



JAVA CƠ BẢN (TT)

Giảng viên: Đỗ Ngọc Như Loan

NGOẠI LỆ (EXCEPTION)

- Ngoại lệ là một sự kiện xảy ra trong quá trình thực thi chương trình, phá vỡ luồng bình thường của chương trình

Ví dụ:

`int i = 4/0;`

- Khi xảy ra một ngoại lệ, nếu không xử lý thì chương trình kết thúc ngay và xuất hiện thông báo lỗi.
- Một vài ngoại lệ xảy ra bởi lỗi của người dùng, một số khác bởi lỗi của lập trình viên và số khác nữa đến từ lỗi của nguồn dữ liệu vật lý.
 - Người dùng nhập dữ liệu không hợp lệ.
 - Truy cập ngoài chỉ số mảng.
 - Một file cần được mở nhưng không thể tìm thấy.
 - Kết nối mạng bị ngắt trong quá trình thực hiện giao tiếp hoặc JVM hết bộ nhớ.
 -
- Exception là lớp cha của tất cả các lớp Exception
- Có thể kế thừa các ngoại lệ đã được định nghĩa trước để tạo ra các ngoại lệ mới.

MỘT SỐ NGOẠI LỆ

- **ArithmeticException:** Nếu chia bất kỳ số nào cho số 0
- **NullPointerException:** Nếu tham chiếu tới đối tượng null.
- **NumberFormatException:** Khi chuyển 1 chuỗi thành số mà chuỗi có định dạng không đúng
- **ArrayIndexOutOfBoundsException:** Nếu bạn thao tác trên mảng với chỉ số phần tử không hợp lệ
- **ClassCastException:** Nếu không thể chuyển kiểu object này sang kiểu object khác



XỬ LÝ NGOẠI LỆ

- Các từ khóa:

- try
- catch
- finally
- throw
- throws



CÚ PHÁP KHỎI LỆNH TRY-CATCH-FINALLY

```
try {  
    // code có thể ném ra ngoại lệ  
} catch(Exception_class_1 ex) {  
    // code xử lý ngoại lệ 1  
} catch(Exception_class_2 ex) {  
    // code xử lý ngoại lệ 2  
} catch(Exception_class_n ex) {  
    // code xử lý ngoại lệ n  
} finally {  
    // code trong khối này luôn được thực thi  
}
```




VÍ DỤ 1

```
public class ExceptionExample1 {  
    public static void main(String[] args) {  
        try {  
            int zero = 0;  
            int average = 10 / zero;  
            System.out.println("Average = " + average);  
        } catch (ArithmeticException ex) {  
            System.out.println(ex);  
        }  
        System.out.println("Finished!");  
    }  
}
```



VÍ DỤ 2

```
public class ExceptionExample2 {  
    public static void main(String[] args) {  
        try {  
            int zero = 0;  
            int average = 10 / zero;  
            System.out.println("Average = " + average);  
        } catch (NullPointerException ex) {  
            System.out.println(ex);  
        } catch (Exception ex) {  
            System.out.println(ex);  
        } finally {  
            System.out.println("Khởi lệnh finally luôn được thực thi");  
        }  
        System.out.println("Finished!");  
    }  
}
```



TỪ KHÓA THROW VÀ THROWS

- **throw:** ném ra ngoại lệ
- **throws:** cảnh báo phương thức có khả năng ném ra ngoại lệ

```
public void methodName() throws ExceptionType {  
    //body method  
    throw new ExceptionType("This is an exception");  
}
```



VÍ DỤ 3

```
import java.io.IOException;
class M {
    void method() {
        try {
            throw new IOException("Loi thiet bi");
        } catch (Exception e) {
            System.out.println("Ngoai le duoc xu ly");
        }
    }
}

public class ExceptionExample3 {
    public static void main(String args[]) {
        M m = new M();
        m.method();
    }
}
```



VÍ DỤ 4

```
import java.io.IOException;
class M {
    void method() throws IOException {
        throw new IOException("Loi thiet bi");
    }
}
public class ExceptionExample4 {
    public static void main(String args[]) {
        try {
            M m = new M();
            m.method();
        } catch (Exception e) {
            System.out.println("Ngoai le duoc xu ly");
        }
    }
}
```



VECTOR

- Là cấu trúc mảng động, hữu ích nếu không biết kích cỡ của mảng.

- Khai báo:

```
import java.util.Vector;
```

```
Vector<ObjectType> v = new Vector<ObjectType>();
```

- Các phương thức:

- size: trả về kích thước vector
- add(Object o): thêm phần tử o vào cuối
- add(int i, Object o): chèn thêm phần tử o vào vị trí i
- get(int i): truy xuất phần tử tại vị trí i
- remove(int i): xóa phần tử tại vị trí i



Ví Dụ

```
import java.util.Vector;
class VectorExample {
    public static void main(String[] args)
    {
        int n = 5;
        Vector<Integer> v = new Vector<Integer>();
        for (int i = 1; i <= n; i++)
            v.add(i);

        for (int i = 0; i < v.size(); i++)
            System.out.print(v.get(i) + " ");

        v.remove(3);
        v.add(1,8);
        for (int i = 0; i < v.size(); i++)
            System.out.print(v.get(i) + " ");
    }
}
```

