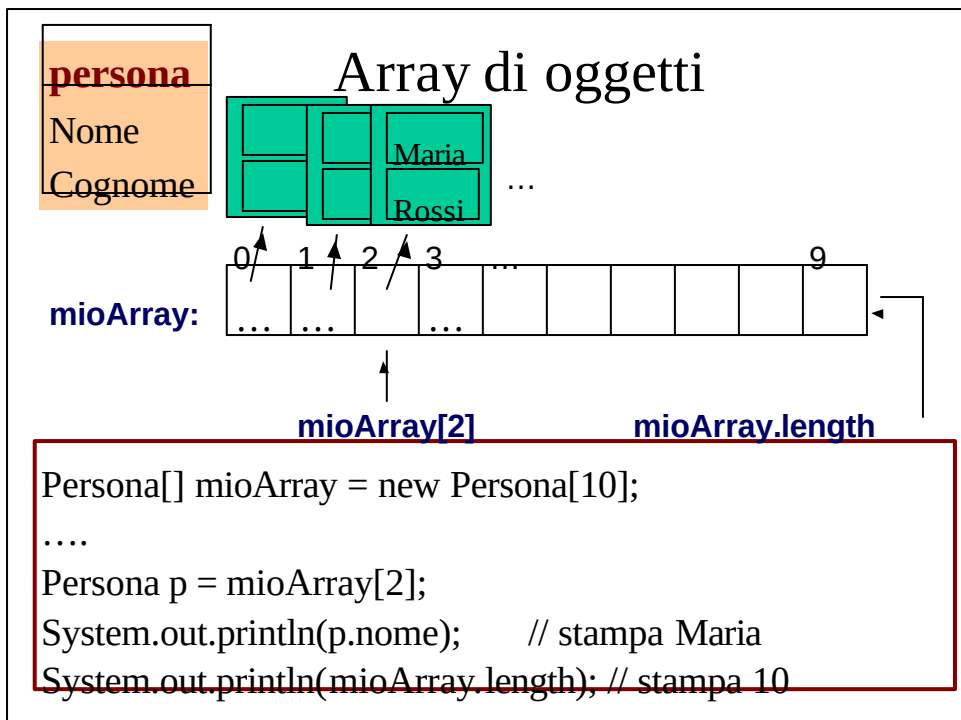


Array

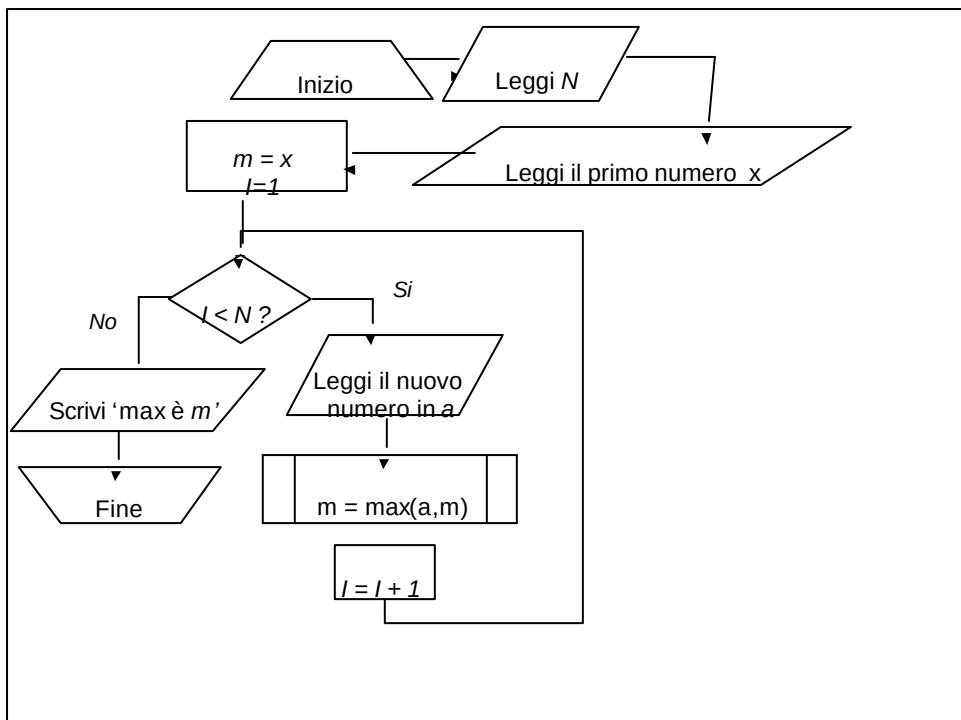
Lucidi aggiunti



Determinare il massimo fra N numeri interi

Algoritmo max_N

1. Leggi il valore di N
2. Leggi il primo numero
3. Assegna al massimo m il valore del primo numero letto
4. Finchè (hai esaminato meno di N numeri)
 - a. Leggi un nuovo numero x
 - b. Trova il maggiore fra m e x
 - c. Assegna il valore del maggiore a m
5. Stampa 'il massimo è...' ed il valore di m e termina



```

class Max_N{
    public static void main(String args[]){
        InterfacciaConsole f = new InterfacciaConsole();
        int n = f.richiediNumero("quanti numeri vuoi confrontare?");
        int max= f.richiediNumero("inserisci il numero");
        int x;
        for(int k=1;k<n;k++){
            x= f.richiediNumero("inserisci il numero");
            if(max<x) max=x;
            //max=max<x?x:max;
        }
        System.out.println("il max tra i numeri immessi è"+ max);
    }
}

```

```

C:\esercizi>java Max_N
quanti numeri vuoi confrontare?
4
inserisci il numero
3
inserisci il numero
5
inserisci il numero
6
inserisci il numero
3
il max tra i numeri immessi e'6

```

```

class Max_N_A{
    public static void main(String args[]){
        InterfacciaConsole f = new InterfacciaConsole();
        int n = f.richiediNumero("quanti numeri vuoi confrontare?");
        int a[]= new int[n];
        for(int k=0;k<n;k++){
            a[k]= f.richiediNumero ("inserisci il numero");
        }
        int max=a[0];
        for(int k=1;k<n;k++){
            if(max<a[k]) max=a[k];
            //max= max<a[k]?a[k]:max;
        }
        System.out.println("il max tra i numeri immessi:");
        for(int k=0;k<n;k++) System.out.println(a[k]);
        System.out.println("e'" + max);
    }
}

```

```

C:\esercizi>java Max_N_A
quanti numeri vuoi confrontare?
5
inserisci il numero
643
inserisci il numero
78
inserisci il numero
567
inserisci il numero
3456
inserisci il numero
77
il max tra i numeri immessi:
643
78
567
3456
77
E' 3456

```

Ordinare un array

```
int temp;  
for(int k=0;k<5;k++){  
    for(int t=0;t<5;t++){  
        if(a[k]>a[t]) {  
            temp=a[k];  
            a[k]=a[t];  
            a[t]=temp;  
        }  
    }  
}
```

Esempio:
Array a in origine:

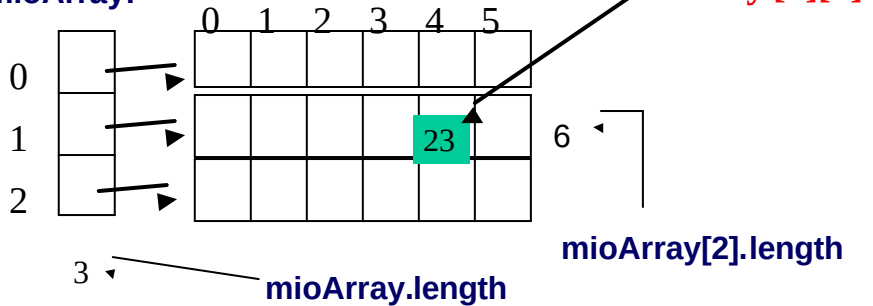
56	77	34	99	78
----	----	----	----	----

Array a dopo l'ordinamento:

99	78	77	56	34
----	----	----	----	----

Array Bidimensionali

mioArray:



```
int mioArray [ ][ ] = new int[3][6];
```

Esempio

```
int am[][]={{3,7,54},{22,45,78}};  
    System.out.println("array.length "+am.length);  
    int p=0;  
    System.out.println("array[1].length"+am[1].length);  
    for(int i=0; i<2; i++){  
        for(int j=0; j<3; j++){  
            System.out.println("array "+i+" "+j+" "+am[i][j]);  
        }  
    }
```

array.length 2	array 0 2 =54
array[1].length 3	array 1 0 =22
array 0 0 =3	array 1 1 =45
array 0 1 =7	array 1 2=78

Argomenti.java

```
class Argomenti{  
    public static void main(String args[]){  
        for(int i=0; i<args.length; i++){  
            System.out.println("args "+i+" "+args[i]);  
        }  
    }  
}
```

```
>Java Argomenti pippo pluto paperino  
args 0 pippo  
args 1 pluto  
args 2 paperino
```

```

class Max_N_Ar1{
    public static void main(String args[]){
        int n = args.length;
        int a[]= new int[n];
        for(int k=0;k<n;k++){
            a[k]= Integer.parseInt(args[k]);
        }
        int max=a[0];
        for(int k=1;k<n;k++){
            if(max<a[k]) max=a[k];
            //max= max<a[k]?a[k]:max;
        }
        System.out.println("il max tra i numeri immessi:");
        for(int k=0;k<n;k++){
            System.out.println(a[k]);
        }
        System.out.println("è "+max);
    }
}

```

Determinazione del
max tra gli
argomenti del
programma

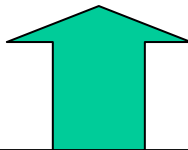
ATTENZIONE

C:\esercizi>java Max_N_Ar

Exception in thread "main"

java.lang.ArrayIndexOutOfBoundsException: 0

at Max_N_Ar.main(Max_N_Ar.java:8)



Se non si digitano numeri separati da spazi l'array args è vuoto e quindi non è possibile accedere ai suoi elementi

Durante l'esecuzione del programma viene visualizzato un errore alla riga 8: int max=a[0];

```

class Max_N_Ar1{
    public static void main(String args[]){
        int n = args.length;
        if (n==0) {
            System.out.println("inserire i numeri da confrontare come argomenti
del programma separati da spazi");
            return;
        }
        int a[]= new int[n];
        for(int k=0;k<n;k++){
            a[k]= Integer.parseInt(args[k]);
        }
        int max=a[0];
        for(int k=1;k<n;k++){
            if(max<a[k]) max=a[k];
            //max= max<a[k]?a[k]:max;
        }
        ....
    }}

```

C:\esercizi>java Max_N_Ar1
 inserire i numeri da confrontare come argomenti del programma separati da spazi

C:\esercizi>java Max_N_Ar1 46 77 3 5 88 9

il max tra i numeri immessi:

46

77

3

5

88

9

e' 88

```
class Ordina_N_Ar1{
    public static void main(String args[]){
        int n = args.length;
        if(n==0) {
            System.out.println("inserire i numeri da ordinare come argomenti");
            return;
        }
        int a[]= new int[n];
        for(int k=0;k<n;k++) a[k]= Integer.parseInt(args[k]);
        int temp;
        for(int k=0;k<n;k++){
            for(int t=0;t<n;t++){
                if(a[k]>a[t]) {temp=a[k];a[k]=a[t];a[t]=temp;}
            }
        }
        System.out.println("ho ordinato i numeri immessi:");
        for(int k=0;k<n;k++) System.out.println(a[k]);
    }
}
```

Esercizi

1. Utilizzare il metodo **charAt** per contare le occorrenze di un carattere in una Stringa

Risultato Esercizio 1

```
class ProvaChar{  
    public static void main(String args[]){  
        String s = args[0];  
        char c=args[1].charAt(0);  
        int count=0;  
        for(int i=0;i<s.length();i++){  
            if(s.charAt(i)==c) count++;  
        }  
        System.out.println("ci sono "+ count+" "+c+" in "+s);  
    }  
}
```

```
C:\>java ProvaChar pappappero p  
ci sono 5 p in pappappero
```

Nota

C:\esercizi1>java ProvaChar

Exception in thread "main"

java.lang.ArrayIndexOutOfBoundsException: 0
at ProvaChar.main(ProvaChar.java:4)

È sempre bene controllare la lunghezza dell'array prima di accedere ad un suo elemento se non si è sicuri delle sue dimensioni

```
class ProvaChar{
    public static void main(String args[]){
        if ( args.length<2)
            System.out.println("manca una stringa e un carattere");
        else { String s = args[0];
            char c=args[1].charAt(0);
            int count=0;
            for(int i=0;i<s.length();i++){
                if(s.charAt(i)==c) count++;
            }
            System.out.println("ci sono "+ count+" "+c+" in "+s);
        }
    }
}
```

```
class ProvaChar3{
    public static void main(String args[]){
        String s = args[0];
        char c=args[1].charAt(0);
        int count=0,i=0;
        while(i<s.length()){
            if(s.charAt(i)==c) count++;
            i++;
        }
        System.out.println("ci sono "+ count+" "+c+" in "+s);
    }
}
```

Esercizi

2. Cercare nella documentazione la classe **Math**

- Utilizzare la classe Math per calcolare la radice quadrata di un double
- Ricordando che un metodo static può essere chiamato senza creare un'istanza dell'oggetto

Risultato Esercizio 2

```
class Prova2{  
    public static void main(String args[]){  
        double x = Math.sqrt(4.0);  
        System.out.println("sqrt "+x);  
    }  
}
```

```
C:\>java ProvaC  
sqrt 2.0
```

Risultato Esercizio 2

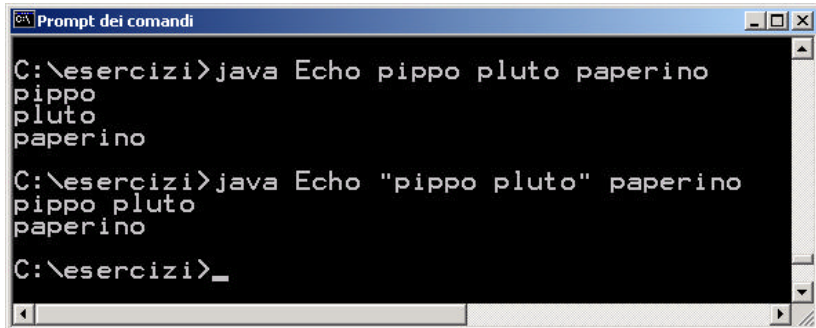
```
class RadiceQuadrata{  
    public static void main(String args[]){  
        double x,y;  
        if(args.length<1) System.out.println("sqrt "+x);  
        else{  
            y= Double.parseDouble(args[0]);  
            x = Math.sqrt(y);  
            System.out.println("sqrt "+x);  
        }  
    }  
}
```

Argomenti

Lucidi aggiunti

Argomenti

```
public class Echo {  
    public static void main (String[] args) {  
        for (int i = 0; i < args.length; i++)  
            System.out.println(args[i]);  
    }  
}
```



```
C:\esercizi>java Echo pippo pluto paperino  
pippo  
pluto  
paperino  
  
C:\esercizi>java Echo "pippo pluto" paperino  
pippo pluto  
paperino  
  
C:\esercizi>_
```

Metodi

Lucidi aggiunti

Definizione di un metodo

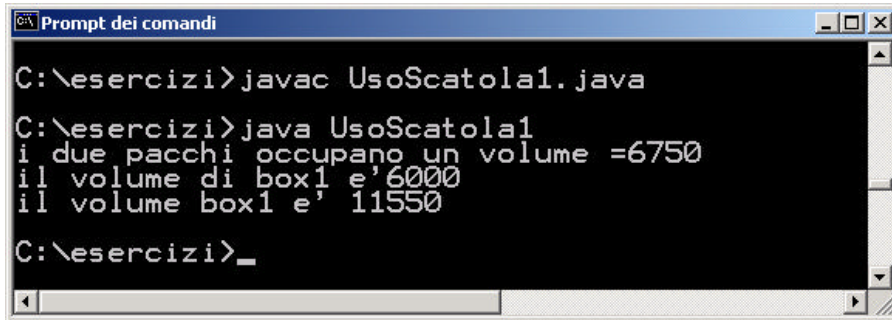
- **Metodo con valore di ritorno e con parametri**

```
<tipo> <nome metodo> (<elenco parametri>) {  
    // Codice di definizione del metodo  
    return <valore di ritorno>;  
}
```

```
int cambiaVolume(int x, int y, int z) {  
    return ((altezza+x)*(larghezza+y)*(profondità+z));  
}
```

UsoScatola1.java

```
class UsoScatola1{  
    public static void main(String args[]){  
        Scatola1 box1= new Scatola1(10,20,30);  
        Scatola1 box2= new Scatola1(5,10,15);  
        int ingombro;  
        ingombro=box1.volume()+box2.volume();  
        System.out.println("i due pacchi occupano un volume  
        =" +ingombro);  
        System.out.println("il volume di box e'" +box1.volume());  
        System.out.println("il volume box e' “ +  
        box1.cambiaVolume(4,5,3) );  
    }  
}
```



```
C:\esercizi>javac UsoScatola1.java
C:\esercizi>java UsoScatola1
i due pacchi occupano un volume =6750
il volume di box1 e' 6000
il volume box1 e' 11550
C:\esercizi>_
```

Definizione classi

Lucidi aggiunti

Esercizio

- Scrivere la classe **Pacco** avente i seguenti attributi **base**, **altezza e profondità**: tre numeri con la virgola rispettivamente la base, l'altezza e la profondità del pacco, e l'attributo **numero** di tipo numero intero che identifica il pacco. Il costruttore ha come parametri tre numeri con la virgola (coerente con la scelta fatta per gli attributi base, altezza e profondità) ed assegna il valore dei parametri agli attributi base, altezza e profondità. Scrivere il metodo **setNumero** che ha come parametro una variabile di tipo numero intero, tale metodo assegna il valore del parametro all'attributo numero. Scrivere il metodo **volume** che restituisce il valore del volume del pacco (prodotto dei tre attributi base, altezza e profondità).

Soluzione

```
class Pacco{
    double base,altezza,profondità;
    int numero;
    Pacco(double x, double y, double z){
        base=x;
        altezza=y;
        profondità=z;
    }
    void setNumero(int x){
        numero=x;
    }
    double volume(){
        return (base*altezza*profondità);
    }
}
```

Costruttore di default

Lucidi aggiunti

Esercizio sul costruttore di default

- Scrivere una classe ScatolaNera avente 9 attributi di cui:
 - 8 attributi sono di ciascun tipo primitivo
 - un attributo è una variabile di riferimento ad un oggetto scatola
- Scrivere una classe UsoScatole che:
 - istanzia un oggetto di tipo Scatola e stampa i suoi tre attributi,
 - istanzia un oggetto di tipo ScatolaNera e stampa i suoi 9 attributi
- Aggiungere alla classe UsoScatole le istruzioni per istanziare un array di tipo Scatola di 3 elementi e stampare tutti gli attributi del secondo elemento dell'array

```

class ScatolaNera {
    int i;
    byte b;
    short s;
    long l;
    double d;
    float f;
    char c;
    boolean x;
    Scatola y;
}

```

ScatolaNera.java



Il costruttore di default

```

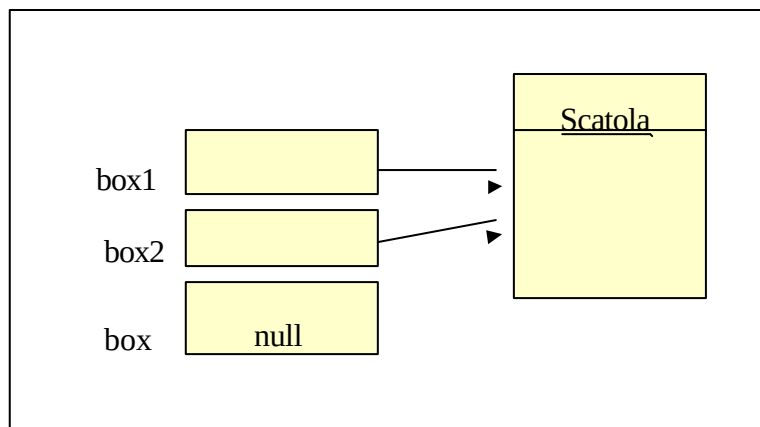
class UsoScatole {
    public static void main(String args[]) {
        Scatola box = new Scatola();
        System.out.println(box.altezza+" "+box.larghezza+" "+box.profondità);
        ScatolaNera boxN = new ScatolaNera();
        System.out.println("int="+boxN.i+" byte="+boxN.b+"
            short="+boxN.s+" long="+ boxN.l);
        System.out.println(" float="+boxN.f+" double="+boxN.d);
        System.out.println(" char="+boxN.c+"-boolean="+boxN.x);
        System.out.println(" scatola="+boxN.y);
        char c; System.out.println(" charNullo="+c+"--");
        Scatola [] a =new Scatola[3];
        System.out.println(a[1].altezza+" "+a[1].larghezza+"
            "+a[1].profondità);
    }
}

```

```
Prompt dei comandi
Microsoft Windows 2000 [Versione 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>set path=%path%;C:\j2sdk1.4.1_02\bin
C:\>cd esercizi
C:\esercizi>javac UsoScatole.java
C:\esercizi>java UsoScatole
0 0 0
int=0 byte=0 short=0 long=0
float=0.0 double=0.0
char= -boolean=false
scatola=null
C:\esercizi>
```

Null

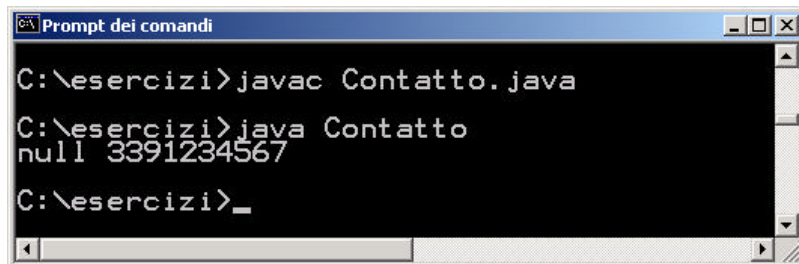


Costruttore e this

Lucidi aggiunti

Esempio

```
class Contatto {  
    String nome;  
    String numero;  
    Contatto(String nome, String numero){  
        nome= nome; //(al posto di this.nome=nome)  
        this.numero=numero;  
    }  
    public static void main(String args[]){  
        Contatto c= new Contatto("mario","3391234567");  
        System.out.println(c.nome+" "+c.numero);  
    }  
}
```



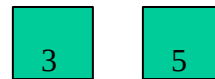
```
C:\esercizi>javac Contatto.java  
C:\esercizi>java Contatto  
null 3391234567  
C:\esercizi>_
```

Passaggio di parametri

Lucidi aggiunti

Generico Programma passaggio per riferimento

```
a=3; b=5;  
funzione(a,b);
```



a b
► 101 111

Indirizzi aree di memoria

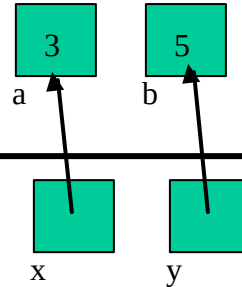
```
funzione(int x,int y){  
    x=x+3;  
    y=y+5  
}
```



x y
Gli indirizzi delle aree di memoria identificate dai parametri attuali vengono copiati nelle aree di memoria identificate dai parametri formali **x e y**

Generico Programma passaggio per riferimento

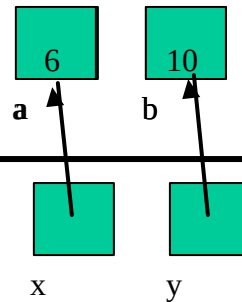
```
a=3; b=5;  
funzione(a,b);
```



```
funzione(int x,int y){  
    x=x+3;  
    y=y+5  
}
```

Generico Programma passaggio per riferimento

```
a=3; b=5;  
funzione(a,b);
```

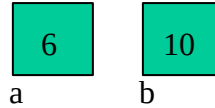


```
funzione(int x,int y){  
    x=x+3;  
    y=y+5  
}
```

L'esecuzione del
metodo modifica i
parametri attuali **a e b**

Generico Programma passaggio per riferimento

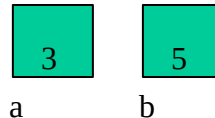
```
a=3; b=5;  
funzione(a,b);  
stampa(a,b);
```



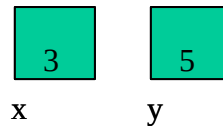
Stamperà 6 e 10

Generico Programma passaggio per valore

```
a=3; b=5;  
funzione(a,b);
```



```
funzione(int x,int y){  
    x=x+3;  
    y=y+5  
}
```



I valori dei parametri
attuali vengono copiati
nelle aree di memoria
identificate dai
parametri formali **x e y**

Generico Programma passaggio per valore

```
a=3; b=5;  
funzione(a,b);
```

3

a

5

b

```
funzione(int x,int y){  
    x=x+3;  
    y=y+5  
}
```

6

x

10

y

L'esecuzione del
metodo modifica i soli
parametri formali **x e y**

Generico Programma passaggio per valore

```
a=3; b=5;  
funzione(a,b);  
stampa(a,b);
```

3

a

5

b

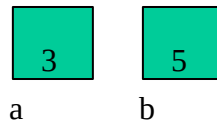
Stamperà 3 e 5

Passaggio parametri

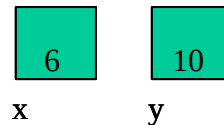
- **In Java i parametri sono sempre passati per valore**
 - **Se il parametro è un tipo primitivo è passato per valore**
 - Se il parametro è un oggetto: la variabile di riferimento è passato per valore
 - Tramite il riferimento all'oggetto si può accedere agli attributi ed ai metodi dell'oggetto e modificare lo stato dell'oggetto

Programma Java

```
...  
a=3; b=5;  
...  
obj.funzione(a,b);
```



```
...  
void funzione(int x,int y){  
    x=x+3;  
    y=y+5  
}  
...
```

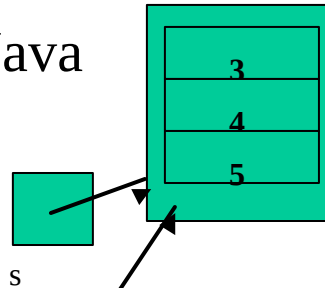


Passaggio parametri

- **In Java i parametri sono sempre passati per valore**
 - Se il parametro è un tipo primitivo è passato per valore
 - **Se il parametro è un oggetto: la variabile di riferimento è passato per valore**
 - Tramite il riferimento all'oggetto si può accedere agli attributi ed ai metodi dell'oggetto e modificare lo stato dell'oggetto

Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);
```

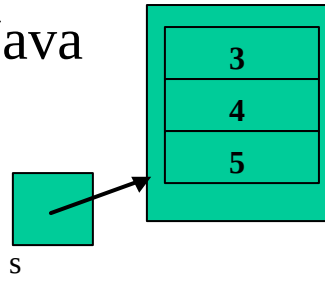


```
...  
void funzione(Scatola x){  
    x=new Scatola(2,2,2);  
}
```

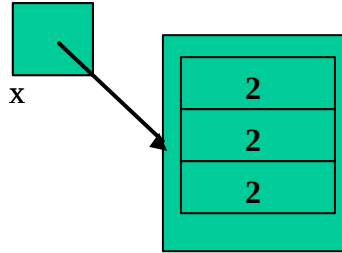


Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);
```

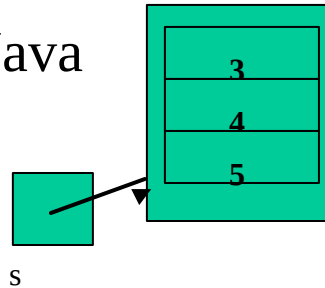


```
...  
void funzione(Scatola x){  
    x=new Scatola(2,2,2);  
}  
...
```



Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);  
...
```



```
System.out.println(s.altezza+" "+s.larghezza+" "+  
+s.profondità);
```

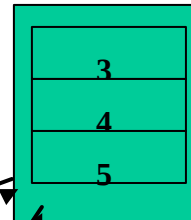
Stamperà: 3 4 5

Passaggio parametri

- **In Java i parametri sono sempre passati per valore**
 - Se il parametro è un tipo primitivo è passato per valore
 - Se il parametro è un oggetto: la variabile di riferimento è passato per valore
 - **Tramite il riferimento all'oggetto si può accedere agli attributi ed ai metodi dell'oggetto e modificare lo stato dell'oggetto**

Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);
```

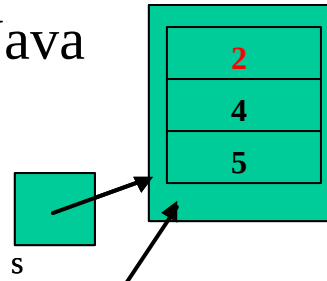


```
...  
void funzione(Scatola x){  
    x.altezza =2;  
}  
...
```

x

Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);
```

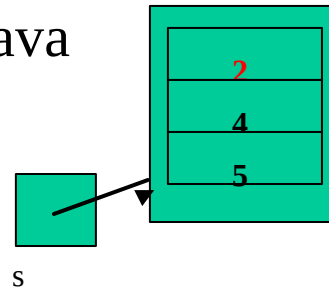


```
...  
void funzione(Scatola x){  
    x.altezza =2;  
}  
...
```



Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);  
...
```

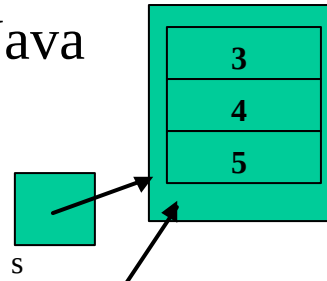


```
System.out.println(s.altezza+" "+s.larghezza+" "+  
+s.profondità);
```

Stamperà: 2 4 5

Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);
```

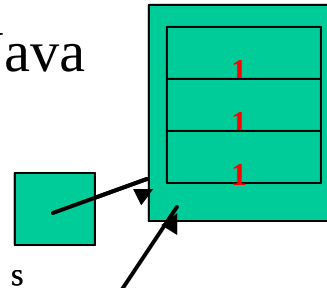


```
...  
void funzione(Scatola x){  
    x.setDimensioni(1,1,1);  
}  
...
```



Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);
```

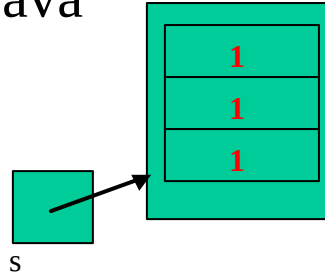


```
...  
void funzione(Scatola x){  
    x.setDimensioni(1,1,1);  
}  
...
```



Programma Java

```
...  
a=3; b=5;  
Scatola s =new Scatola(3,4,5);  
obj.funzione(s);  
...
```



```
System.out.println(s.altezza+" "+s.larghezza+" "+  
+s.profondità);
```

Stamperà: **1 1 1**