3. 유리수 계산기 구현

개요

정수와 유리수를 표현하고 계산할 수 있는 계산기 시스템을 Java로 구현하세요.

수학적으로 정수는 유리수의 부분집합이므로, 이를 반영한 상속 기반 설계를 적용하고, 모든 입력에 대해 철저한 검증을 수행해야 합니다.

구현 요구사항

1. 클래스 설계

1.1 XNumber 인터페이스

```
public interface XNumber extends Comparable<XNumber> {
    XNumber add(XNumber other);
    XNumber subtract(XNumber other);
    XNumber multiply(XNumber other);
    XNumber divide(XNumber other);
}
```

1.2 XRational 클래스

- XNumber 인터페이스 구현
- 분자(numerator)와 분모(denominator) 필드 (final)
- 생성 시 자동으로 기약분수로 변환
 - 기약분수는 최대 공약수(GCD Greatest Common Divisor)를 구해 계산 가능
 - ㅇ 유클리드 호제법
- 분모는 항상 양수로 저장
- 모든 연산 메서드 구현

1.3 XInteger 클래스

- XRational 클래스 상속 (분모가 1인 특수한 유리수)
- modulo(XNumber other) 메서드 추가
- 정수 간 연산 결과가 정수인 경우 XInteger 객체 반환
- 정수간 연산이 아닌 경우, UnsupportedOperationException 발생

1.4 XRationalCalculator 클래스

- 문자열 수식을 받아 계산
- 중위 표기법 지원
- 연산자 우선순위: *, /, %>+, -> 비교연산자
- 괄호 지원

2. 지원 연산

2.1 산술 연산

- 덧셈 (+), 뺄셈 (−), 곱셈 (*), 나눗셈 (/)
- 나머지 (%) XInteger 간에만 가능

2.2 비교 연산

```
• >, <, ==, >=, <=, !=
```

3. 입출력 형식

3.1 입력

- 정수: 5, -3
- 유리수: [3,4] (= 3/4), [-5,2] (= -5/2)

3.2 출력

- 정수 (분모가 1): 5
- 유리수: [3,4]

4. 파라메터 검증 요구사항

4.1 생성자 검증

- 분모가 0인 경우 → ArithmeticException
- 잘못된 형식 → IllegalArgumentException

4.2 연산 메서드 검증

- null 파라메터 → IllegalArgumentException
- 0으로 나누기 → ArithmeticException
- 유리수에 나머지 연산 → UnsupportedOperationException

4.3 계산기 검증

- null 또는 빈 수식 → IllegalArgumentException
- 잘못된 수식 형식 → IllegalArgumentException
- 괄호 불일치 → IllegalArgumentException

5. 구현 예시

```
// 객체 생성
XInteger i1 = new XInteger(5);
XInteger i2 = new XInteger(3);
XRational r1 = new XRational(3, 4);
XRational r2 = new XRational(6, 8); // 자동으로 3/4로 변환
```

```
// 연산
                                         // 8
System.out.println(i1.add(i2));
System.out.println(i1.divide(i2));  // [5,3]
System.out.println(i1.modulo(i2));  // 2
System.out.println(r1.equals(r2));
                                          // true
// 계산기 사용
XRationalCalculator calc = new XRationalCalculator();
System.out.println(calc.evaluate(List.of("5", "+", "3"))); // 8
System.out.println(calc.evaluate(List.of("10", "/", "3"))); // [10,3]
System.out.println(calc.evaluate(List.of("10", "%", "3"))); // 3
System.out.println(calc.evaluate(List.of("[1,2]", "+", "[1,3]"))); //
[5,6]
System.out.println(calc.evaluate(List.of("(", "5", "+", "3", "*", "[",
"1", ",", "2", "]"))); // 4
System.out.println(calc.calculate("5 + 3"));
                                                                  // 8
System.out.println(calc.calculate("10 / 3"));
                                                                 // [10,3]
System.out.println(calc.calculate("10 % 3"));
                                                                  // 3
System.out.println(calc.calculate("[1,2] + [1,3]"));
                                                                 // [5,6]
                                                                  // 4
System.out.println(calc.calculate("(5 + 3) * [1,2]"));
```

6. 테스트 항목

다음 항목들을 테스트합니다:

1. 생성 및 정규화

- ㅇ 기약분수 변환
- ㅇ 음수 분모 처리
- ㅇ 정수/유리수 자동 변환

2. 연산

- ㅇ 각 연산별 정상 동작
- 정수 간 연산 → 정수 결과
- 혼합 연산 → 유리수 결과

3. 예외 처리

- o null 파라메터
- ㅇ 0으로 나누기
- ㅇ 잘못된 입력 형식
- ㅇ 유리수 나머지 연산

4. 계산기

- ㅇ 기본 연산
- o 연산자 우선순위
- ㅇ 괄호 처리
- ㅇ 잘못된 수식

7. 제출물

1. 소스 코드

- ㅇ 클론 받아서 작성한 프로젝트 전체를 압축
- o 압축하기 전에 ./target 폴더 삭제

8. 평가

- 단위 테스트를 통해 자동 검증
- 문제와 함께 제공된 코드는 코드 작성을 위해 지원
- 별도의 검증 코드 셋을 통해 확인

9. 제약사항

- Java 8 이상 사용
- 외부 라이브러리 사용 금지 (JUnit 제외)
- 모든 XNumber 객체는 불변(immutable)이어야 함