FLCounter

Author's Note:

FilesLinesCounter is a form of exercise to gain practical coding skills and knowledge in the field of C ++.

M.K.

Overview

FLCounter is a program written in C++ that helps count the number of files, empty and non-empty lines in provided paths on your hard drive and their subfolders.

The library program uses Boost.Filesystem to search files on disk and calculate lines in each file.

Included Libraries

- **<boost/filesystem.hpp>**: provides operations on files and directories, such as path resolution, iteration, and manipulation.
- **<algorithm>**: contains a collection of functions for algorithms to work on ranges of elements.
- **<fstream>**: provides classes for reading from and writing to files.
- **<limits>**: contains a collection of constants that represent the maximum and minimum values that different data types can hold.
- **<string>**: provides classes for working with strings.
- **<vector>**: provides a container for storing a dynamic array of elements.
- **<iostream>**: provides input and output stream classes for working with the console.

Namespace

The code defines the fs namespace, which is an alias for the boost::filesystem namespace. The boost::filesystem library provides facilities for working with files and directories.

Structs

DataInput holds the entered paths, which are provided as command-line arguments. The constructor of DataInput takes a vector of fs::path objects as input.

DataOutput holds the results of the program execution. It has three members:

files_number: the total number of files found in the specified paths
num_non_empty: the total number of non-empty lines found in the specified paths
num_empty: the total number of empty lines found in the specified paths

Functions

The code defines three functions: **countFiles, countLines, and operator<<.**

countFiles takes an fs::path object as input and returns the number of files found in the specified directory and its subdirectories. If the specified path is not a valid directory, the function throws an invalid_argument exception.

countLines takes an fs::path object as input and returns a tuple containing the number of empty and non-empty lines found in the specified directory and its subdirectories. If the specified path is not a valid directory, the function throws an invalid_argument exception. The function uses std::for_each to iterate over all files in the directory and its subdirectories, and uses an std::ifstream object to read the contents of each file. It counts the number of empty and non-empty lines in each file using a loop that reads each line of the file and checks whether it is empty or non-empty. If the file cannot be opened, the function throws a runtime_error exception.

operator<< takes an output stream (std::ostream) and a DataOutput object as input, and sends the program output to the output stream. The function returns the output stream.

main takes in command line arguments, which are assumed to be paths to directories that the program will search for files and count the number of empty and non-empty lines in. The main function creates a DataInput object to hold the entered paths and a DataOutput object to hold the results of the execution. Then it calls the filesCounter function for each path in the DataInput object to count the number of files in each directory and adds these counts to the num_empty and num_non_empty variables in the DataOutput object. Finally, the main function prints the results held in the DataOutput object to the console.

If any exceptions are thrown during the execution of the program, the main function catches them and prints an error message to the console before exiting with a failure code.