

Complex.java

```
public class Complex {
    // variabile statica o di classe
    // ne esiste una istanza per la classe, non una per oggetto
    // nasce con il programma (al primo refs al classe)
    // muore al termine del programma
    // può essere inizializzata (altrimenti iniz. di default)
    // qui è usata come contatore di numeri complessi
    static int num_complex=0;

    // variabili di istanza
    double Re;
    double Imm;

    // se definisco un costruttore il default non vale più
    // in tale costruttore si una il riferimento this
    // this è inizializzato all'oggetto su cui è invocato
    // il metodo di istanza o il costruttore
    // in tale esempio risolve problemi di visibilità
    // perchè il parametro Re 'oscura' la visibilità della
    // variabile di istanza Re
    // si incrementa inoltre la variabile statica num_complex
    // perchè si è costruito un nuovo oggetto
    public Complex(double Re, double Imm){
        this.Re=Re;
        this.Imm=Imm;
        num_complex++;
    }

    // se definisco un costruttore il default non vale più
    // per cui conviene definirlo
    // in tal caso l'uso di this è inutile
    // in genere si usa per rendere i programmi più leggibili
    public Complex(){
        this.Re=0;
        this.Imm=0;
        num_complex++;
    }

    // metodo somma, restituisce la somma di del num complesso su cui è
    // invocato e del parametro
    public Complex somma(Complex A){
        Complex Risu = new Complex();
        Risu.Re=Re+A.Re;
        Risu.Imm=Imm+A.Imm;
        return(Risu);
    }

    // esempio di somma con l'utilizzo di this
    public Complex somma1(Complex A){
        double Re = this.Re+A.Re;
        double Imm = this.Imm+A.Imm;
        //Complex Risu = new Complex();

        return(new Complex(Re,Imm));
    }

    // metodo notevole toString
    // invocato automaticamente quando compare un riferimento ad un oggetto
}
```

Complex.java

```
// e ci si aspetta una stringa
// esempio: System.out.println(C) doce C è un complex
// se è definito il metodo toString, viene invocato automaticamente C.toString()
// e il risultato viene visualizzato
// cio vale anche nelle espressioni
public String toString(){
    return(Re+" "+Imm+"j");
}

// metodo notevole equals
// serve per confrontare 2 numeri complessi
// solitamente si inserisce nella classe
public boolean equals(Complex A){
    boolean Risu = false;
    if(A!=null&&Re==A.Re && Imm==A.Imm)
        Risu=true;
    return(Risu);
}

// metodi setter e getter
// leggono e scrivono le variabili di stato (di istanza) della classe.
// sono particolarmente utili quando le variabili di istanza sono dichiarate
// private. In tal modo non sono accessibili al di fuori della classe
// e l'unico modo per accedervi è tramite metodi scritti dall'autore della classe.
// si reaizza l'information hiding, perchè l'utente della classe non vede lo stato,
// e l'accesso allo stato avviene solo tramite metodi scritti dall'autore della classe,
// evitando ad esempio inconsistenze o accessi non autorizzati
public double GetRe() {
    return (this.Re);
}

public double GetImm() {
    return (this.Imm);
}

public void SetRe(double A) {
    this.Re=A;
}

public void SetImm(double A) {
    this.Imm=A;
}

// esempi d'uso
public static void main(String[] args) {
    // TODO Auto-generated method stub

    // Complex A = new Complex(); // errore se sedinisco il costruttore con parametri
    // ma non il costruttore di default (senza parametri)
    Complex A = new Complex(1,2);
    System.out.println("esempi uso toString e variabile statica");
    // esempio d'uso del metodo toString (implicito)
    System.out.println("A= "+A);
    // esempio d'uso della variabile statica che conta i complessi creati
    System.out.println("numcomplex= "+num_complex);
    System.out.println();

    Complex B=new Complex(1,2);
    System.out.println("B= "+B);
    System.out.println("numcomplex= "+num_complex);
    System.out.println();
}
```

Complex.java

```
// esempio d'uso di somma ed equals
System.out.println("esempio uso somma ed equals");
Complex C = A.somma(B);
System.out.println("A+B= "+C);
System.out.println("numcomplex= "+num_complex);
System.out.println("A==B "+A.equals(B));
System.out.println("numcomplex= "+num_complex);
System.out.println();

// esempio d'uso di somma1(che utilizza this) ed equals
// esempio d'uso del modo generale di accedere alla variabile static
System.out.println("Test somma1");
C = A.somma1(B);
System.out.println("A+B= "+C);
System.out.println("numcomplex= "+num_complex);
System.out.println("A==B "+A.equals(B));
System.out.println("numcomplex= "+Complex.num_complex);
System.out.println();

// esempi d'uso di getter e setter
System.out.println("Test Getter");
System.out.println("Re di A= "+A.GetRe());
System.out.println("Imm di A= "+A.GetImm());
System.out.println("Test Setter");
A.SetRe(10);
A.SetImm(10);
System.out.println("A= "+A);
}
}
```