Gebze Technical University
Department of Computer Engineering
CSE 241/501
Object Oriented Programming / Programming
Fall 2024
Homework # 1
Due date Nov 17 2024

Project Definition: ANSI Terminal-Based Spreadsheet Program

Overview

In this project, you will create a simplified spreadsheet program inspired by VisiCalc. The program will run in an ANSI terminal and use the provided AnsiTerminal class for displaying text and capturing user input. You will demonstrate your understanding of C++ fundamentals, OOP concepts, STL usage, and file handling.

The goal is to build a functional spreadsheet application that can handle basic calculations, support cell referencing, and save/load data in CSV format. The program will include essential spreadsheet features such as cell referencing, formulas, and automatic recalculation.

Requirements

The project will include the following requirements, which students should implement and document in a PDF file as part of their submission.

1. Documentation (PDF File)

- UML Class Diagram: Create and include a UML diagram representing the class structure of the spreadsheet application, showing relationships between key classes. Use plantuml.com for producing UML diagrams.
- Features Implemented: Describe the features implemented and clearly list any missing features.
- AI-Based Assistance: Note the sections of the code where any assistance was obtained from ChatGPT or other AI-based tools.
- User Manual: Include a user manual that describes how to use the program, complete with example commands and expected results.

2. Basic Calculations and Formulas

- Users should be able to enter numbers, strings and perform basic arithmetic calculations in cells (e.g., =A1 + B2 - C3).

- The program should support addition, subtraction, multiplication, and division within formulas. No parentheses are allowed.

3. Automatic Recalculation

- The spreadsheet should automatically update dependent cells when a cell value is modified. Ignore the cyclic reference problem.
- For example, if cell B2 contains =A1 + 5, changing the value of A1 should automatically recalculate the value of B2.

4. Column and Row Labels

- Rows will be labeled numerically (1, 2, 3, ...) and columns alphabetically (A, B, C, ..., AA, AB, ...), similar to typical spreadsheet software, to facilitate data tracking.

5. Cell Referencing

- Users can reference other cells within formulas using cell identifiers (e.g., B1 + C1).
- The program will support absolute cell referencing but not relative cell addressing.
- Supported functions:
  - SUM: Sums a range of cells, e.g., SUM(A1..A10).
  - AVER: Averages a range of cells, e.g., AVER(A1..A10).
  - STDDEV: Calculates the standard deviation of a range of cells, e.g., STDDEV(A1..A10).
  - MAX: Returns the maximum value in a range, e.g., MAX(A1..A10).
  - MIN: Returns the minimum value in a range, e.g., MIN(A1..A10).
- Example formula: B1 - MIN(A1..A10). No parentheses will be used in the formulas.

6. Visual Interface

- The user interface should resemble VisiCalc's layout as shown at https://www.pcjs.org/software/pcx86/app/other/visicalc/1981/.
- The top line displays the contents of the currently selected cell, and it will also function as the input area where users type commands and formulas.

7. File Operations (CSV Format)

- Implement functionality to save and load spreadsheet data in CSV format. This ensures compatibility with other spreadsheet applications such as Microsoft Excel.
- Save each row of the spreadsheet as a line of comma-separated values in the .csv file.
- Unicode characters do not need to be supported.

- File operations such as make new file, save file, load file, save file as should be implemented.

8. Data Storage

- Use a 2D std::vector to store cell data for the spreadsheet grid. Each cell in the vector will contain either a number, formula, or string value.

Submission Requirements

- Source Code: All source files, including the provided AnsiTerminal.h and AnsiTerminal.cpp files.
- Include a header file and a CPP file for each class.
- Documentation PDF:
  - UML Diagram of the class structure.
  - Description of implemented features and any missing parts.
  - Declaration of AI assistance, if applicable.
  - A User Manual explaining the usage of the program with examples.
- Do not use any functions from the standard C library (like `printf`), do not use C arrays. For math functions you may use standard C functions.
- Use C++ standard classes such as string, vector, list, etc.
- Use all the OOP techniques that we have learned in the lectures such as consts, C++11 features (range for loops, strong enums, auto keyword, decltype keyword, etc.)
- Do not forget to indent your code and provide meaningful comments.
- We will provide a number of CSV files to test your program
- You should submit your work to the Teams page using the instructions from the TAs.
- You will demo your homework online

Suggested Timeline

Week 1: Project setup, class diagram, and initial Spreadsheet and Cell class development.
Week 2: Implement formula parsing and cell referencing with the FormulaParser class.
Week 3: Develop FileManager class for CSV operations and finish remaining features. Also do testing, bug fixing, and final documentation.