1.

```
public interface Stack {

	int length();	//현재 스택의 크기
	int capacity();	// 스택에 저장가능한 수 (비어있는 스택)
	String pop();	// top에 저장된 문자열 꺼내기
	boolean push(String val);//스택에 문자열 저장

}
```

2.

```
public class StringStack implements Stack { //Stack 인터페이스 상속

	private int size;		// 스택 크기
	private int top =-1;	//텅빈스택은 -1로 초기화
	private String stack[]; // 문자열저장 배열

	public StringStack(int size) { //생성자 파라미터로 int size
		this.size = size;
		stack = new String[size];
		this.top = size;

	}
	public boolean stackNull() { // 배열의 널체크 배열이 비었으면 true
					//하나라도 있으면 false
		if((stack == null)) {
			return true;
		}else {
			return false;
		}
	}

	@Override
	public int length() { // 현재 스택의 크기

		return size;
	}
```

```
    @Override
    public int capacity() { // 현재 비어있는 스택

        return size - top; //스택의 최대 사이즈에서 현재까지 차있는 스택에서
                           //top을 빼면 현재 스택이 몇칸 남았는지 알 수 있다.
    }


    @Override
    public String pop() { //배열을 출력할 때 사용하는 함수
        int temp = top;
        top++;
        return stack[temp]; // temp에 저장된 top의 인덱스부터 출력
    }

    @Override
    public boolean push(String val) { //문자열을 입력할때마다 top은 하나씩줄음
        if (top > 0) {//스택의 사이즈가 0보다 크다면 실행
            stack[top - 1] = val; // 스택의 가장 마지막 인덱스부터 저장
                                  //꺼낼 때 0번지부터 꺼내기 위해서
            top--;
            return true;
        } else {// top이 0이되면 false 리턴 -> 스택이 전부 다참

            return false;
        }
    }

}
```

3.

```
public class StackApp { // 메인

	public static void main(String[] args) {
		int size;		// 배열의 크기, 스택 크기
		String val;		//문자열을 입력하기 위한 변수
		Scanner sc = null;

		sc = new Scanner(System.in);
		System.out.println("0보다 큰수를 입력하세요.");
		System.out.print("총 스택 저장 공간의 크기 입력>>");

		do {// 입력값이 0보다 작거나 같으면 계속 입력
			size = sc.nextInt();
		} while (size <= 0);

		StringStack stack = new StringStack(size); // 객체 생성
							//파라미터로 size-> 스택의크기 결정
while (true) {//무한 루프

	System.out.print("문자열 입력 >>");
	val = sc.next();

	if(val.equals("그만")) {

		if(stack.stackNull()) {//널체크 배열이 비어있다면 리턴된 true로 조건검사
			System.out.println("스택이 비어있습니다.");
		}else {
			System.out.print("현재 스택에 저장된 모든 문자열 팝 : ");

							//배열에 저장된 문자열들을 모두 출력
			for(int i = 0;i<stack.length();i++) {//스택의 크기만큼 반복
				System.out.print(stack.pop() + " ");
			}
				break; //모두 출력했으면 while 탈출
			}
		}else {
			if(!stack.push(val)) { //배열에 저장
				System.out.println("스택이 꽉 차서 푸시 불가!");
				}
			}
```

```
                }
                sc.close();

        }

}
```