Code Structure Measurement Of SPL 1

SE 3204 – Software Metrics Lab

SUBMITTED TO

Dipok Chandra Das
Assistant Professor, IIT
Noakhali Science and Technology University

SUBMITTERD BY

Anupa Das Shormi (BFH1925020F)
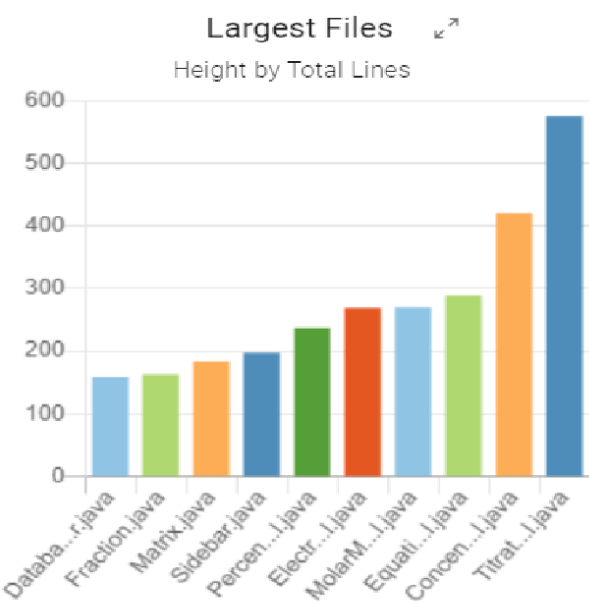Md.Rayhan Billah(ASH925028M)
Md.Rokonuzzaman(ASH1925018M)
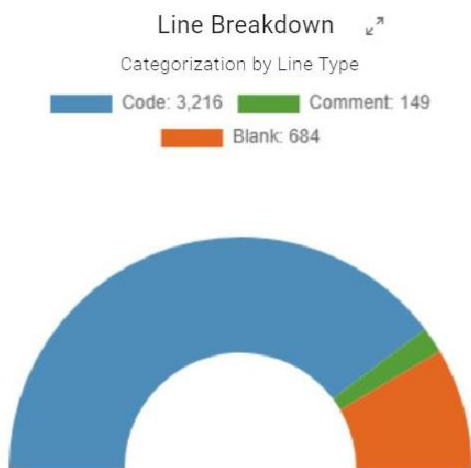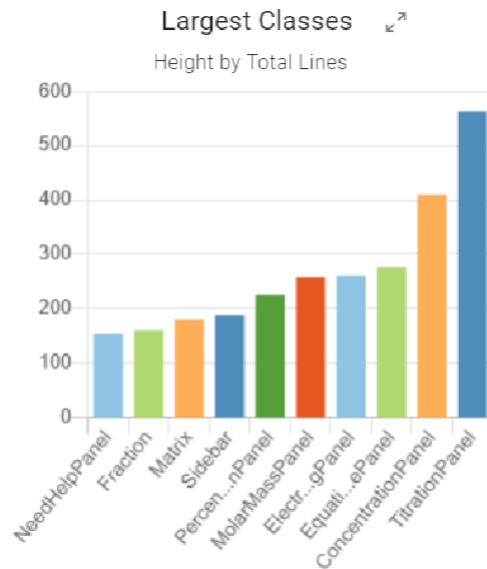
Project Overview

4,032	27	32	171
Lines	Files	Classes	Functions

Line Breakdown

Categorization by Line Type

Code: 3,216   Comment: 149   Blank: 684

Largest Files

Height by Total Lines

Largest Classes

Height by Total Lines
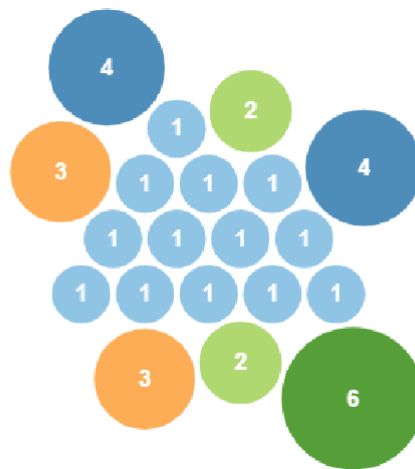
Most Complex Classes

1 = { Atom, Fraction, DatabaseSerializer, CompoundManager, compound**,** EquationBalancePanel, Concentration, Converter Inner class, Fraction, Exception classes**}**

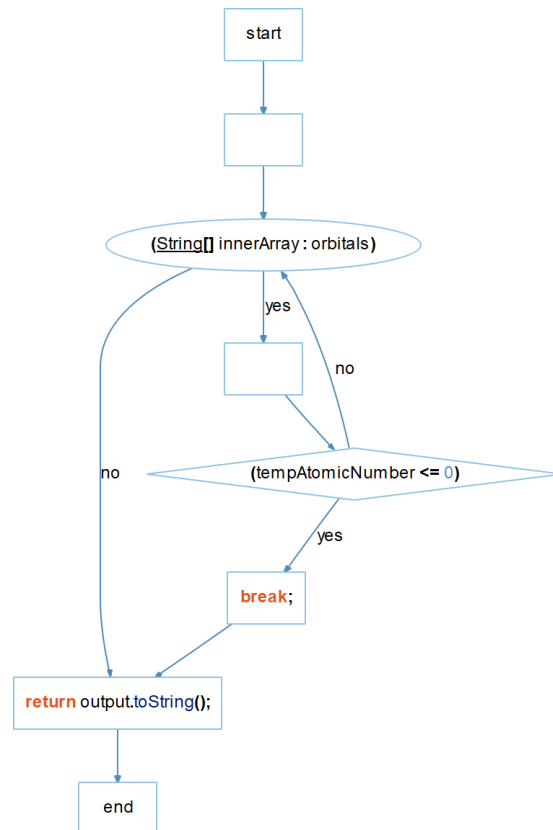**2 = {** ConcentrationPanel, Converter **}**

3 = { Matrix**,** Titration **}**

4 = { Formater, TitrationPanel **}**

6 = { EquationBalancer **}**

Class: Atom

1. Method name: getElectronConfig

```
                           start

                           [   ]

                  (String[] innerArray : orbitals)
                              |
                            yes
                              |
                           [   ]
                                    no
                  (tempAtomicNumber <= 0)
        no                       |
                               yes
                               |
                            break;
                              |
        return output.toString();
                       |
                     end
```

Cyclomatic complexity = 3

2. Method name: getInstance

```
                        start
                          |
                   (isValid(symbol))
                    yes        no
                     |          |
        return new Atom(symbol);   [   ]
                          \       /
                           end
```
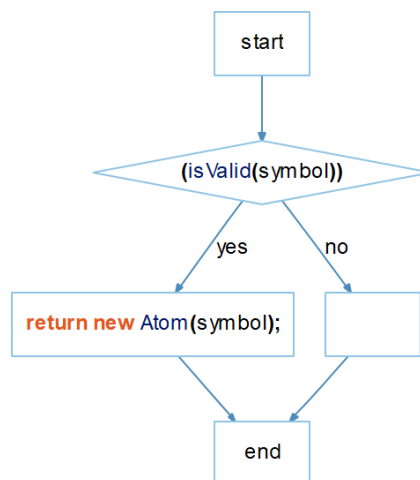
Cyclomatic complexity = 2

Class: Compound

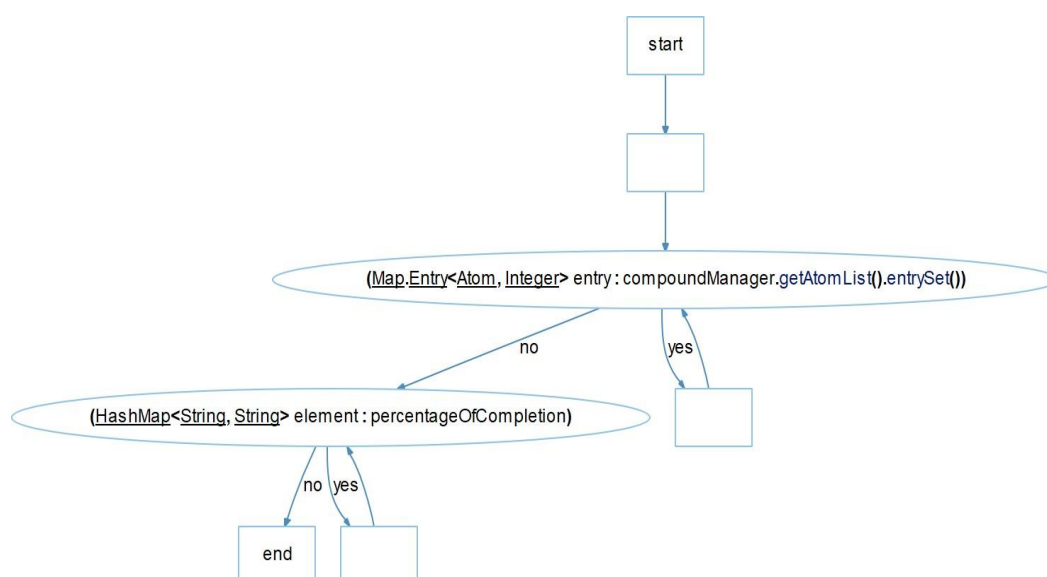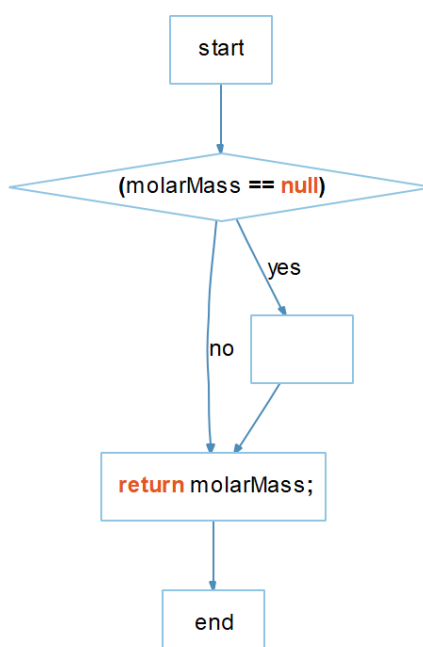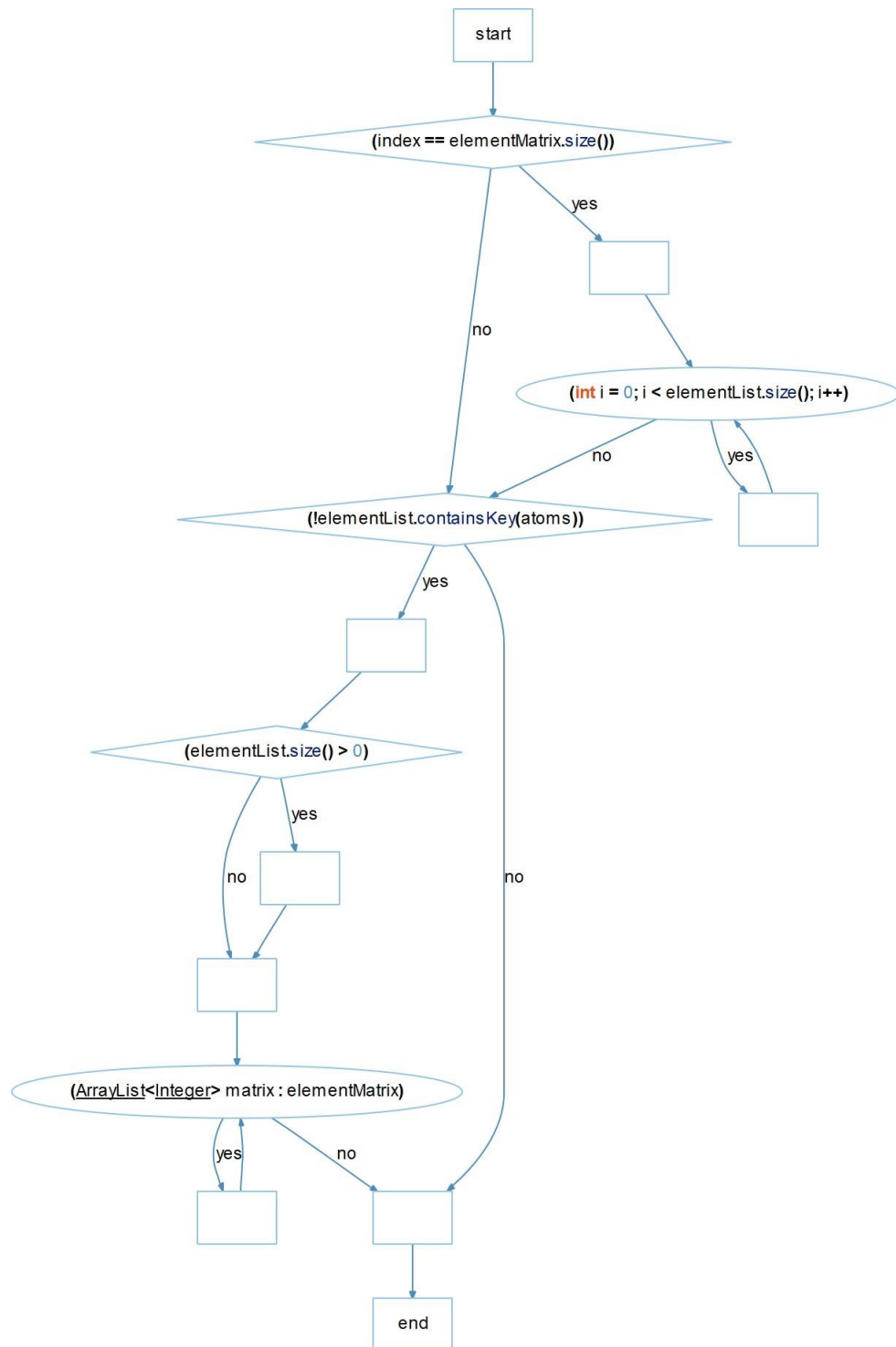3. Method name: calculatePercentageOfCompletion

Cyclomatic complexity= 3
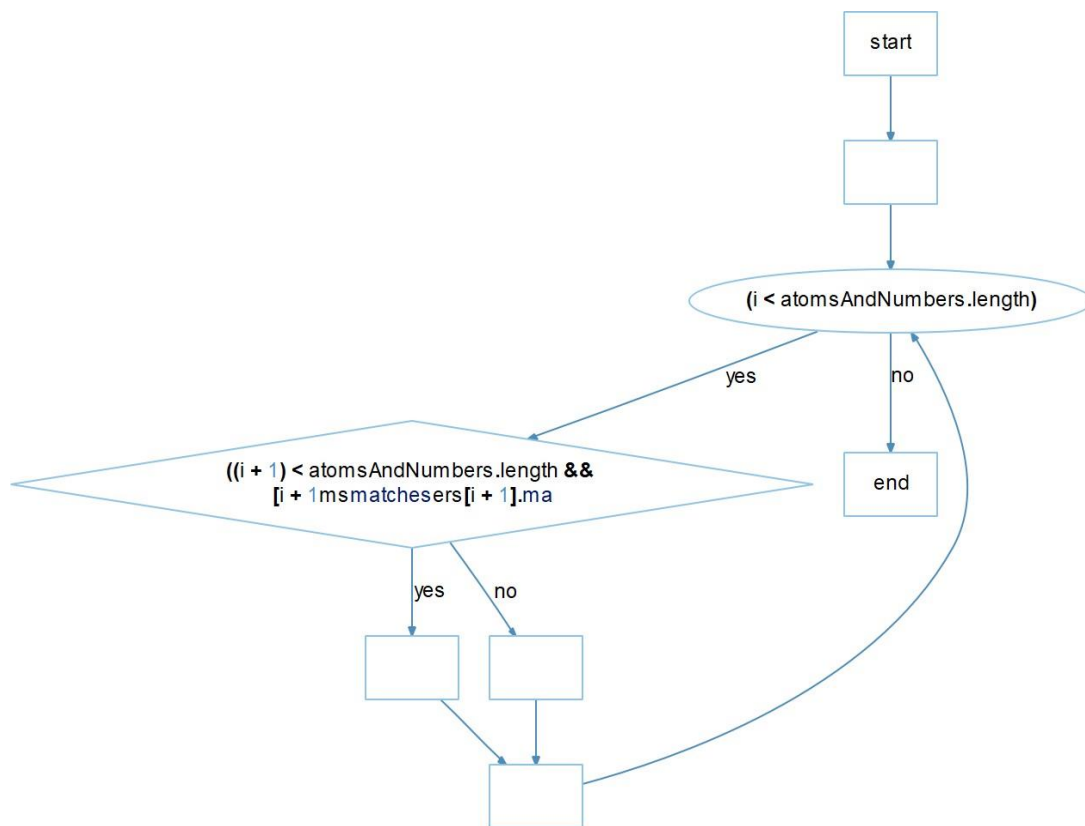
4. Method name: getMolarMass

Cyclomatic complexity= 2

Class: CompoundManager

5. Method name: addToMatrix
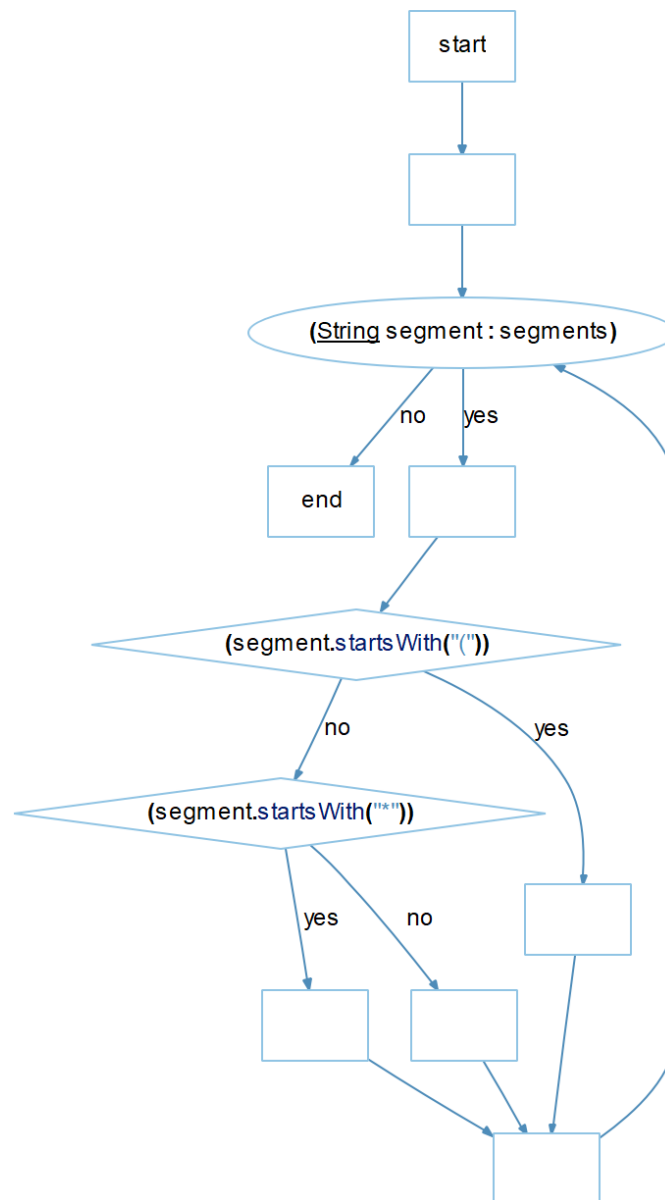
```
                              ┌─────────┐
                              │  start  │
                              └─────────┘
                                   │
                                   ▼
              ◇(index == elementMatrix.size())◇
                   │                    │
                   │ no             yes │
                   │                    ▼
                   │                 ┌──────┐
                   │                 │      │
                   │                 └──────┘
                   │                    │
                   │                    ▼
                   │      (int i = 0; i < elementList.size(); i++)
                   │           │ no              yes │
                   ▼           ▼                     ▼
        ◇(!elementList.containsKey(atoms))◇       ┌──────┐
              │ yes              │ no             └──────┘
              ▼                  │
          ┌──────┐               │
          │      │               │
          └──────┘               │
              │                  │
              ▼                  │
        ◇(elementList.size() > 0)◇
           │ no       │ yes      │
           │          ▼          │
           │      ┌──────┐       │
           │      │      │       │
           │      └──────┘       │
           ▼          │          │
        ┌──────┐◄─────┘          │
        │      │                 │
        └──────┘                 │
           │                     │
           ▼                     │
   (ArrayList<Integer> matrix : elementMatrix)
       │ yes        │ no         │
       ▼            ▼            │
   ┌──────┐     ┌──────┐◄────────┘
   │      │     │      │
   └──────┘     └──────┘
                   │
                   ▼
                ┌─────┐
                │ end │
                └─────┘
```

Cyclomatic complexity= 6

6. Method name: findAtoms

```
                                    ┌─────────┐
                                    │  start  │
                                    └─────────┘
                                         │
                                    ┌─────────┐
                                    │         │
                                    └─────────┘
                                         │
                          ╭──────────────────────────╮
                          │ (i < atomsAndNumbers.length) │
                          ╰──────────────────────────╯
                   yes  ╱                  │ no
                       ╱                   │
        ◇ ((i + 1) < atomsAndNumbers.length &&        ┌─────┐
          [i + 1msmatchesers[i + 1].ma                │ end │
                                                       └─────┘
          yes  │      │ no
               │      │
          ┌─────┐  ┌─────┐
          │     │  │     │
          └─────┘  └─────┘
                │    │
             ┌─────┐
             │     │
             └─────┘
```

Cyclomatic complexity = 3

7. Method name: compoundDecipher

start

(String segment : segments)

no   yes

end

(segment.startsWith("("))
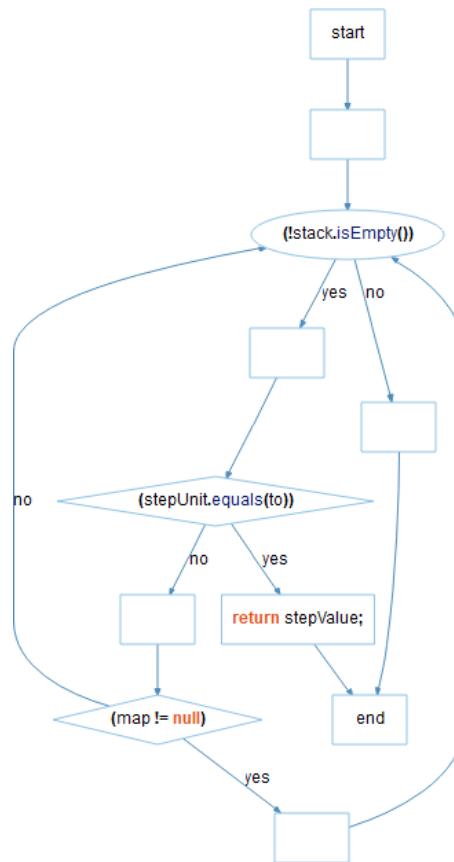
no   yes

(segment.startsWith("*"))

yes   no

Cyclomatic complexity = 4

Class: Converter
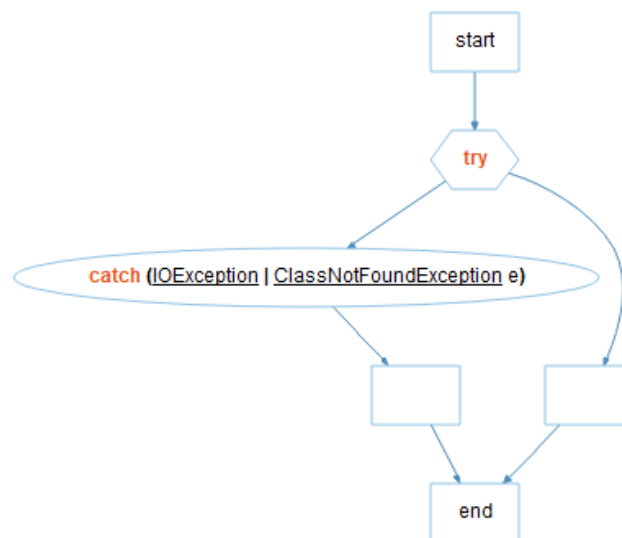
8. Method name: convert

Cyclomatic complexity = 4

Class: Database

9. Method name: Database
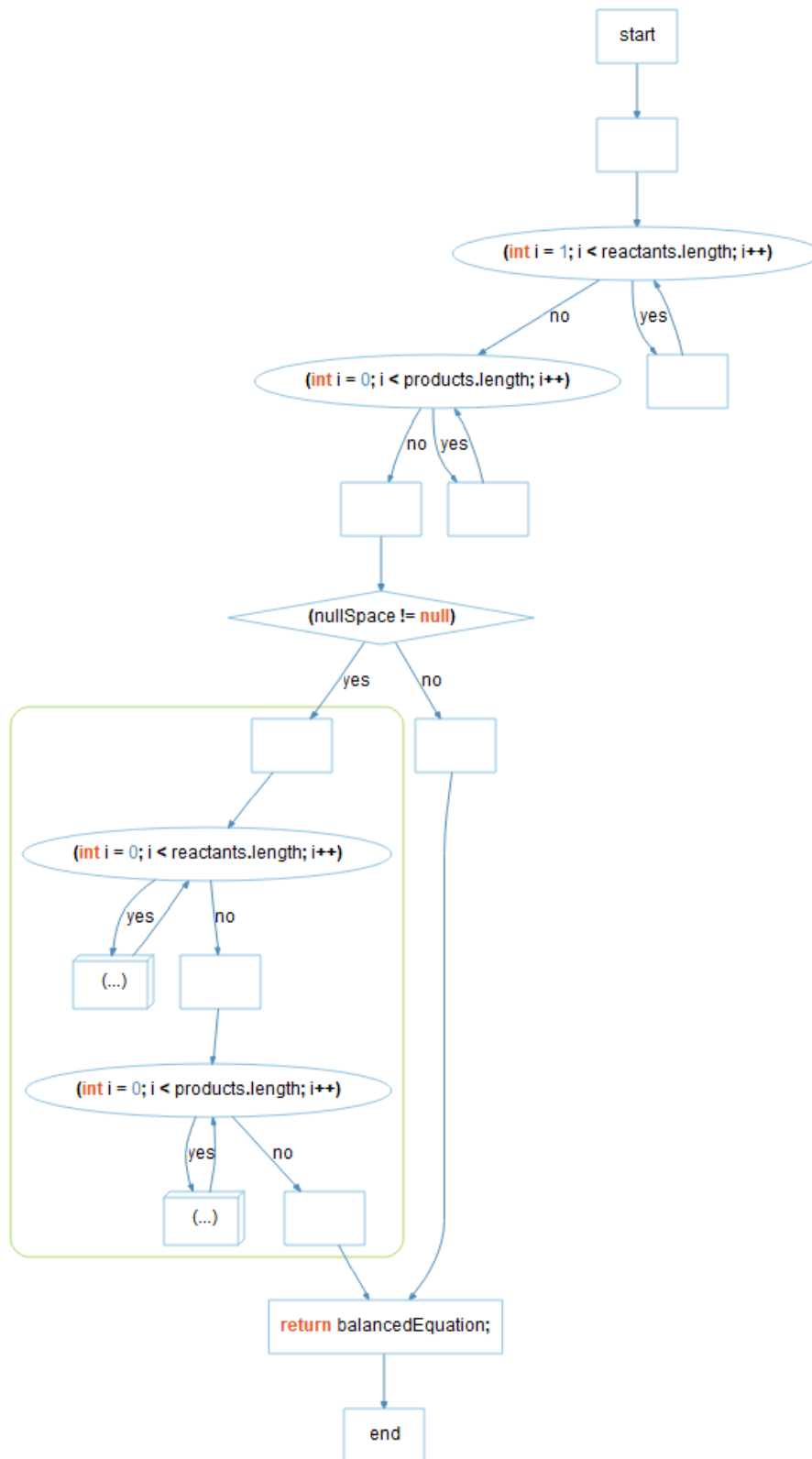

Cyclomatic complexity = 2

Class: EquationBalancer
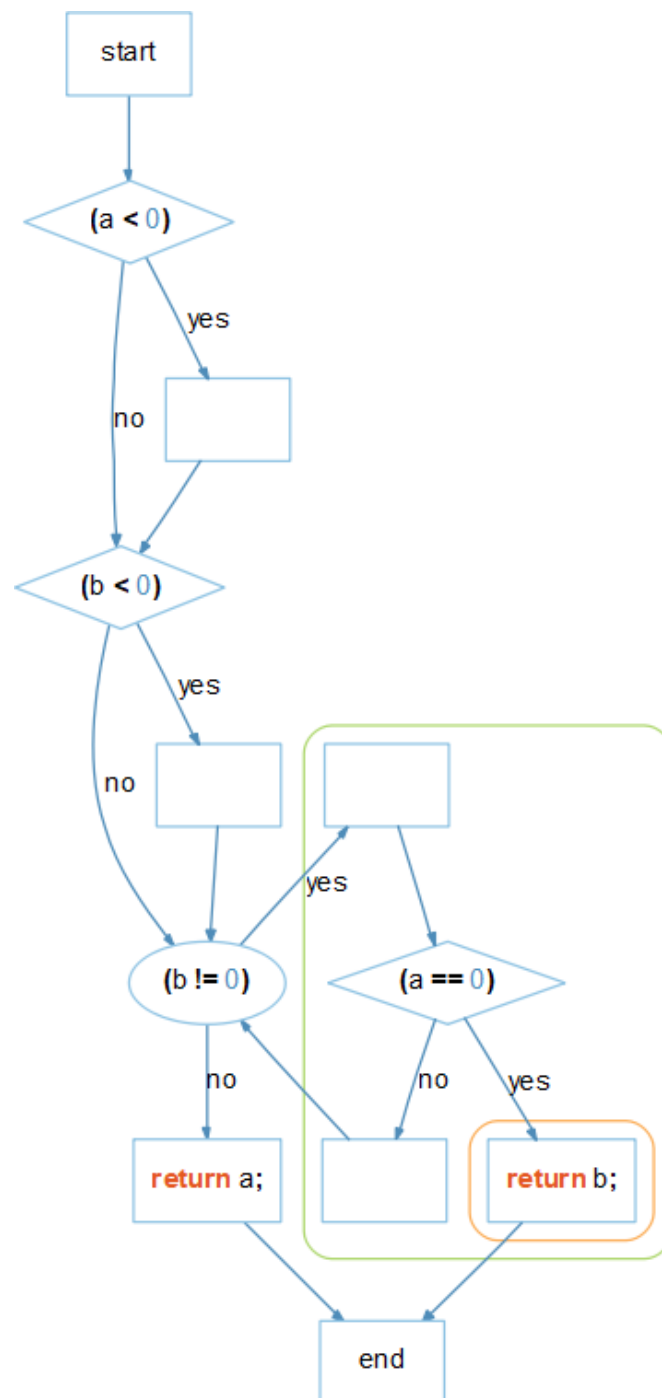
10. Method name: balance

start

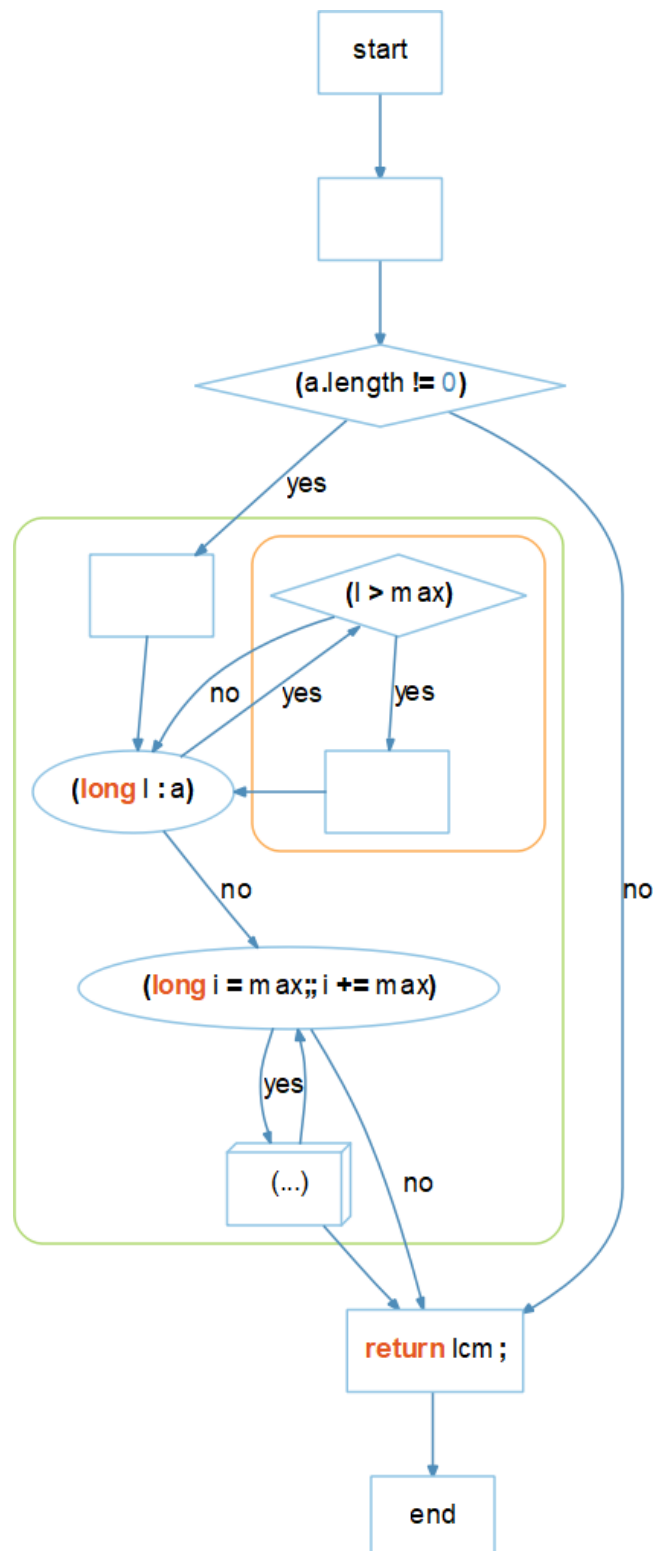(int i = 1; i < reactants.length; i++)
no   yes

(int i = 0; i < products.length; i++)
no   yes

(nullSpace != null)
yes   no

(int i = 0; i < reactants.length; i++)
yes   no
(...)

(int i = 0; i < products.length; i++)
yes   no
(...)

return balancedEquation;

end

Cyclomatic complexity = 12

Class: Fraction

11. Method name: getGCD

Cyclomatic complexity = 5

12. Method name: getLCM

Cyclomatic complexity = 8
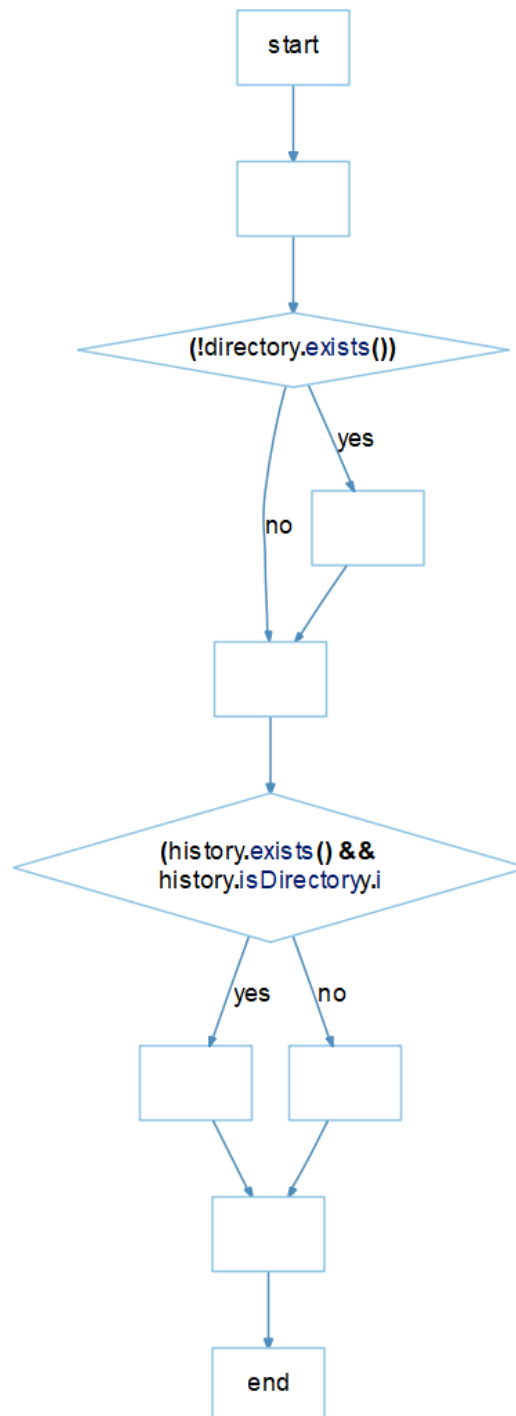
13. Method name: normalize

start

(int i = 0; i < fraction.length; i++)

no   yes

(fraction[i].
) != 0)

yes   no   yes

(int i = 0; i < result.length; i++)

no

return result;

end
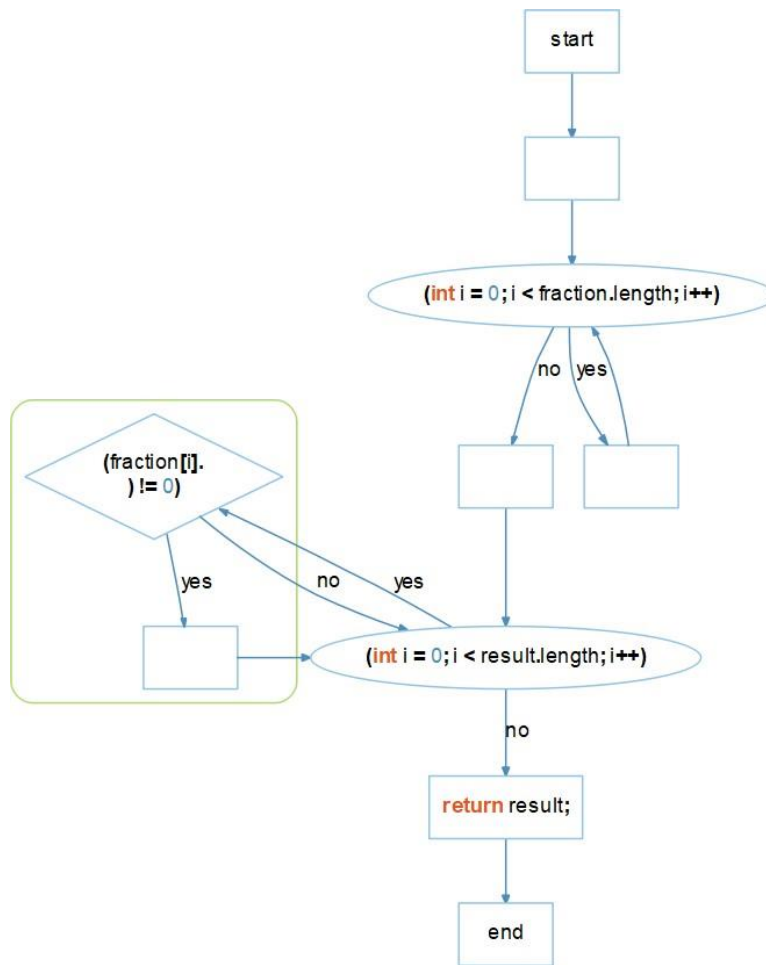
Cyclomatic complexity = 4

Class: History

14. Method name: Add

```
                              ┌──────────┐
                              │   start   │
                              └──────────┘
                                   │
                              ┌──────────┐
                              │          │
                              └──────────┘
                                   │
                          ◇ (!directory.exists()) ◇
                                │        │
                              no│        │yes
                                │     ┌──────────┐
                                │     │          │
                                │     └──────────┘
                                │        │
                              ┌──────────┐
                              │          │
                              └──────────┘
                                   │
                    ◇ (history.exists() &&       ◇
                      history.isDirectoryy.i
                          │           │
                       yes│           │no
                    ┌──────────┐  ┌──────────┐
                    │          │  │          │
                    └──────────┘  └──────────┘
                          │           │
                          └──────────┘
                              │
                          ┌──────────┐
                          │          │
                          └──────────┘
                              │
                          ┌──────────┐
                          │   end     │
                          └──────────┘
```
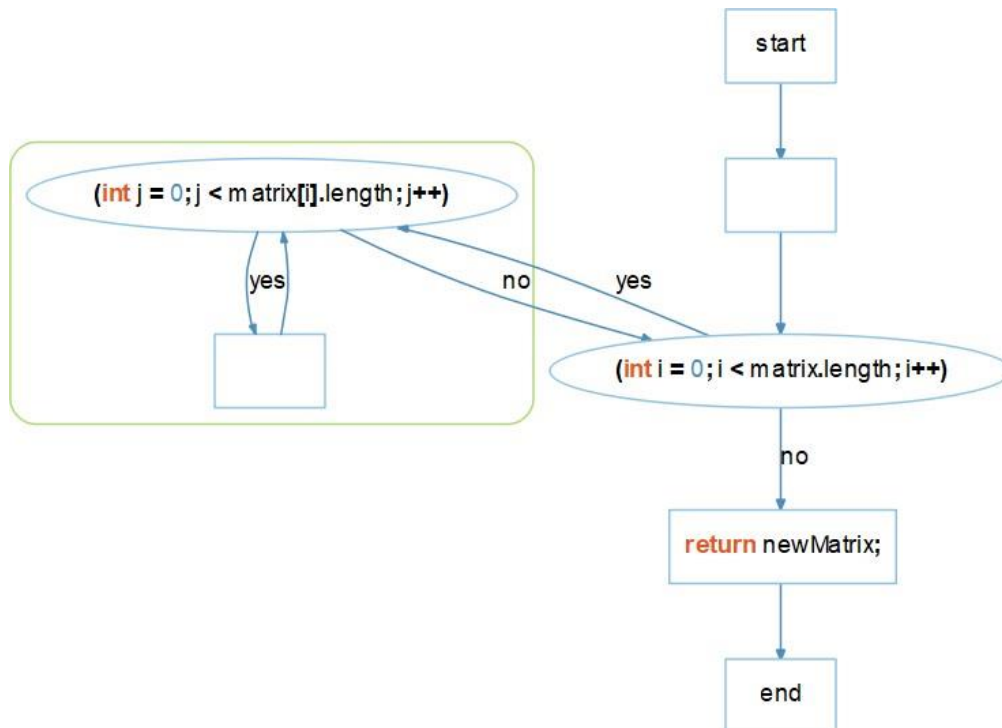
Cyclomatic complexity = 3

Class: Matrix

15. Method name: intToFraction

Cyclomatic complexity = 3

16. Method name: nullSpace


```
start
        ↓
    [          ]
        ↓
(echelonFormed.getHeight() > echelonFormed.getWidth())
        │ yes                              │ no
        ↓
(int k = 0; k < echelonFormed.getWidth(); k++)
        │ yes        │ no
        ↓
      (...)
        ↓
return echelonFormed.nullSpaces;
        ↓
      end
```
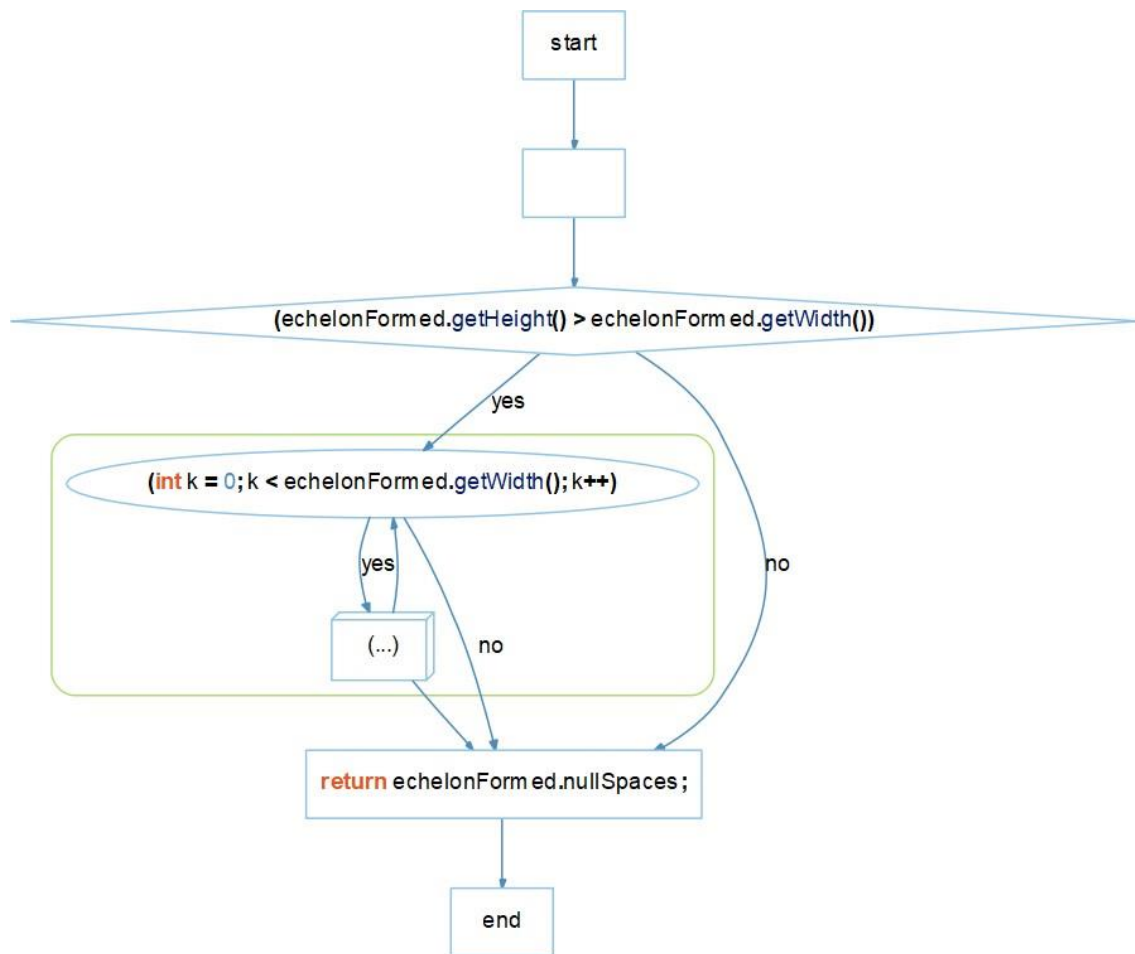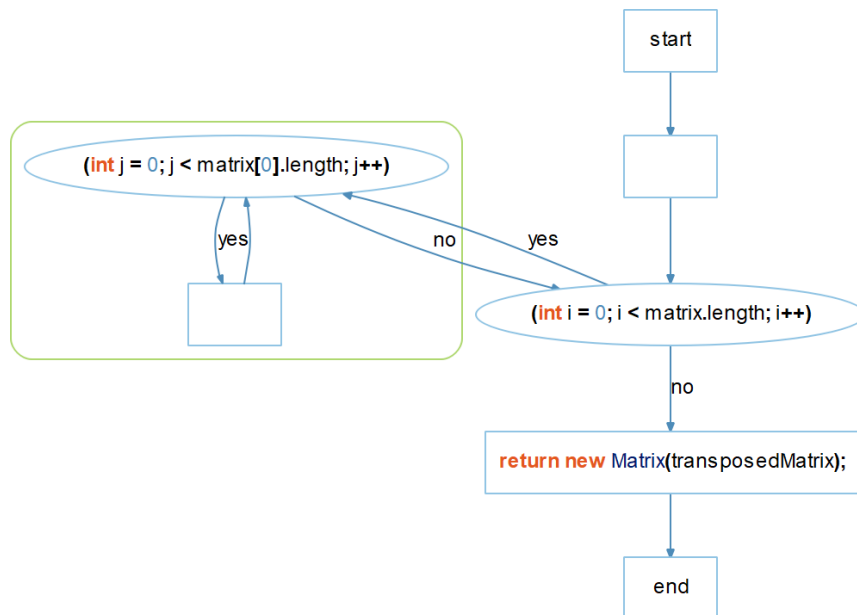
Cyclomatic complexity = 12

17. Method name: gaussianElimination
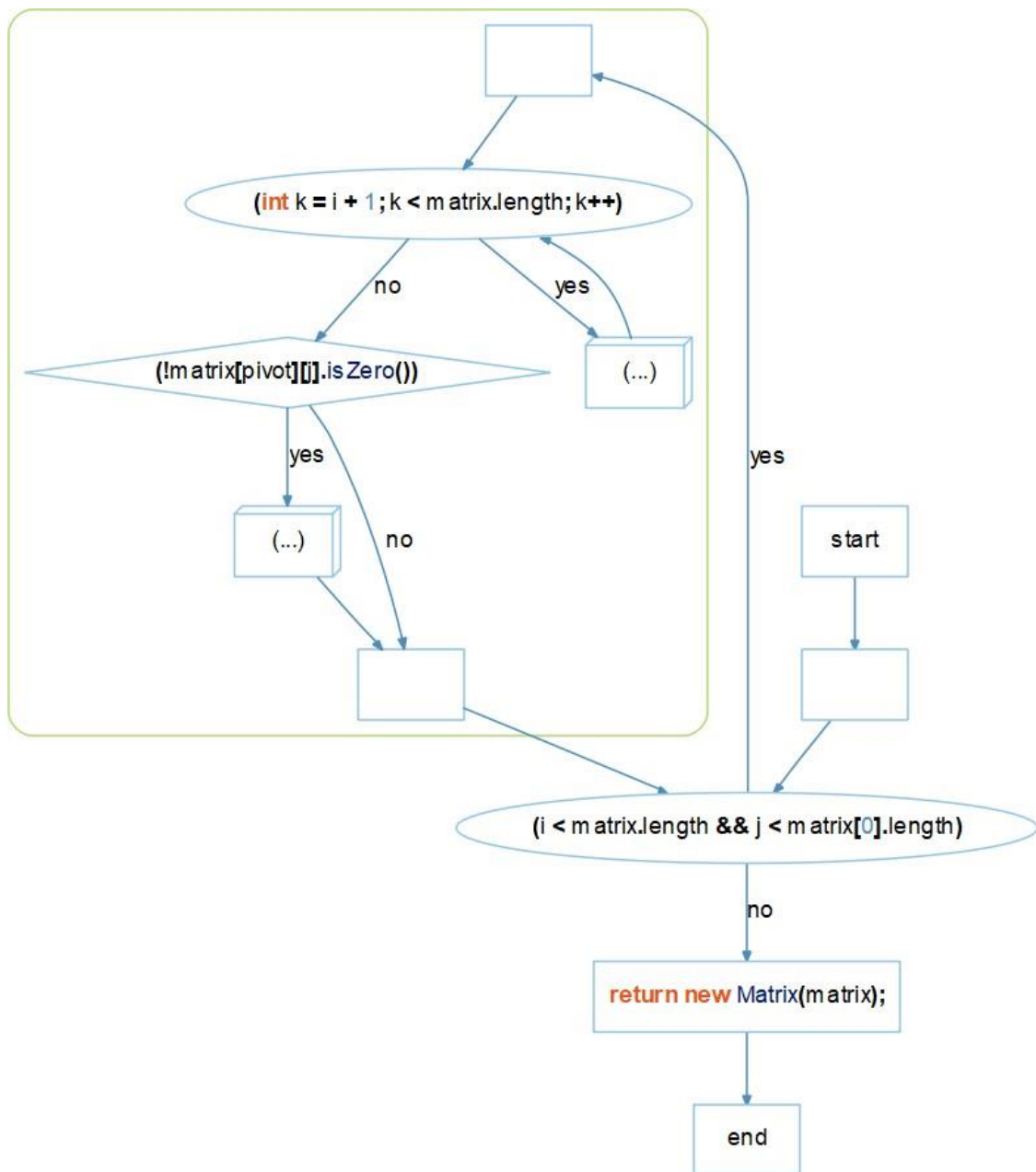
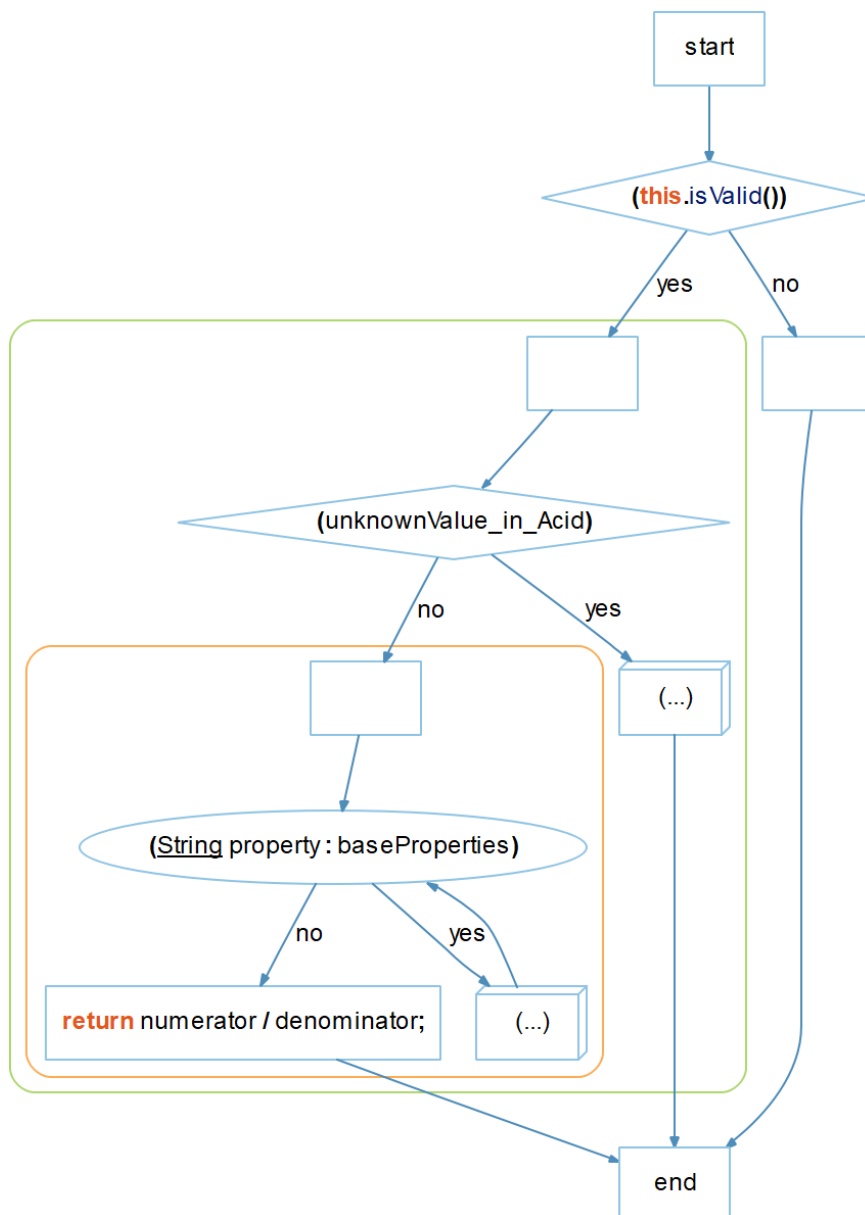Cyclomatic complexity = 9

18. Method name: transpose

Cyclomatic complexity = 3

19. Method name: backSubstitute
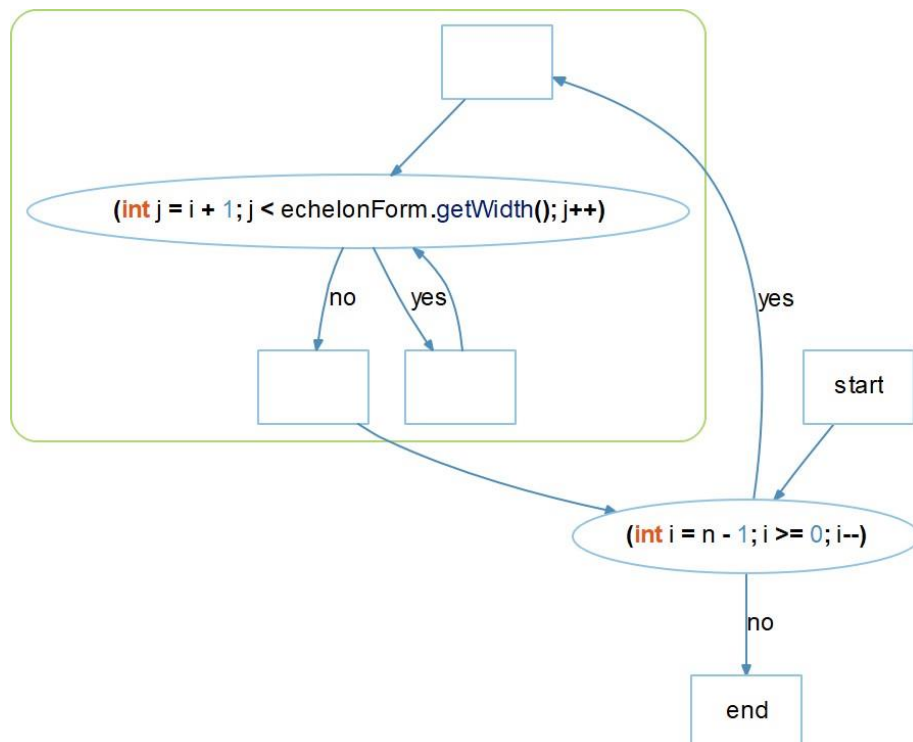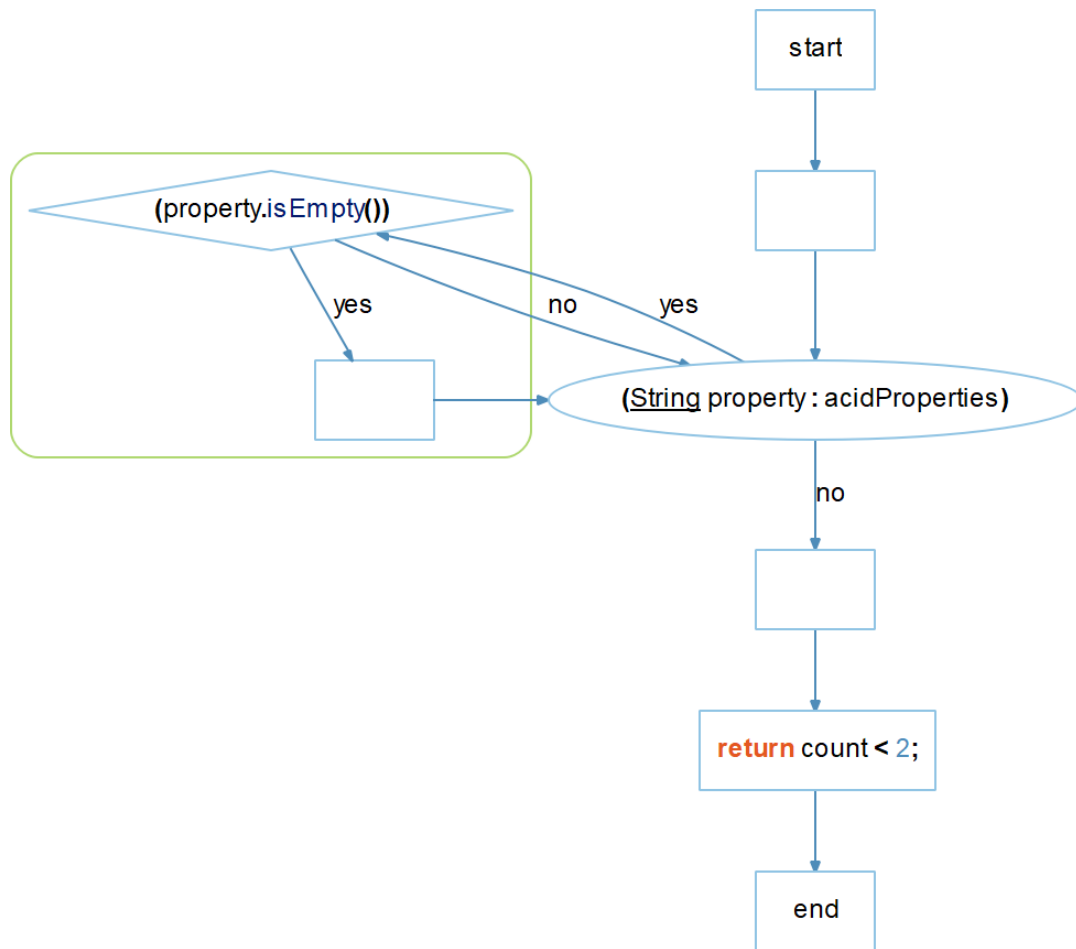

Cyclomatic complexity = 3

Class Titration

20. Method name: getUnknownValue

Cyclomatic complexity = 7

21. Method name: isValid

Cyclomatic complexity = 3