

javaTryCatchThrow.java

```
/*
Programma : javaTryCatchThrow

Autore : Riontino Raffaele 4 informatici (corso serale)
        ITIS Molinari - Milano          17/02/2011

Funzionalità : Prove sull'utilizzo del blocco "try catch".
               Questo blocco serve per delimitare una porzione di codice
               detta "pericolosa", cioè che in caso di errore (come ad esempio
               l'inserimento di una stringa in una variabile di interi , il mal
               funzionamento della stampante , ecc) può portare alla chiusura
               improvvisa del programma e alla perdita dei dati su cui si stava lavorando.
               Nella parte try si delimita la porzione di codice da controllare,
               mentre nella parte catch viene catturata l'eventuale eccezione sollevata
               e la visualizzazione dell'errore riscontrato.
               In questo programma viene gestita l'eccezione della divisione per 0.

*/

import java.io.*;

public class javaTryCatchThrow
{
    public static void main (String args[])
    {
        //creo l'oggetto per leggere gli input
        BufferedReader input = new BufferedReader (new InputStreamReader (System.in));
        float num1 = 0, num2 = 0 , ris = 0;
        boolean eccezione = false;
        /*se in una variabile float si prova a mettere un carattere diverso da un numero,
        viene sollevata un eccezione e se non si usa il blocco "try catch" l'esecuzione
        del programma si interrompe */

        do{
            try {
                System.out.print("inserisci il primo numero (prova ad inserire un carattere) : ");
                num1 = Float.parseFloat(input.readLine());
                eccezione = false;
            }
            catch (Exception e) //Exception è un gestore delle eccezioni generiche (super classe)
            {
                System.out.println("eccezione sollevata : " + e);
                eccezione = true;
            }
        }while(eccezione);

        do{
            try {
                System.out.print("inserisci il secondo numero (0 per l'eccezione della divisione): ");
                num2 = Float.parseFloat(input.readLine());
                eccezione = false;
            }
            catch (Exception e)
            {
                System.out.println("eccezione sollevata : " + e);
                eccezione = true;
            }
        }while (eccezione);

        /*Se la divisione per un numero = 0 non viene gestita, il
        risultato che si può ottenere (in java) è 'infinity' (utilizzato in alcuni
        calcoli matematici) o 'NaN' (not a number).
        Per poter sollevare l'eccezione che non si può effettuare la divisione
        per 0 bisogna :
        1 - creare una classe derivata dalla super classe Exception (in questo
           esempio 'DivByZero');
        2 - inserire nel costruttore il metodo 'super ("messaggio da visualizzare") ;'
        3 - inserire nel blocco try il controllo 'if (num2 == 0) throw new DivByZero();'
        4 - nel blocco try catturare l'eccezione con 'catch(Exception e) {...}'
           oppure con 'catch (DivByZero e) {...}'

        */
        try
```

javaTryCatchThrow.java

```
{
    ris = num1 / num2;
    if (num2 == 0) throw new DivByZero();
    /*
    in alternativa si può effettuare il controllo sul risultato senza creare la classe
    derivata DivByZero (per gli interi questa eccezione non è gestita):
    if (Float.isInfinite(ris) || Float.isNaN(ris)) //controllo che non ritorna infinity o NaN
        throw new ArithmeticException(); //sollevo l'eccezione da catturare con
            catch(ArithmeticException e){.....} oppure
            catch(Exception e) {.....}

    */
    System.out.println("Risultato della divisione : " + ris);
}
catch (DivByZero e) //cattura le eccezioni sollevata
//catch (Exception e) // in alternativa
{
    System.out.println("Divisione non effettuata : " + e);
}
} //fine main
}
```