

## 2. 계산기 토큰 파서(Token Parser) 구현

### 문제 설명

- 문자열 배열로 된 수식을 입력받아 적절한 Token 객체로 변환하는 파서를 구현하세요.
- 이 파서는 앞서 구현한 Token 시스템을 활용하여 각 문자열을 알맞은 Token 객체로 변환해야 합니다.

### 구현 요구사항

#### 1. XTokenParser 클래스 구조

```
public class XTokenParser {  
    /**  
     * 문자열 배열을 Token 객체 리스트로 변환합니다.  
     *  
     * @param tokens 문자열로 된 토큰 배열  
     * @return Token 객체 리스트  
     * @throws IllegalArgumentException 잘못된 입력이나 인식할 수 없는 토큰  
     */  
  
    /* 메소드 속성 부여 public/protected/private, static, final*/  
    List<XToken> parse(String[] tokens) {  
        // 구현 필요  
    }  
}
```

#### 2. 변환 규칙

- 숫자: 정수 형태의 문자열 → `XTokenNumber`
- 비교 연산자: `<, >, ==, !=, <=, >=` → `XTokenLogicOperator`
- 후순위 산술연산자: `+, -` → `XTokenLowOrderArithmeticOperator`
- 선순위 산술연산자: `*, /, %` → `XTokenHighOrderArithmeticOperator`
- 괄호: `(, )` → `XTokenParenthesis`

#### 3. 예외 처리

- null 입력
- 인식할 수 없는 토큰
- 공백만 있는 문자열(단, 빈 문자열은 무시)

### 구현 예시

```
String[] input1 = {"5", "+", "3", "*", "2"};  
List<XToken> tokens1 = XTokenParser.parse(input1);  
// [XTokenNumber("5"), XTokenLowOrderArithmeticOperator("+"),  
//  XTokenNumber("3"), XTokenHighOrderArithmeticOperator("*"),
```

```
XTokenNumber("2"]

String[] input2 = {"10", ">", "5"};
List<XToken> tokens2 = XTokenParser.parse(input2);
// [XTokenNumber("10"), XTokenLogicOperator(">"), XTokenNumber("5")]

String[] input3 = {"10", ">", "", "5"};
List<XToken> tokens3 = XTokenParser.parse(input3);
// [XTokenNumber("10"), XTokenLogicOperator(">"), XTokenNumber("5")]
```

## 주의사항

- 파서는 단순히 토큰 변환만 수행하며, 수식의 문법적 유효성은 검증하지 않음
- 모든 예외는 명확한 메시지와 함께 발생시킬 것