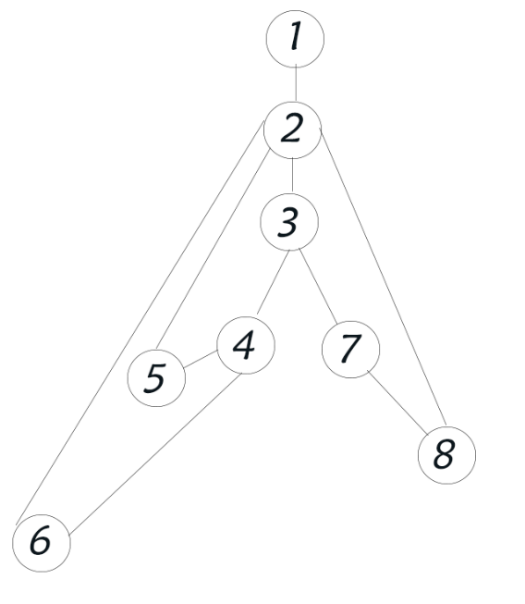
|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 测试用例 | 预期结果 | 测试结果 |
| Information.vaildAnswer | “No Meaning!” | false | 通过 |
| “-2” | false | 通过 |
| “-4/7” | false | 通过 |
| “7/4” | false | 通过 |
| “2/101” | false | 通过 |
| 200/101” | false | 通过 |
| “101” | false | 通过 |
| “50” | true | 通过 |
| “4/7” | true | 通过 |
| Fraction.getFraction | (1,0) | “No Meaning!” | 通过 |
| (0,1) | 0 | 通过 |
| (3,3) | 1 | 通过 |
| (3,-3) | -1 | 通过 |
| (2,4) | 1/2 | 通过 |
| (4,2) | 2 | 通过 |
| (2,-4) | -1/2 | 通过 |
| (4,-2) | -2 | 通过 |
| Fraction.transform | “1/2” | (1,2) | 通过 |
| “3/2” | (3,2) | 通过 |
| “1÷2” | (1,2) | 通过 |
| “2” | (2,1) | 通过 |
| Check.transform | 3 1 2 + + | 1 2 + 3 + | 通过 |
| 2 1 + 3 + | 1 2 + 3 + | 通过 |
| 3 2 1 + 4 \* + | 1 2 + 4 \* 3 + | 通过 |
| Check.isRepeated | “3+(1+2)\*4”和“(2+1)×4+3” | true | 通过 |
| “(1+2)\*(2+2)”和“(2+2)\*(1+2)” | false | 通过 |
| “(1+2)/4”和 “(1+2)÷4” | true | 通过 |
| CompeteRPN.answer-FromStack | 3 1 2 + 4 \* 6 ÷ - | “1” | 通过 |
| 2 1 + 4 \* 6 ÷ 3 - | “1” | 通过 |
| CompeteRPN.add | “1””3” | “4” | 通过 |
| “1/2””2” | “5/2” | 通过 |
| “3/4””3/2” | “9/4” | 通过 |
| RPN.transformToRPN | 3-(2+1)×4÷6 | “3 2 1 + 4 × 6 ÷ -” | 通过 |
| RPN.toStringArray | 3-(2+1)×4÷6 | “3”“-”“(”“2”“+”“1”“)”“×”“4”“÷”“6” | 通过 |

单元测试

针对可以测试并且需要测试的类与方法编写了单元测试（部分方法由随机数作为处理主体，不便测试），测试用例与结果整合如下表，并在表后附上相关程序流图：

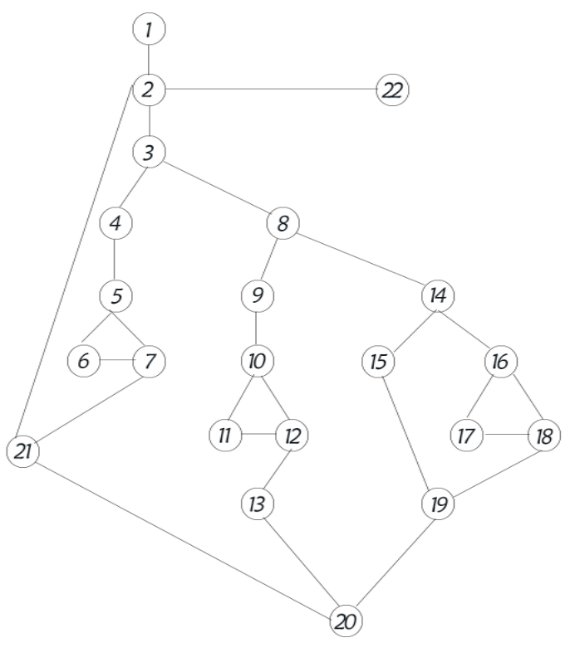
**关键函数流图：**

****

Check.isRepeated

3.if(!equation1.peek().equals(equation2.peek()))

4. if(((equation1.peek().equals("\*") || equation1.peek().equals("×")) && (equation2.peek().equals("\*") || equation2.peek().equals("×"))) || ((equation1.peek().equals("/") || equation1.peek().equals("÷")) && (equation2.peek().equals("/") || equation2.peek().equals("÷"))))

****

Check.transform

3. if(Information.isDigit(str))

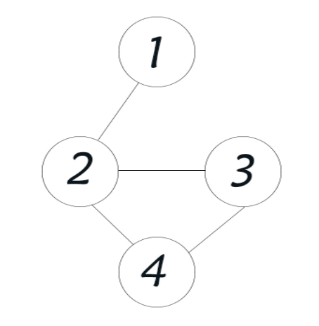
5. if(numOfTarget < 2)

8. if(!flag || numOfTarget == 2)

10.if((str.equals("+")||str.equals("\*")||str.equals("×"))&&max(first,second).equals(first))

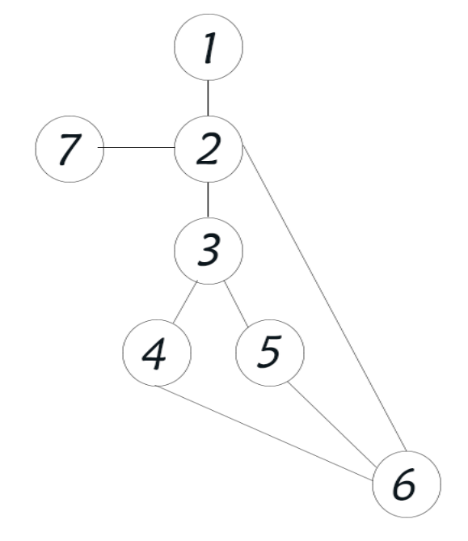
14. if(numOfTarget == 1)

16. if(numOfTarget == 0 && numOfDigit > 0)

****

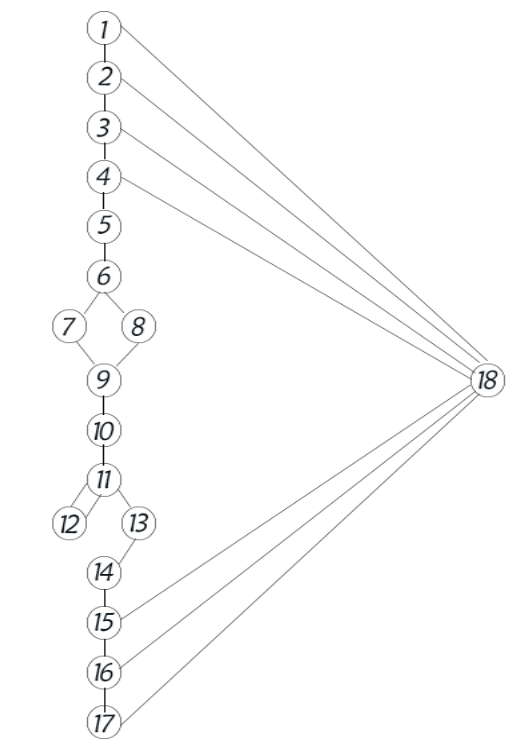
CompeteRPN.add/minus/multiply

2.if(Fraction.isFraction(a)||Fraction.isFraction(b))

****

CompeteRPN.answerFromStack

3. if(Information.isDigit(str))

****

Fraction.getFraction

1.if(denominator==0)

2.if(numerator==0)

3.if(numerator==denominator)

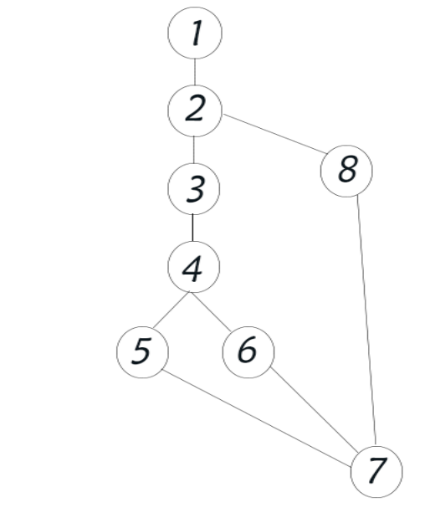
4.if(numerator+denominator==0)

6.if(Math.abs(numerator)>Math.abs(denominator))

15. if(denominator==1)

16. if(denominator==-1)

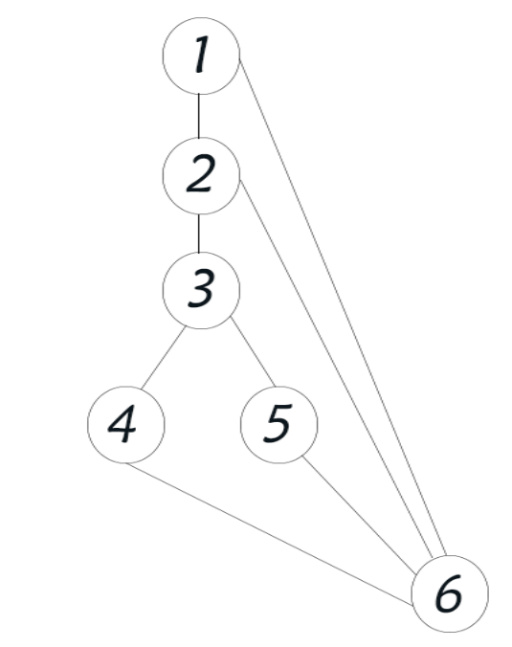
17.if(numerator\*denominator<0)

****

Fraction.transform

2. if(Fraction.isFraction(a))

4. if(endIndex == -1)

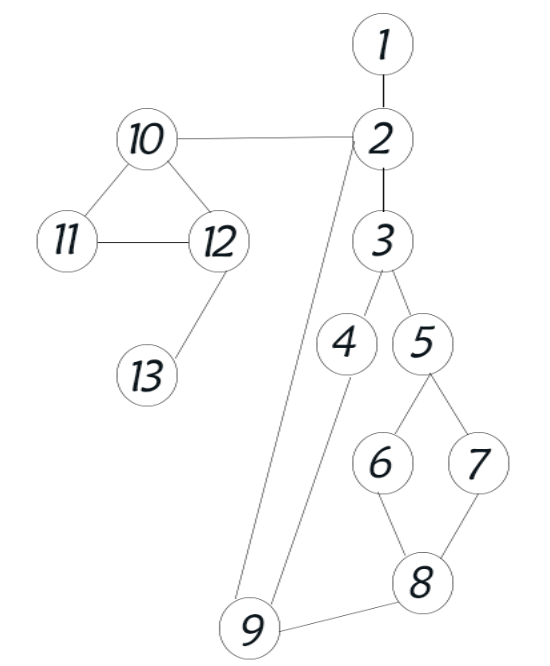
****

Information.vaildAnswer

1.if(answer.equals("No Meaning!"))

2. if(answer.charAt(0) == '-')

3. if(Fraction.isFraction(answer))

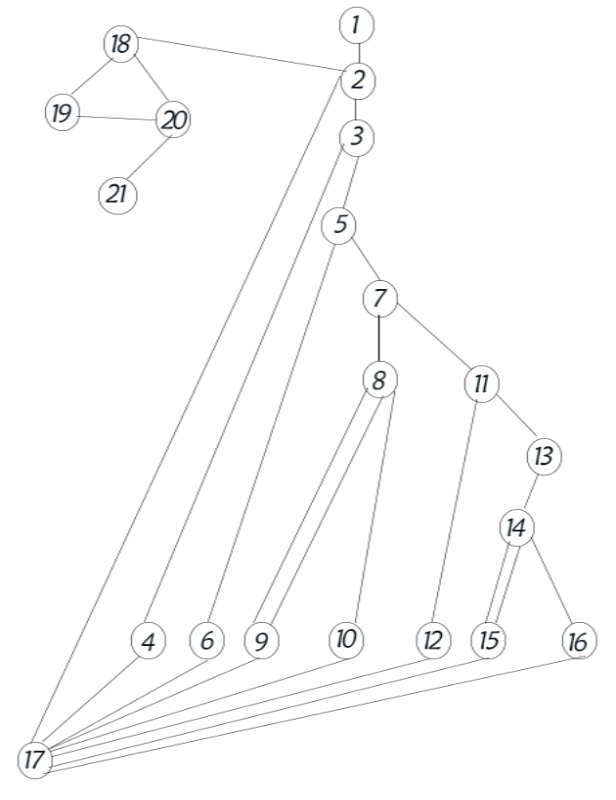
****

RPN.toStringArray

3. if (Character.isDigit(c))

5. if (digit.length() != 0)

10. if(digit.length() != 0)

****

RPN.transformToRPN

3. if(Information.isDigit(i))

5.if(operator.empty()||i.equals("(") || operator.peek().equals("("))

7. if(i.equals(")"))

11.if(Information.priority(i)>Information.priority(operator.peek()))

14.if(Information.priority(i) <= Information.priority(operator.peek()))