Project Team 2
Project-2A

Ben Gagliano, Zach Reid,
Rebekah Tippy,
&
Brad Mercurio

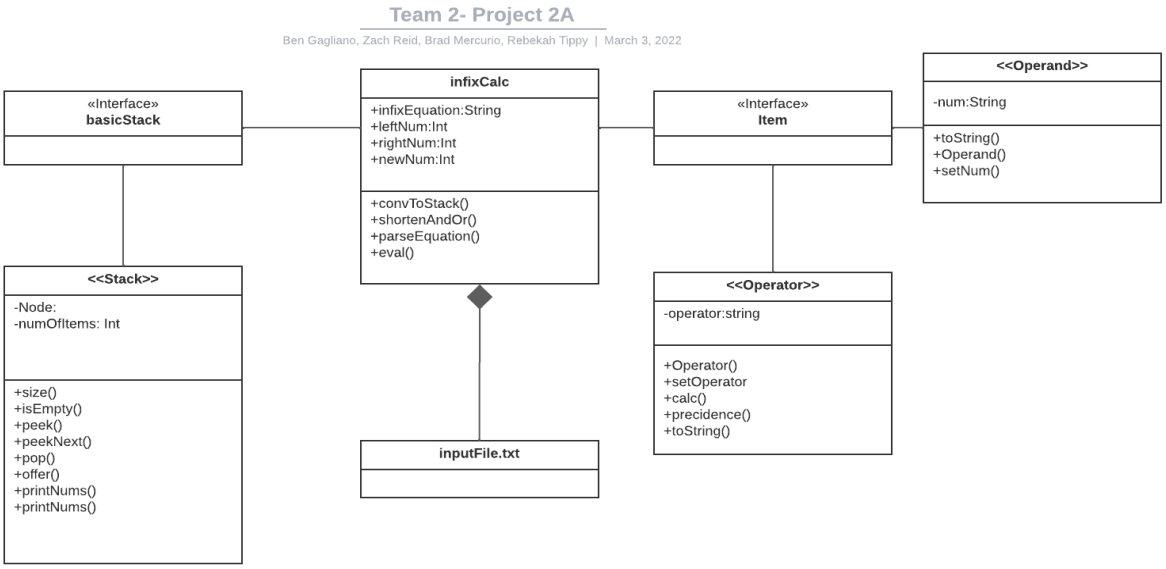
System Design

The program created is a calculator that takes a .txt file and outputs all the equations solved by utilizing stacks that read and calculate the elements input file. The calculator allows the user to input the supplied .txt file, to solve arithmetic and logical equations.

The main class is infixCalc, consisting of the stacks and the methods that calculate and output the answers. The Stack class basicStack interface provides the required methods to manipulate the stacks in the system. The Operand and Operator Classes are joined together by the Item Interface, providing the methods used to read the equations in the .txt file.

The data structures used are a multitude of stacks that are utilized by the methods calculating the equations.

UML Class Diagram



Test Results and Analysis

Expected Results:

1 + 2 * 3	= 7
2 + 2 ^ 2 * 3	=14
1 == 2	=0
1+3 > 2	=1
(4>=4) && 0	=0
(1+2) *3	=9
2%2+2^2-5*(3^2)	=-41
2^3	=8
6*2	=12
6-2	=4
6>5	=1
6 != 5	=1
6 > 5 && 4 > 5	=0
1 || 0	=1
(3+4) || 1	=1
(2 > 3) -2	=-2
5 ^ 2 % 7 && (4 -4)	=0
3 / (6 * 5 - 30)	=error divide by 0

Test 1:

```
1 + 2 * 3
2 + 2 ^ 2 * 3
1 == 2
1+3 > 2
(4>=4) && 0
(1+2) *3
2%2+2^2-5*(3^2)

2^3
6*2
6-2
6>5
6 != 5
6 > 5 && 4 > 5
1 || 0

(3+4) || 1
(2 > 3) -2
5 ^ 2 % 7 && (4 -4)

3 / (6 * 5 - 30)
```

Results of test 1:

7

14

0

1

0

9

-41

8

12

4

1

1

0

1

1

-2

0

error divide by 0

Test 2:

1 + 5 * 2
2 + 4 ^ 2 * 4
2 == 2
16+3 < 5
(5>=5) && 0
(1+2) *2
4%2+2^2-5*(3^2)

6^2
10*2
71-2
6 < 5
7 != 5
7 > 5 && 8 > 5
1 || 0

(5+4) || 1
(5 > 3) -2
4 ^ 2 % 7 && (4 -4)

(3 / (6 * 2 - 12))-2

Results test 2:

11

66

1

0

0

6

-41

36

20

69

0

1

1

1

1

-1

0

error divide by 0

Analysis:

The system works just as expected, pulling the equations from the .txt file and outputting all results accurately.

Project Conclusion

Overall, this code works remarkably well, calculating any equation and spitting out an accurate answer. Future edits to make would be to separate infixCalc and main() and to combine the stacks so only two are needed to be utilized. We would also eliminate the usage of the interfaces since in these situations they seem to be a bit more unnecessary. As alway, we can redesign it so the time complexity is faster. Lastly, adding a GUI and creating a user interface could make the program more versatile and user friendly.

Team Member Contribution

Ben Gagliano - Team Lead

- Project Management
- System/UML Design(Report)
- Report Design
- Github Troubleshooter/infixCalc Troubleshooter

Zach Reid - Programmer/Lead Troubleshooter

- System Structure & Design(Code)
- Stack Class & basicStack Interface
- Program Editor/Troubleshooter

Brad Mercurio - Data Analyst

- Analysis and Test Results
- .txt File Design
- infixCalc Troubleshooter

Rebekah Tippy - Lead Programmer

- System Structure & Design(Code)
- Operand & Operator Classes/Item Interface

All Contributed to Class infixCalc

We experienced major difficulties with the initial github repo due to conflicting methods of editing and adding to the repo. The error caused the zip file to load incorrectly so we had to take the code and make a brand new project and edit it to the right version of java. After 5 hours of troubleshooting we fixed the errors in the program and successfully got it to work.