TRY - CATCH

1. Java Exception:

Khi thực thi mã lệnh Java sẽ xuất hiện những lỗi không lường trước được, lỗi này có thể đến từ người dùng hoặc lập trình viên như : Dữ liệu không hợp lệ, chia cho 0, truy cập vào chỉ số không hợp lệ trong mảng, đối tượng chưa được cấp phát bộ nhớ...

Khi những lỗi này xảy ra trong quá trình thực thi, chương trình của bạn sẽ ném ra một ngoại lệ thông báo về lỗi gặp phải khiến chương trình bị kết thúc ngay lập tức.

1. Java Exception:

Để bắt và xử lý ngoại lệ, tránh việc chương trình bị kết thúc khi xảy ra lỗi ta sử dụng **try và catch. Try** sẽ giúp ta định nghĩa các câu lệnh có khả năng sinh ra lỗi và **catch** giúp xử lý khi xảy ra lỗi.

2. Try - catch:

Cú pháp:

```
try{
    //Code
}
catch(Exception e){
    //Code
}
```

Ví dụ bắt ngoại lệ chia cho 0:

```
public static void main(String[] args) {
        try {
            int a = 10, b = 0;
            System.out.println(a / b);
        } catch (Exception e) {
            System.out.println(e);
        }
        System.out.println("Continue !");
}
```

OUTPUT

```
java.lang.ArithmeticException: / by zero
Continue !
```

Mặc dù xảy ra lỗi nhưng chương trình vẫn sẽ tiếp tục thực thi các câu lệnh bên dưới.

2. Try - catch:

Bắt ngoại lệ nhập dữ liệu không hợp lệ:

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            int n = sc.nextInt();
        } catch (InputMismatchException e) {
            System.out.println("Invalid input !");
        }
    }
```

2. Try - catch:

Bắt ngoại lệ truy cập chỉ số không hợp lệ:

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            int[] a = {1, 2, 3};
            System.out.println(a[4]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Invalid index !");
        }
    }
```

2. Try - catch:

Bắt ngoại lệ chưa khởi tạo đối tượng:

```
public static void main(String[] args) {
        try {
            ArrayList<Integer>[] arr = new ArrayList[10];
            arr[0].add(100);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
```

OUTPUT

java.lang.NullPointerException

2. Try - catch:

Bạn có thể bắt cùng lúc nhiều ngoại lệ:

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            int[] a = {1, 2, 3};
            System.out.println(a[1]);
            int x = 10, y = 0;
            System.out.println(x / y);
        } catch (ArrayIndexOutOfBoundsException | ArithmeticException e) {
            System.out.println(e.getMessage());
        }
}
```

OUTPUT

```
2
/ by zero
```

3. Finally:

Câu lệnh bên trong finally luôn được thực thi cho dù có xảy ra ngoại lệ hay không.

Ví dụ 1:

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            int[] a = {1, 2, 3};
            System.out.println(a[4]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Invalid index !");
        } finally{
            System.out.println("Finally block is always executed !");
        }
    }
}
```

OUTPUT

```
Invalid index !
Finally block is always executed !
```

3. Finally:

Ví dụ 2:

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            int[] a = {1, 2, 3};
            System.out.println(a[1]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Invalid index !");
        } finally {
            System.out.println("Finally block is always executed !");
        }
}
```

OUTPUT

```
2
Finally block is always executed !
```