

# Corso di Laurea Ingegneria Informatica

## Fondamenti di Informatica 1

---

Dispensa E06

## Passaggio di parametri

---

**C. Limongelli**

Novembre 2007

# Contenuti

---

- ❑ **Passaggio di tipi primitivi**
- ❑ **Passaggio di tipi riferimento: Stringhe**
- ❑ **Passaggio di tipi riferimento: Oggetti**
- ❑ **Esempio: la classe Razionale**

# Passaggio di parametri di tipo primitivo...

- Indicare che cosa viene visualizzato sullo schermo dall'esecuzione della seguente applicazione Java.

```
class LegameParametri {
    public static void alfa(int a, int b) {
        System.out.println("2: a vale " + a + ", b vale " + b);
        a = a + 3;
        b = b - 3;
        System.out.println("3: a vale " + a + ", b vale " + b);
    }
    public static void main(String[] args) {
        int a, b;
        a = 5;
        b = 8;
        System.out.println("1: a vale " + a + ", b vale " + b);
        alfa(b-2,a+2);
        System.out.println("4: a vale " + a + ", b vale " + b);
    }
}
```

# ... Passaggio di parametri di tipo primitivo

- Indicare che cosa viene visualizzato sullo schermo dall'esecuzione della seguente applicazione Java.

```
class LegameParametri {
    public static void alfa(int b, int c) {
        int a;
        a = 2;
        System.out.println("2: a = " + a + ", b = "+b+ ", c = " + c);
        a = a + 3;
        b = b - 3;
        c = a + b;
        System.out.println("3: a = " + a + ", b = "+b+ ", c = " + c);
    }
    public static void main(String[] args) {
        int a, b, c;
        a = 4;
        b = 7;
        c = 5;
        System.out.println("1: a = " + a + ", b = "+b+ ", c = " + c);
        alfa(a+2,b-2);
        System.out.println("4: a = " + a + ", b = "+b+ ", c = " + c);
    }
}
```

# Passaggio di parametri di tipo primitivo

- Indicare che cosa viene visualizzato sullo schermo dall'esecuzione della seguente applicazione Java.

```
class LegameParametri {
    public static void alfa(int a, char b) {
        System.out.println("2: a vale " + a + ", b vale " + b);
        a = a + 3;
        b = (char)(b - 3);
        System.out.println("3: a vale " + a + ", b vale " + b);
    }
    public static void main(String[] args) {
        char a; int b;
        a = '5';
        b = 8;
        System.out.println("1: a vale " + a + ", b vale " + b);
        alfa(b-2,a);
        System.out.println("4: a vale " + a + ", b vale " + b);
    }
}
```

# Passaggio di parametri di tipo String: oggetti non modificabili

---

- ❑ Tutti gli oggetti della classe **String** sono **immutabili** e quindi il loro valore non può essere modificato dopo che sono stati creati.
- ❑ La classe **String** non mette a disposizione alcun metodo per modificare lo stato dell'oggetto.
- ❑ Supponendo che alla variabile **s** di tipo **String** sia assegnata una stringa, assegnare una nuova stringa alla variabile **s** corrisponde a creare un nuovo oggetto e quindi un nuovo riferimento che viene assegnato a **s**, senza modificare la vecchia stringa.
- ❑ La stringa puntata dal vecchio riferimento rimane in memoria, ma non è più accessibile.

# Passaggio di parametri di tipo String

- Quali valori vengono stampati dalla seguente applicazione?

```
class LegameStringhe {
    public static void alfa(String k) {
        System.out.println(" dentro alfa = " + k);
        /* crea un nuovo oggetto il riferimento al
           precedente viene perso */
        k = k.concat("giorno");
        System.out.println(" dentro alfa = " + k);
    }
    public static void main(String[] args) {
        String k = "buon";
        System.out.println("nel main prima di alfa = " + k);
        alfa(k);
        System.out.println("nel main dopo alfa = " + k);
    }
}
```

# Passaggio di parametri di tipo riferimento

## □ Da ricordare:

- La modifica dell'oggetto il cui riferimento è passato come parametro avviene solo quando si modifica lo stato dell'oggetto, senza che il riferimento sia modificato.

## □ Esempio: la classe numero

<b>Numero</b>
valore : int
<b>&lt;costruttore&gt;</b> Numero(int v) <b>&lt;operazioni&gt;</b> int getValore() void setValore(int v)



# Esempio

```
class DemoNumeroPassPar{
    public static void alfa(Numero n){
        // viene creato un nuovo oggetto referenziato da n
        n = new Numero(50);
        System.out.println("n1 in alfa = " + n.getValore());}

    public static void beta(Numero n){
        // viene modificato lo stato dell'oggetto referenziato da n
        n.setValore(50);
        System.out.println("n1 in beta = " + n.getValore());}

    public static void main(String[] args) {
        Numero n1;
        n1 = new Numero(10); //crea il numero
        System.out.println("n1 nel main = " + n1.getValore());
        alfa(n1);
        System.out.println("n1 nel main dopo alfa = "
            + n1.getValore());

        beta(n1);
        System.out.println("n1 nel main dopo beta= " +
            n1.getValore());
    }
}
```

# Esempio: la classe razionale

- ❑ Si vogliono definire le operazioni sui numeri razionali
- ❑ Un razionale è un oggetto il cui stato è rappresentato da due valori **nu** e **de**
- ❑ Le operazioni sono:
  - Semplificazione di un razionale
  - Verifica se due razionali sono uguali
  - Addizione di due razionali
  - Moltiplicazione di due razionali

# La classe razionale

## □ Stato:

```
/* attributi */  
private long nu; //numeratore  
private long de; //denominatore
```

## □ Costruttore:

```
/* costruttore del razionale */  
public Razionale(long a, long b) {  
    nu = a; de = b;  
}
```

# Semplificazione e metodo equals

```
/* semplifica un numero razionale es: 4/18 -> 2/9 */
```

```
public void simpl() {  
    long app;  
    app = mcd(nu,de);  
    nu = nu/app;  
    de = de/app;  
}
```

```
/* verifica se due razionali sono uguali */  
public boolean equals(Razionale z) {  
    z.simpl();  
    this.simpl();  
    return ( this.nu == z.nu && this.de == z.de );  
}
```

# Addizione

```
/* addizione tra due numeri razionali */  
public Razionale add(Razionale z) {  
    long num, den;  
    Razionale r;  
    num = nu * z.de + z.nu * de;  
    den = de * z.de;  
    r = new Razionale(num,den);  
    r.simpl();  
    return r;  
}
```

## □ Osservazione:

- si potrebbe migliorare? Se devo sommare  $23/144 + 15/72$  come procede l'algoritmo?
- Come procediamo noi?

# Dall'oggetto alla stringa

```
public String toString() {  
    String str = "(" + nu + "/" + de + ")";  
    return str;  
}
```

# Altre operazioni da implementare

```
public Razionale divide(Razionale z) {  
    ...  
}
```

```
public Razionale subtract(Razionale z) {  
    ...  
}
```

```
public Razionale multiply(Razionale z) {  
    ...  
}
```

# Metodi di supporto

```
/* calcola il massimo comun divisore di due interi; non deve  
essere vista al di fuori della classe perche' e'  
funzionale alle operazioni tra i razionali*/
```

```
private static long mcd(long x, long y){  
    long t,r;  
  
    if ( x < y ) { /* se x < y li scambio */  
        t = x;  
        x = y;  
        y = t; }  
    while ( y > 0 ) {  
        r = x % y;  
        x = y;  
        y = r;  
    }  
    return x;  
}  
}
```



# Uso della classe razionale

```
public class TestRazionale {  
    public static void main(String[] args) {  
        Razionale x,z,r,a;  
  
        x = new Razionale(1,3);  
        x.simpl();  
        System.out.println("x = " + x.toString());  
  
        z = new Razionale(8,12);  
        z.simpl();  
        System.out.println("z = " + z.toString());  
  
        r = x.add(z);  
        System.out.println("r = " + r.toString());  
        a = new Razionale(2,3);  
        System.out.println("a = " + a.toString());  
        System.out.println("verifico che a e r siano uguali");  
        System.out.println(r.equals(a));  
    }  
}
```