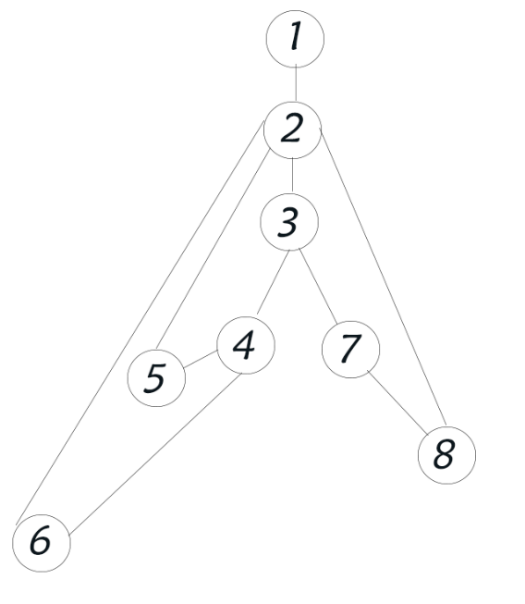
**单元测试：**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 测试用例 | 预期结果 | 测试结果 |
| Exes.generateExes  Param1:numOfExes  Param2:limit  Param3:bracketExists | 100 10 YES | 生成一个List题目集 | 通过 |
| 10 20 YES |
| 100 10 NO |
| 10 20 NO |
| -1 10 YES | 无法生成题目集 | 通过 |
| 10 -1 NO |
| 0 15 NO |
| 20 0 YES |
| Exes.gcd  Param1:num1  Param2:num2 | 5 10 | 5 | 通过 |
| 3 2 | 1 |
| 0 10 | 0 |
| -8 12 | 4 |
| -15 -20 | 5 |
| Exes.lcm  Param1:num1  Param2:num2 | 3 1 | 3 | 通过 |
| 21 7 | 21 |
| 0 5 | 0 |
| -4 7 | -28 |
| -4 -7 | 28 |
| Fraction.add  Param1:f1  Param2:f2 | “1/5” “2/3” | “13/15” | 通过 |
| “-5/7” “8/9” | “11/63” |
| “0” “1/4” | “1/4“ |
| “abc” “def” | NumberFormatException |
| Fraction.min  Param1:f1  Param2:f2 | “1/5” “2/3” | “-7/15” | 通过 |
| “-5/7” “8/9” | “-101/63” |
| “0” “1/4” | “-1/4“ |
| “abc” “def” | NumberFormatException |
| Fraction.mul  Param1:f1  Param2:f2 | “1/5” “2/3” | “2/15” | 通过 |
| “-5/7” “8/9” | “-40/63” |
| “0” “1/4” | “0” |
| “abc” “def” | NumberFormatException |
| Fraction.div  Param1:f1  Param2:f2 | “1/5” “2/3” | “3/10” | 通过 |
| “-5/7” “8/9” | “-45/56” |
| “0” “1/4” | “0” |
| “1/4” “0” | ArithmeticException |
| “abc” “def” | NumberFormatException |
| Fraction.strToFrac  Param：String1 | “1/4” | (1,4) | 通过 |
| “-2/3” | (-2,3) |
| “3/0” | ArithmeticException |
| “2” | (2,1) |
| “a/b” | NumberFormatException |
| Fraction.toString  Param:F1 | (1,4) | “1/4” | 通过 |
| (-2,3) | “-2/3” |
| (3,0) | ArithmeticException |
| (a,b) | NumberFormatException |
| Solution.InfixToSuffix  Param:Infix | “1+2\*3” | “1 2 3 \* +” | 通过 |
| “(1+2)\*3” | “1 2 + 3 \*” |
| “a+b\*(c-d)-e” | NumberFormatException |
| Solution.SuffixSolution  Param:Suffix | “1 2 3 \* +” | “7“ | 通过 |
| “1 2 + 3 \*” | “9 |
| “a b c \* +“ | 异常 |

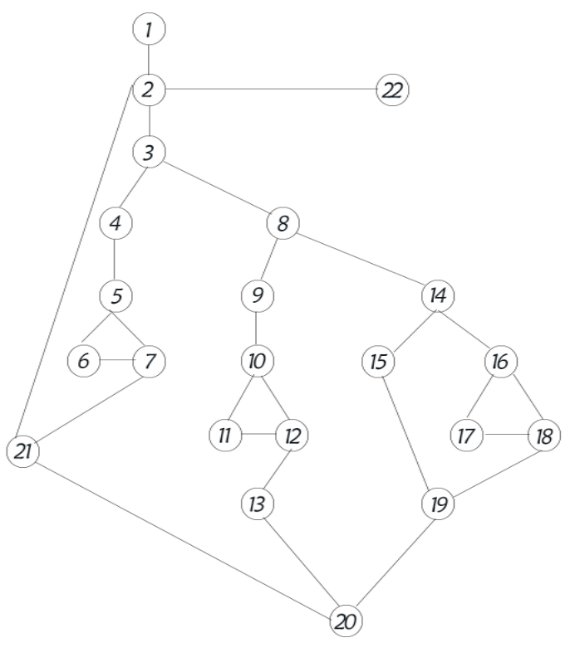
**关键函数流图：**

****

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))

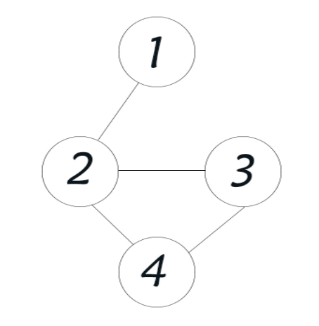
5. if(numOfTarget < 2)

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

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

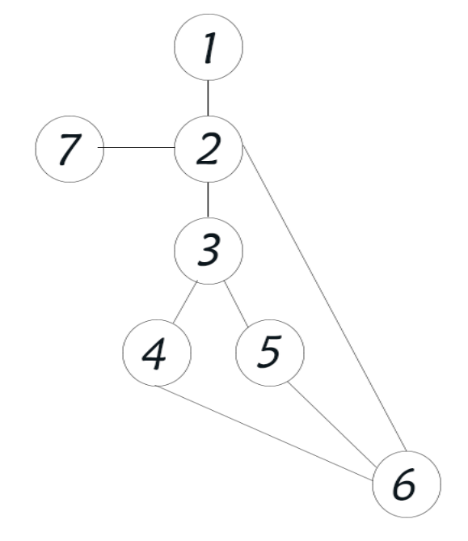
14. if(numOfTarget == 1)

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

****

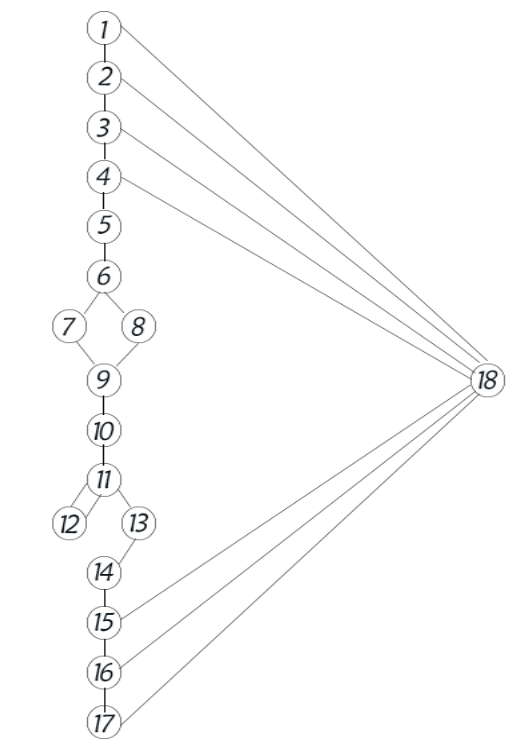
CompeteRPN.add/minus/multiply

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)

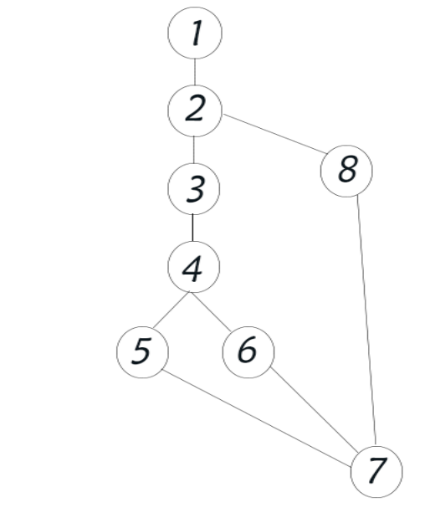
4.if(numerator+denominator==0)

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

15. if(denominator==1)

16. if(denominator==-1)

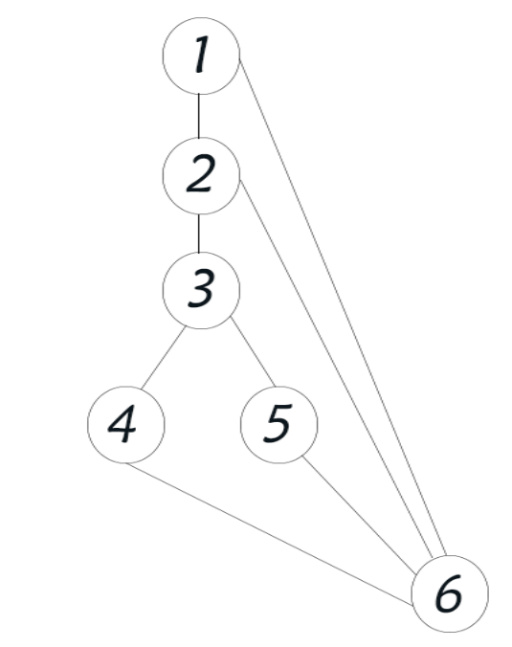
17.if(numerator\*denominator<0)

****

Fraction.transform

2. if(Fraction.isFraction(a))

4. if(endIndex == -1)

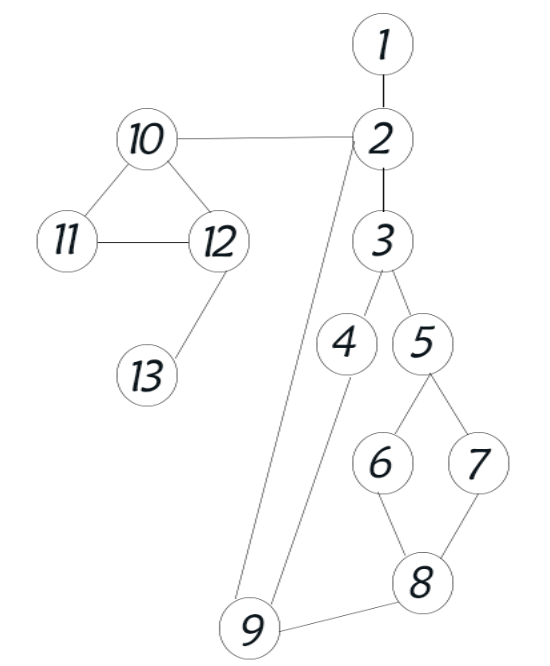
****

Information.vaildAnswer

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

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

3. if(Fraction.isFraction(answer))

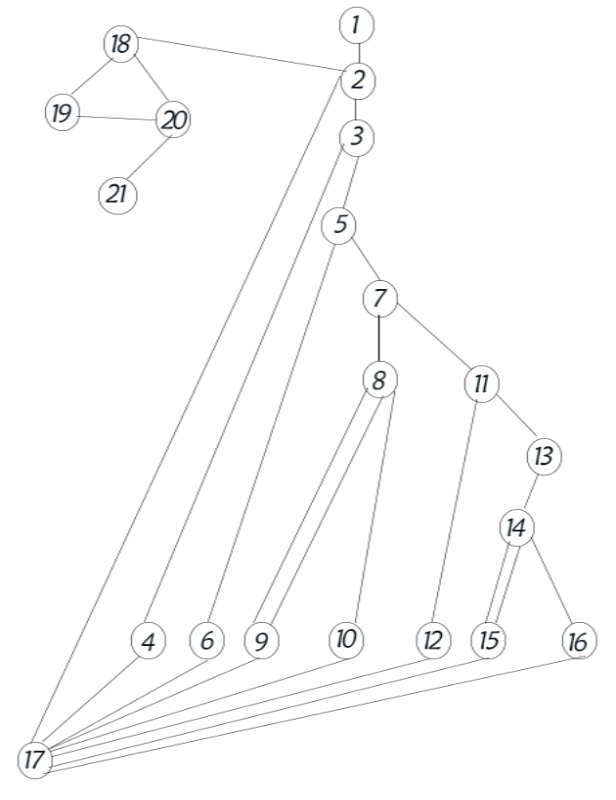
****

RPN.toStringArray

3. if (Character.isDigit(c))

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

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

****

RPN.transformToPrn

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()))