

## Operatori di traslazione

```
public class IntOps {
    public static void main(String[] args) {
        int a = 1; // decl. + init.
        int b = -2; // decl. + init.
        int c, d, e; // decl.

        c = a << 2; // c = 4

        // signed shift
        d = b >> 1; // d = -1
        System.out.println("d = " + d);

        // unsigned shift
        e = b >>> 1; // e=2147483647=(2^n - 2)/2

        System.out.println("e = " + e);
    }
}
```

## Classe Math

```
public class FloatMath {
    public static void main(String[] args) {

        double d = +2.67;
        long l;

        // troncamento
        l = (long)d;
        System.out.println(l); // 2

        // arrotondamento
        l = Math.round(d);
        System.out.println(l); // 3

        d = 2.0;
        double s = Math.sqrt(d);
        System.out.println(s); // 1.4142...

    }
}
```

## Classe Double

```
public class CostantiFloat {
    public static void main(String[] args) {

        double f = +1.0, q = 0.0;
        double r;

        System.out.println("Max value: " +
            Double.MAX_VALUE);
        System.out.println("Min value: " +
            Double.MIN_VALUE);
        System.out.println("POSITIVE INFINITY: " +
            Double.POSITIVE_INFINITY);
        System.out.println("NEGATIVE INFINITY: " +
            Double.NEGATIVE_INFINITY);
        System.out.println("NOT a NUMBER: " +
            Double.NaN);

        r = f/q;
        if ( Double.isInfinite(r) )
            System.out.println("r e' infinito");
        if ( r == Double.POSITIVE_INFINITY )
            System.out.println("r e' uguale a
                POSITIVE_INFINITY");
        r = q/q;
        if ( Double.isNaN(r) )
            System.out.println("r e' NaN");
        if ( r != Double.NaN )
            System.out.println("I NaN sono tutti
                diversi tra di loro");
    }
}
```

## Operatori booleani

```
public class BoolOps {
    public static void main(String[] args) {
        boolean a = true;
        boolean b = false;
        boolean u, d, n, x;

        u = a==b;    // u = false
        System.out.println("u = " + u);
        d = a != b;  // d = true
        System.out.println("d = " + d);
        n = !b;      // n = true
        System.out.println("n = " + n);

        int i = 1;
        int j = 2;
        boolean v;

        v = (i==2) & (j==2);
        // (j==2) e' valutata
        System.out.println("v = " + v);
        v = (i==2) && (j==2);
        // (j==2) non e' valutata
        System.out.println("v = " + v);
    }
}
```

## Variabili finali

```
public class Costanti {
    public static void main(String[] args) {
        final double VOLUME_BOTTIGLIA = 2.0;
        final double LITRI_PER_ONCIA =
            0.029573529;
        final double VOLUME_LATTINA = 12 *
            LITRI_PER_ONCIA;

        int bottiglie = 4;
        int lattine = 10;

        double volumeTotale =
            bottiglie * VOLUME_BOTTIGLIA +
            lattine * VOLUME_LATTINA;
        System.out.println("Volume totale = " +
            volumeTotale);
    }
}
```

## Assegnamento

```
public class Assegnamento {
    public static void main(String[] args) {
        int c;
        long l;

        int a = 1, b = 2;
        c = a+b; // identita'
        l = a+b; // estensione int->long

        long l1 = 1, l2 = 2;
        l = l1+l2; // identita'
        // c = l1+l2; compile-time error
    }
}

/*****
*****

public class Assegnamento2 {
    public static void main(String[] args) {
        int a, b;
        a = b; // compile-time error
    }
}
```

## Letterali e Conversioni

```
public class LetteraliEConversioni {
    public static void main(String[] args) {

        char c2 = 'Z';
        char c1 = 0xffff; // max char (hex)
        byte b = 0x7f;    // max byte (hex)
        // byte b = 0x1ff;

        short s = 0x7fff; // max short (hex)
        int i1 = 0x2f;     // hex lowercase
        int i2 = 0X2F;    // hex uppercase
        int i3 = 0177;    // octal

        long n1 = 200L;   // long: suffisso 'L'
        long n2 = 200l;   // long: suffisso 'l'
        long n3 = 200;

        float f1 = 1;
        // float f1 = 1.0;
        float f2 = 1F;
        float f3 = 1f;
        float f4 = 1e-45f;
        float f5 = 1e+9f;
        double d1 = 1d;
        double d2 = 1D;
        double d3 = 47e47d;
    }
}
```

## Cast

```
public class Cast {
    public static void main(String[] args) {

        double b = 3.0;
        double d = 2.6;

        double r;
        r = b + d;
        System.out.println(r); // 5.6

        int i;
        // i = b + d;
        // double non e' assegnabile a int
        i = (int)(b + d);
        // double -> int per troncamento
        System.out.println(i); // 5
    }
}
```

## Promozioni Unarie

```
public class Promozioni {
    public static void main(String[] args) {
        int r;
        long l = 5;

        int i = 1;
        r = i << 1; // r = 32
        System.out.println(r);

        short s = 1;
        r = s << 1; // r = 32
        System.out.println(r);

        // short t;
        // t = (s << 1); compile-time error
        //
    }
}
```

## Promozioni Binarie

```
public class PromozioniBinarie {
    public static void main(String[] args) {
        int i = 0;
        float f = 1.0f;
        double d = 2.0;
        boolean r;

        // int*float -> float*float
        // float==double -> double==double
        r = (i * f) == d;
        System.out.println(r);

        byte b=0x1f;
        char c = 'G';
        //char&byte -> int&int
        int control = c&b;
        System.out.println(control);

        //int:float -> float:float
        f = (b == 0)? i : 4.0f;
        System.out.println(1.0/f);
    }
}
```

## Espressione di controllo

```
public class EspressioneDiControllo {
    public static void main(String[] args) {
        int a = 1;
        if ( a != 0 )
            // alla C++
            // if (a) // compile-time error
            System.out.println("a e' diverso
                               da zero");
        else
            System.out.println("a e' uguale
                               a zero");
    }
}

/*****
*****/
public class EspressioneDiControllo {
    public static void main(String[] args) {
        int a = 0;
        if ( a = 1 ) // compile-time error
            System.out.println("a e' diverso
                               da zero");
        else
            System.out.println("a e' uguale
                               a zero");
    }
}
```

## "Funzioni"

```
public class Funzioni {
    static int test(int testval, int target){
        if (testval > target)
            return +1;
        else if (testval == target)
            return 0;
        else return -1;
    }
    public static void main(String[] args) {
        System.out.println(test(10, 5));
        System.out.println(test(5, 10));
        System.out.println(test(5, 5));
    }
}
```