

## Bài thực hành số 11 – Ôn tập lại các kiến thức hướng đối tượng và ngoại lệ

### Bài 1.

#### 1. Cài đặt lớp Stack:

- private Object[] data;
- public Stack(int size)
- public boolean empty();
- public boolean full();
- public void put(Object data)
- public Object get(int index)

thỏa mãn các điều kiện sau:

1. Các phần tử null không được phép đẩy vào stack
2. Không được chèn thêm phần tử nếu Stack đã đầy
3. Là không hợp lệ nếu lấy ra các phần tử từ một Stack rỗng
4. Các phần tử null không thể được lấy ra từ mảng
5. Mảng các đối tượng không thể chứa bất kì phần tử null nào
6. Mảng không chứa đối tượng phải chứa toàn giá trị null;

2. Viết chương trình demo, tạo một ngăn xếp có kích thước cố định, test các sản phẩm để kiểm tra xem nó có thỏa mãn các điều kiện trên không? (Gợi ý: cần ném và bắt các ngoại lệ).

Chú ý: Sinh viên có thể bổ sung thêm các thuộc tính và phương thức phù hợp

### Bài 2:

1. Cài đặt lớp PhanSo để thực hiện các phép toán (cộng, trừ, nhân, chia với 1 phân số cho trước) theo mô tả như sau:

- Thành viên, phương thức private:
  - int tu
  - int mau
  - void toigian(); (tối giản phân số, phân số khi được khởi tạo hoặc cộng, trừ, nhân, chia với 1 phân số khác cần được tối giản trước khi trả về kết quả)
- Phương thức public:
  - PhanSo (int n, int d); //constructor, mặc định: tu=1, mau=1;
  - final PhanSo add (final PhanSo a); //cộng
  - final PhanSo sub (final PhanSo a); //trừ
  - final PhanSo multiple (final PhanSo a); //nhân
  - final PhanSo divide (final PhanSo a); //chia
  - String toString(); //đưa phân số về dạng string để in ra các output chuẩn.

2. Viết chương trình demo, tạo 2 phân số, thực hiện các phép tính cộng, trừ, nhân và chia hai phân số này rồi in kết quả ra màn hình.

Chú ý:

- Các phương thức constructor và divide cần bắt và trả về ngoại lệ DevidedByZeroException. Để làm được như vậy, cần tạo thêm lớp DevidedByZeroException kế thừa lớp Exception chuẩn và chỉ cần thêm hàm khởi tạo.

```
class DevidedByZeroException extends Exception {  
    public DevidedByZeroException () { super("Attempted to divide by zero");}  
    public DevidedByZeroException (String msg) { super(msg); }  
}  
  
public class DivideByZeroException extends ArithmeticException {  
    public DivideByZeroException(){super ("Attempted to divide by  
zero");}  
    public DivideByZeroException(String message ){ super(message ); }  
}
```