Function 7 : a^{b^x}

Deliverable 4

SOEN-6011 Software Engineering Processes

Desideratum implementation

The current version has the following implementation :

- Memento of the user interactions using Memento design pattern
- Displaying accurate user results
- Performing accurate results without by the core implementation of the transcendental function
- The calculation of Taylor series and logarithmic function
- Using User interface patterns to improve usability
- Supports both the GUI and command interface by changing execution
- Making a correct, efficient, maintainable, robust and usable system
- Maintaining an accuracy and precision with a normal calculator
- Handling both the cases of the integers and decimal powers to provide precise results
- Verifying inputs and outputs as real numbers.
- Proper domain verification

Current Version

- All tasks achieved
- Good reviews reported by fellow team mates
- Interactive GUI improved user experience
- Memento pattern help achieve user history

Critical Decisions [Formulas]

- 1. Decision: The implementation of the Taylor series helped with the precision of up-to 50 points of the decimal places by morphing the function a^{b^x} to two parts, specifically y = $e^{x*\log b}$ and z = $\exp^{y*\log a}$, where 'z' is the final result of the expression.
  Result: The implementation of the Taylor series helped achieved accurate results.
- Decision: The calculation of the log function with the base 10 instead of the natural log and converting the result.
  Result: The implementation of the log_{10} was much easier and with the simple formula $\log_e(x) = \frac{\log_{10}(x)}{\log_{10}(e)}$

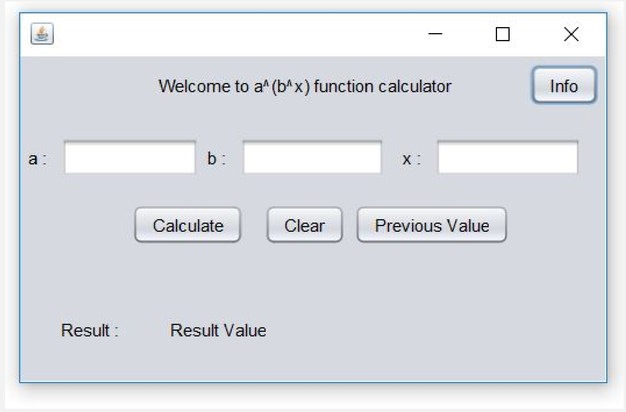
Critical Decisions [Coding]

- **GUI interface:** Code contains implementation of both command line and graphical interface. But ease of use GUI was preferred.
- **Memento Pattern:** The memorization of last calculation for the user's convenience aimed to make real world calculator.

Critical Decision

1. **Dual Implementation:** The results with normal Taylor series and logarithmic function provided irregular results with integer power. To get specific result the integer version of the code was implemented.
2. **Testing:** The aim of the calculator was to provide accurate results but every calculator is implemented in different ways. So for proper testing the decimal precision was set up to 5 and 4 in some cases.
3. **Exception Handling:** The decision was to print the exception messages as coder also so implementer can understand the specific working. Or to be user specific. The proper usage of 'JOptionPane' represented the message in the UI based interface.
4. **Rounding Digits:** Rounding digits for normal numbers had no problems but with large numbers the exponent value got dropped off. So, the decision was made to remove the rounding off numbers.
5. **Messaging:** Proper messages and information of the function was specified in the interface.

Graphical Interface



Improvements

- Rounding off Results : One of the future improvements were the rounding off the results.
- Memento: Improving the memento pattern to store the results of last 10 calculations and adding buttons to use value in new calculations.
- Interface : The code consists proper working for GUI and command line. An addendum would be to giver user the choice of working with the interface user wants.
- Exponent Dropping: New improvement for the code was to round off numbers without using internal libraries and find a way to manage the exponent
- Bounding Improvement: The biggest value in Java is double's max value, but improving max boundary further to have precise results.

Lessons Learned

- Implementation and learning of Memento Design Pattern
- Improving testing and java docs
- Core function implementation of Log function and Taylor Series
- Proper IEEE documentation methods
- Managing workings with team members.

Contact Information

- Himanshu Kohli [ID: 40070839, Team-E]
- GitHub: `https://github.com/Hkohli30/SOEN_6011`