


 Spiegazione try, catch e throws in Java	2
♦ try	2
♦ catch	2
♦ throws	2
 ESEMPIO SPIEGATO	2
 Esercizio da Verifica – Comportamento Eccezioni	3
Consegna:	3
MWE atteso:	3
 Obiettivo della verifica:	4



Spiegazione **try**, **catch** e **throws** in Java

♦ **try**

Il blocco **try** viene usato per **scrivere il codice che potrebbe causare un errore (eccezione)** durante l'esecuzione. Se avviene un errore, l'esecuzione salta immediatamente al blocco **catch**.

♦ **catch**

Il blocco **catch** serve a **gestire l'eccezione** (l'errore) che può verificarsi nel blocco **try**. Si può stampare un messaggio, scrivere su un file di log, fermare il programma, o fare altro.

♦ **throws**

La parola chiave **throws** si usa nella **firma di un metodo** per dire che **quel metodo può generare un'eccezione** (di tipo controllato o unchecked). Chi chiama il metodo deve quindi **gestire l'eccezione** oppure propagare a sua volta il **throws**.



ESEMPIO SPIEGATO

```
public class Eccezioni {  
    // Array con 10 elementi (da indice 0 a 9)  
    public static Integer[] array = new Integer[10];  
  
    // Metodo con gestione dell'errore (try-catch)  
    public static void accessoArrayTryCatch() {  
        try {  
            // Tentativo di accesso a un indice non valido (11)  
            System.out.println("Valore: " + array[11]);  
        } catch (IndexOutOfBoundsException e) {  
            // Messaggio d'errore stampato se c'è l'eccezione  
            System.out.println("Errore: Indice non valido! " + e);  
        }  
    }  
  
    // Metodo che DELEGA la gestione dell'errore a chi lo chiama  
    public static void accessoArrayThrows() throws ArrayIndexOutOfBoundsException {  
        System.out.println("Valore: " + array[11]);  
    }  
}
```

```

public static void main(String[] args) {
    // Chiamata al metodo con try-catch
    accessoArrayTryCatch();

    // Chiamata al metodo che "lancia" l'eccezione
    try {
        accessoArrayThrows();
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Gestione esterna: " + e.getMessage());
    }
}
}

```



Esercizio da Verifica – Comportamento Eccezioni

Consegna:

Scrivi un programma Java che contiene:

1. Un array di 5 numeri interi.
2. Un metodo `leggiIndice(int i)` che:
 - Usa `try` e `catch` per gestire l'accesso all'indice `i` dell'array.
 - Se l'indice è valido, stampa il valore.
 - Se l'indice è fuori dai limiti, stampa un messaggio di errore.

MWE atteso:

```

public class ControllaIndice {
    public static int[] numeri = {10, 20, 30, 40, 50};

    public static void leggiIndice(int i) {
        try {
            System.out.println("Valore in posizione " + i + ": " + numeri[i]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Errore: indice " + i + " non valido!");
        }
    }
}

public static void main(String[] args) {
    leggiIndice(2); // OK
}

```

```
    leggiIndice(7); // Errore  
  }  
}
```

Obiettivo della verifica:

- Capire quando e perché usare `try` / `catch`.
- Riconoscere un errore di accesso a un array.
- Gestire correttamente l'eccezione.

Ti serve anche una versione con input da tastiera per rendere l'esercizio più interattivo?