

Viene di seguito mostrato il codice di massima di una ipotetica classe che svolga le funzioni fondamentali di una calcolatrice elementare. Nelle ultime lezioni (martedì 17/2) i partecipanti hanno potuto implementarne una secondo i propri criteri (gli studenti dei turni del giovedì faranno altrettanto).

Confrontando il proprio codice con quello qui elencato, sarebbe opportuno apportare dei miglioramenti per ottenere i seguenti risultati:

- 1) Creare una classe nuova che implementi le funzioni di lettura dallo standard input di una stringa; tale classe dovrebbe essere dichiarata nel main e sfruttata per la lettura dell'input da parte dell'utente. Per fare ciò si dovranno implementare al minimo i due metodi attualmente dichiarati nella classe che contiene il main: `readChar()` e `readLine()`. Lo scopo di tale classe sarà quello di fare da base per classi da questa derivata e che saranno specializzate ad esempio per ritornare solo i valori per gli operatori o degli operandi.
- 2) Attualmente il main è implementato per utilizzare ogni volta una coppia di operandi su cui effettuare una sola operazione; ebbene dovrà essere modificato tale comportamento per agire sempre sull'ultimo risultato della precedente operazione.
- 3) Aggiungere un test per vedere se l'operatore ha introdotto un operando errato: es ha introdotto un stringa contenente valori alfanumerici o abbia separato numeri con spazi bianchi di difficile interpretazione
- 4) Aggiungere un test per l'inserimento di un operando corretto limitandovi al momento alle quattro operazioni;
- 5) Inoltre qualsiasi modifica che riteniate utile;

Il tutto sarà ripreso all'inizio della prossima lezione e rivisto assieme

Buon Lavoro

```
/*
 * Calc.java
 */

/**
 */

public class Calc {
    private float val, mem;
    /** Creates a new instance of Calc */
    public Calc() {
        MemClear();
        Clear();
    }
    public void Store(float v){
```

```

        mem=v;
    }
    public void MemClear(){
        mem=0.0f;
    }
    public void MemSum(float tosum){
        mem+=tosum;
    }
    public void MemSubtract(float tosub){
        mem-=tosub;
    }
    public float getMem(){
        return mem;
    }
    public void PrintMem(){
        System.out.println(mem);
    }
    //
    public void Clear(){
        val=0.0f;
    }
    public void Set(float v){
        val=v;
    }
    public void Sum(float tosum){
        val+=tosum;
    }
    public void Subtract(float tosub){
        val-=tosub;
    }
    public void Multiply(float tomul){
        val*=tomul;
    }
    public void Divide(float todiv){
        val/=todiv;
    }
    public float GetResult(){
        return val;
    }
    public void Print(){
        System.out.println(val);
    }
    public static char readChar()
    {
        //inizializza il valore di ritorno

```

```

    int charAsInt = -1;
    //legge il carattere più vecchio rimasto nello
    // standard input
    try
    {
        charAsInt = System.in.read();
    }
    catch(java.io.IOException e)
    {
        System.out.println(e.getMessage());
        System.out.println("Fatal error. Ending
Program.");
        System.exit(0);
    }
    //ritorna il carattere letto o -1 se si è
    verificato un errore
    return (char)charAsInt;
}

// Legge un'intera riga dallo standard input.
public static String readLine()
{
    //variabile che contiene il valore carattere
    //letto dallo standard input
    char nextChar;
    //variabile che contiene il valore interpretado
    //della linea letta dallo standard input
    String result = "";
    //variabile di controllo per il loop di lettura
    boolean done = false;
    //loop di lettura
    while (!done)
    {
        //prossimo carattere letto dallo s.i.
        nextChar = readChar();
        //letto il carattere di fine linea
        if (nextChar == '\n')
            done = true;
        else
            //appende un carattere letto da s.i. alla
            //variabile che
            //verrà restituita al metodo chiamante
            result = result + nextChar;
    }
    //ritorna il valore con la stringa letta da s.i.

```

```

        return result;
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        //dichiarazione ed allocazione di
        //memoria per la classe Calc
        Calc c=new Calc();
        //variabile che governa lo stato del loop di
        //input da s.i.
        boolean ancora=true;
        do{
            //input primo operando
            System.out.print("Inserisci il primo operando
:");

            String uno=readLine();
            //test per la continuazione dell'input
            if(uno.compareToIgnoreCase("fine")!=0){
                //??aggiungere test per operando errato
                c.Set(Float.parseFloat(uno));
                //input operatore
                System.out.print("Inserisci l'operatore
:");

                String oper=readLine();
                //??aggiungere test per input operatore
                //errato????????
                //input secondo operando
                System.out.print("Inserisci il secondo
operando :");

                String due=readLine();
                //????????????aggiungere test per operando
                // errato????????????
                if(oper.equals("/") || oper.equals("/")){
                    c.Divide(Float.parseFloat(due));
                }
                else if(oper.equals("*")){
                    c.Multiply(Float.parseFloat(due));
                }
                else if(oper.equals("+")){
                    c.Sum(Float.parseFloat(due));
                }
                else if(oper.equals("-")){
                    c.Subtract(Float.parseFloat(due));
                }
            }
        } while (ancora);
    }
}

```

```
        //stampa risultato
        System.out.print("Risultato : ");
        c.Print();
    }
    else
        //imposta la variabile ancora per fare
        //terminare il loop di lettura da s.i.
        ancora=false;
    }while(ancora);
}
}
```