SDF-Project1-Report

Harshit Daheriya

May 2025

Design

To design the algorithms, I adopted foundational methods inspired by standard arithmetic procedures:

Addition

For integer addition, I implemented the traditional algorithm taught in primary school—adding digits from right to left, while carrying over when sums exceed a single digit.

Subtraction

Similarly, subtraction follows the school-level method of subtracting digits from right to left, with borrowing when the digit being subtracted is greater than the current digit.

Multiplication

I used the classic multiplication technique: multiply each digit of the second number with the first, shift appropriately (based on position), and sum the intermediate results to obtain the final product.For Float Multiplication I first converted the given float into and integer by removing decimal and then used AInteger multiplication function to compute the result to float by again adding decimals at proper place.

Division

For division, I implemented the Newton-Raphson algorithm, a numerical method I learned in the third-semester course on Numerical Methods. This allows for efficient and accurate division, especially for arbitrary-precision floating-point numbers. This algorithm works by initially guessing the reciprocal of the second argument and then iterating multiple times to improve the estimate to the desired accuracy.Newton-Raphson is implemented in AFloat class where it computes the result with accuracy of 1000 digits after the decimal point. To

compute the division of two AInteger we pass them to AFloat Division and just take the part before the decimal for our answer.

UML Diagrams

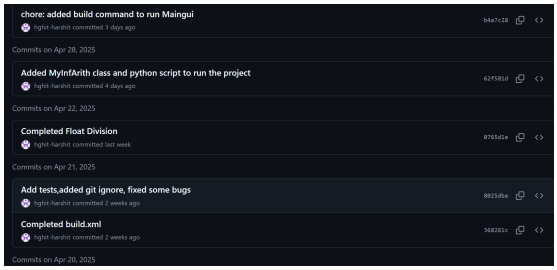
AInteger
m_value : String - m_sign : int # m_digits : int
+ Add(other : AInteger) : AInteger - Compare(other : AInteger) : int + Sub(other : AInteger) : AInteger + Mult(other : AInteger) : AInteger + Div(other : AInteger) : AInteger + Print() : void + GetValue() : String

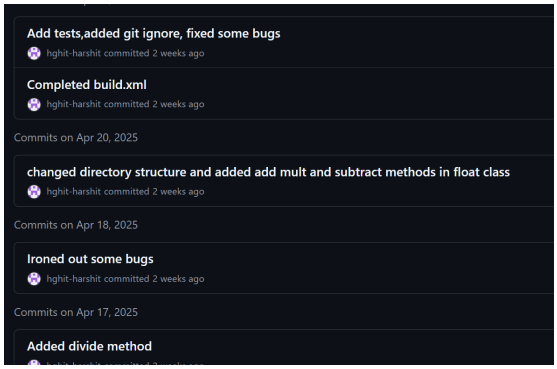
AFloat
m_value : String - m_sign : int # m_digits : int
+ Add(other : AFloat) : AFloat - Compare(other : AFloat) : int + Sub(other : AFloat) : AFloat + Mult(other : AFloat) : AFloat + Div(other : AFloat) : AFloat - isZero() : boolean + Print() : void + GetValue() : String

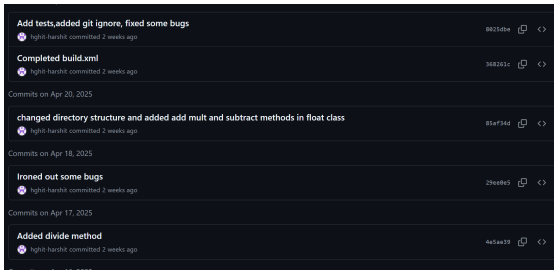
Key Learning

In this project I learned, first of all, how to code in Java. Coming from a C++ background, I did not have much trouble understanding the OOP aspects of the language, but the lack of pointers certainly forced me to think differently. I also learned how to use Git and Docker and how to use Apache Ant to build Java projects.

Git Snapshots




Add tests,added git ignore, fixed some bugs
hghit-harshit committed 2 weeks ago
Completed build.xml
hghit-harshit committed 2 weeks ago
Commits on Apr 20, 2025
changed directory structure and added add mult and subtract methods in float class
hghit-harshit committed 2 weeks ago
Commits on Apr 18, 2025
Ironed out some bugs
hghit-harshit committed 2 weeks ago
Commits on Apr 17, 2025
Added divide method


Add tests,added git ignore, fixed some bugs
hghit-harshit committed 2 weeks ago
8025dbe
Completed build.xml
hghit-harshit committed 2 weeks ago
368261c
Commits on Apr 20, 2025
changed directory structure and added add mult and subtract methods in float class
hghit-harshit committed 2 weeks ago
85ef34d
Commits on Apr 18, 2025
Ironed out some bugs
hghit-harshit committed 2 weeks ago
29ee0e5
Commits on Apr 17, 2025
Added divide method
hghit-harshit committed 2 weeks ago
4e5ae39