Scatola s = new Scatola();
        s.interno = 100;
        s.aumenta(s);
        System.out.println(s.interno);
    }
}

class Scatola {
    public int interno;
    public void aumenta(Scatola scatola) {
        scatola.interno++;
    }
}
```

- A. 0 B. 1 C. 100 D. 101

LP1 – Lezione 10

23 / 40

D 7

Qual è l'output della seguente applicazione:

```
class D7 {
    public static void main(String args[]) {
        double d = 12.3;
        Decremento dec = new Decremento();
        dec.decrementa(d);
        System.out.println(d);
    }
}

class Decremento {
    public void decrementa(double dec) {
        dec = dec - 1.0;
    }
}
```

- A.** 0.0 **B.** -1.0 **C.** 12.3 **D.** 11.3

LP1 – Lezione 10

24 / 40

D 8

Come si può forzare la garbage collection di un oggetto?

- A.** Il garbage collector non può essere forzato.
B. Con una chiamata a `System.gc()`.
C. Con una chiamata a `System.gc()`, passando il riferimento all'oggetto.
D. Con una chiamata a `Runtime.gc()`.
E. Ponendo tutti i riferimenti a quell'oggetto a `null`.

LP1 – Lezione 10

25 / 40

D 9

Qual è il range di valori per una variabile di tipo `short`?

- A.** Dipende dall'hardware che ospita la JVM.
B. $0 \dots 2^{16} - 1$
C. $0 \dots 2^{32} - 1$
D. $-2^{15} \dots 2^{15} - 1$
E. $-2^{31} \dots 2^{31} - 1$

LP1 – Lezione 10

26 / 40

D 10

Qual è il range di valori per una variabile di tipo byte?

- A. Dipende dall'hardware che ospita la JVM.
- B. $0 \dots 2^8 - 1$
- C. $0 \dots 2^{16} - 1$
- D. $-2^7 \dots 2^7 - 1$
- E. $-2^{15} \dots 2^{15} - 1$

LP1 – Lezione 10

27 / 40

D 11

Quali sono i valori di x, a, b dopo l'esecuzione del codice:

```
int x, a = 6, b = 7;  
x = a++ + b++;
```

- A. x=15, a=7, b=8
- B. x=15, a=6, b=7
- C. x=13, a=7, b=8
- D. x=13, a=6, b=7

LP1 – Lezione 10

28 / 40

D 12

Quali delle seguenti espressioni sono legali?

- A. `int x=6; x=!x;`
- B. `int x=6; if (!(x>3)) {}`
- C. `int x=6; x=~x;`

LP1 – Lezione 10

29 / 40

D 13

Quali delle seguenti espressioni risultano in un valore positivo in x?

- A. `int x=-1; x = x >>> 5;`
- B. `int x=-1; x = x >>> 32;`
- C. `byte x=-1; x = x >>> 5;`
- D. `int x=-1; x = x >> 5;`

LP1 – Lezione 10

30 / 40

D 14

Quali delle seguenti espressioni sono legali?

- A. `String x="Ciao"; int y=7; x += y;`
- B. `String x="Ciao"; int y=7; if (x == y) {}`
- C. `String x="Ciao"; int y=7; x = x + y;`
- D. `String x="Ciao"; int y=7; y = y + x;`
- E. `String x=null;`
`int y = (x!=null) &&`
`(x.length()>0) ? x.length() : 0;`

LP1 – Lezione 10

31 / 40

D 15

Qual è il risultato dell'esecuzione del seguente codice?

```
public class Xor {  
    public static void main(String args[]) {  
        byte b = 10; // 00001010 binario  
        byte c = 15; // 00001111 binario  
        b = (byte)(b ^ c);  
        System.out.println("b vale " + b);  
    }  
}
```

- A. b vale 10
- B. b vale 5
- C. b vale 250
- D. b vale 245

LP1 – Lezione 10

32 / 40

D 16

Qual è il risultato della compilazione ed esecuzione del seguente codice?

```
1. public class Condizionale {  
2.     public static void main(String args[]) {  
3.         int x = 4;  
4.         System.out.println("Il valore e' " +  
5.             ((x > 4) ? 99.99 : 9));  
6.     }  
7. }
```

- A. Il valore e' 99.99
- B. Il valore e' 9
- C. Il valore e' 9.0
- D. Un errore di compilazione alla linea 5

LP1 – Lezione 10

33 / 40

D 17

Qual è l'output del seguente frammento di codice?

```
int x=3; int y=-10;
System.out.println(y % x);
```

- A. 0
- B. 1
- C. -1
- D. -3

LP1 – Lezione 10

34 / 40

D 18

Qual è l'output del seguente frammento di codice?

```
int x=1;
String [] nomi = {"Mario", "Anna", "Carlo"};
nomi[--x] += ".";
for (int i=0; i < nomi.length; i++) {
    System.out.println(nomi[i]);
}
```

- A. L'output include Mario. con un punto finale.
- B. L'output include Anna. con un punto finale.
- C. L'output include Carlo. con un punto finale.
- D. Nessun nome stampato ha il punto finale.
- E. Viene lanciata l'eccezione `ArrayIndexOutOfBoundsException`.

LP1 – Lezione 10

35 / 40

D 19

Quali linee fanno parte dell'output del seguente codice?

```
for (int i=0; i<2; i++) {
    for (int j=0; j<3; j++) {
        if (i == j) {
            continue;
        }
        System.out.println("i=" + i + " j=" + j);
    }
}
```

- A. i=0 j=0
- B. i=0 j=1
- C. i=0 j=2
- D. i=1 j=0
- E. i=1 j=1
- F. i=1 j=2

LP1 – Lezione 10

36 / 40

D 20

Quali linee fanno parte dell'output del seguente codice?

```
esterno: for (int i=0; i<2; i++) {  
    for (int j=0; j<3; j++) {  
        if (i == j) {  
            continue esterno;  
        }  
        System.out.println("i=" + i + " j=" + j);  
    }  
}
```

- A. i=0 j=0
- B. i=0 j=1
- C. i=0 j=2
- D. i=1 j=0
- E. i=1 j=1
- F. i=1 j=2

LP1 – Lezione 10

37 / 40

D 21

Quali tra queste sono costruzioni legali di loop?

A.

```
while (int i<7) {  
    i++;  
    System.out.println("i=" + i);  
}
```

B.

```
int i=3;  
while (i) {  
    System.out.println("i=" + i);  
}
```

C.

```
int j=0;  
for (int k=0; j + k != 10;  
    j++, k++) {  
    System.out.println(  
        "j=" + j + " k=" + k);  
}
```

D.

```
int j=0;  
do {  
    System.out.println(  
        "j=" + j++);  
    if (j == 3) {continue loop;}  
} while (j < 10);
```

LP1 – Lezione 10

38 / 40

D 22

Qual è l'output di questo frammento di codice?

```
int x=0, y=4, z=5;
if ( x>2 ) {
    if ( y<5 ) {
        System.out.println("uno");
    } else {
        System.out.println("due");
    }
} else if ( z>5 ) {
    System.out.println("tre");
} else {
    System.out.println("quattro");
}
```

- A. uno
- B. due
- C. tre
- D. quattro

LP1 – Lezione 10

39 / 40

D 23

Dato il codice:

```
1. int j=2;
2. switch (j) {
3.     case 2:
4.         System.out.print("2");
5.     case 2+1:
6.         System.out.print("3");
7.         break;
8.     default:
9.         System.out.print(j);
10.        break;
11. }
12. System.out.println();
```

Quale dei seguenti enunciati è vero?

- A. Il codice è illegale a causa dell'espressione alla linea 5.
- B. I tipi accettabili per la variabile di controllo di uno switch sono byte, short, int, long.
- C. L'output è 2.
- D. L'output è 23.
- E. L'output è 232.

LP1 – Lezione 10

40 / 40