|  |  |
| --- | --- |
| Программа для определения, являются ли все числа в последовательности числами Фибоначчи | |
| Внутренняя спецификация | |
| Студент | Вольвич Д. С. |
| Преподаватель | Якимов Г. А. |
| Сдано |  |

1. Назначение

Программа предназначена для определения, является ли заданная последовательность чисел числами Фибоначчи. Числа Фибоначчи — числовая последовательность, первые два числа которой являются 0 и 1, а каждое последующее за ними число является суммой двух предыдущих.

1. Описание структур данных

1.1 vector<int> sequence

Назначение: хранит последовательность чисел, извлечённую из строки.

Тип: динамический массив целых чисел (int).

Использование:

Передаётся в функцию isFibonacciSequence() для анализа корректности.

1.2 int errorIndex

Назначение: хранит индекс первого элемента последовательности, нарушающего правило Фибоначчи.

Тип: int.

Использование:

Используется для генерации пользовательского сообщения об ошибке.

1.3 ifstream inputFile и ofstream outputFile

Назначение: файловые потоки для чтения (input.txt) и записи (output.txt).

Типы:

ifstream — входной поток.

ofstream — выходной поток.

Использование:

Открытие, проверка доступности файлов, работа с содержимым.

1.4 string line

Назначение: временная переменная для построчного чтения из файла.

Тип: строка (std::string).

Использование:

Буфер для каждой строки, полученной через getline.

1.5 vector<string> allLines

Назначение: хранит все строки, считанные из входного файла input.txt.

Тип: динамический массив строк.

Использование:

Проверка количества строк во входном файле.

Проверка на пустой файл.

1.6 int num

Назначение: временная переменная для хранения каждого числа, извлекаемого из потока.

Тип: int.

Использование:

В цикле при разборе строки с числами.

3. Описание алгоритмов функций

1. Главная функция программы:

int main()

Выполняет чтение файла, вызов функции isFibonacciSequence, isValidLine, validateInput и вывод результата.

Возвращаемое значение: 0 – успешно; >0 – ошибка.

Алгоритм функции:

// Открываем входной файл

// Открываем выходной файл

// Если не существует входного файла, то выводим в консоль "Неверно указан файл с входными данными. Возможно, файл не существует." и возвращаем код ошибки 1.

// Если не удается создать выходной файл, то выводим в консоль "Неверно указан файл для выходных данных. Возможно указанного расположения не существует или нет прав на запись." И возвращаем код ошибки 2.

// Проверяем ошибки ввода при помощи функции validateInput.

// Проверяем последовательность чисел во входном файле на соответствие последовательности чисел Фибоначчи.

// Вызываем функцию isFibonacciSequence.

// Если она возвращает true, то записываем в output.txt "Это числа Фибоначчи".

// Иначе если она возвращает false, то записываем в output.txt "Это не числа Фибоначчи, потому что n-е число в данной последовательности не вычисляется как сумма двух предыдущих", где n = (errorIndex + 1).

// Возвращаем 0.

1. isValidLine(const string& line)

Проверяет, что входная строка содержит только цифры и пробелы.

Параметры:

const string& line - строка, которую нужно проверить

Возвращаемое значение: true – если строка корректна; false – если строка содержит недопустимые символы.

Алгоритм функции:

// Проходим по каждому символу строки

// Если символ не является цифрой или пробелом

// Считать строку некорректной (вернуть false)

// Иначе вернуть true

3. int validateInput(ifstream& inputFile, ofstream& outputFile, vector <int>& sequence)

Проверяет корректность входного файла.

Параметры:

inputFile – входной файл

outputFile – выходной файл

sequence – последовательность чисел

Возвращаемое значение: 0 – нет ошибок; 3 – входной файл пуст; 4 – более одной строки; 5 – недопустимые символы; 6 – меньше 2 чисел; 7 – больше 1000 чисел.

Алгоритм функции:

// Проверяем ошибки ввода.

// Считываем все строки из входного файла для проверки количества строк.

// Если входной файл пустой, то записываем в output.txt "Входной файл пустой." и возвращаем код ошибки 3.

// Если во входном файле больше одной строки, то записываем в output.txt "Программа принимает на вход файлы из одной строки." и возвращаем код ошибки 4.

// Проверяем при помощи функции isValidLine, есть ли во входном файле недопустимые символы.

// Если есть, то записываем в output.txt "Программа принимает на вход только целые неотрицательные числа." и возвращаем код ошибки 5.

// Проверяем, есть ли во входном файле отрицательные или нецелые числа.

// Если есть, то записываем в output.txt "Программа принимает на вход только целые неотрицательные числа." и возвращаем код ошибки 5.

// Проверяем количество чисел во входном файле.

// Если количество чисел во входном файле меньше двух, то записываем в output.txt "Минимальное количество чисел в последовательности - 2." и возвращаем код ошибки 6.

// Проверяем количество чисел во входном файле.

// Если количество чисел во входном файле больше тысячи, то записываем в output.txt "Максимальное количество чисел в последовательности - 1000." и возвращаем код ошибки 7.

// Если ошибок не обнаружено, возвращаем 0.

4. bool isFibonacciSequence(const vector<int>& sequence, int& errorIndex)

Проверяет, соответствует ли последовательность чисел последовательности чисел Фибоначчи.

Параметры:

const vector<int>& sequence – последовательность чисел

int& errorIndex – индекс первого ошибочного элемента

Возвращаемое значение: true – если последовательность корректна; false – если есть несоответствие.

Алгоритм функции:

// Проверка первого элемента, который должен быть равен нулю.

// Если первый элемент не равен нулю, то задать errorIndex=0 и вернуть false.

// Проверка второго элемента, который должен быть равен единице.

// Если второй элемент не равен единице, то задать errorIndex=1 и вернуть false.

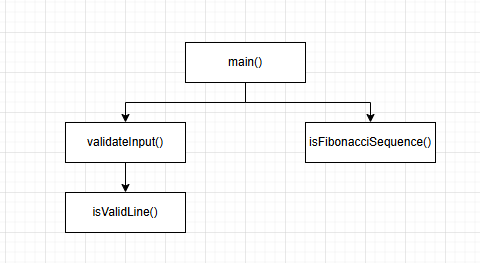
// Проверка следующих элементов.

// Проходим по всей последовательности.

// Если текущий элемент не равен сумме двух предыдущих, то задаем errorIndex, равный индексу данного элемента и возвращаем false.

// Если не было обнаружено никаких несоответствий, то возвращаем true.

4. Диаграмма вызовов функций.



5. Диаграмма потоков данных.

