| Программа для расшифровки текста | |
| --- | --- |
| Внутренняя спецификация | |
| Студент | Фролов Н.А. |
| Преподаватель | преп. каф. ПОАС Матюшечкин Д.С. |
| Сдано |  |
| Лабораторная работа №2 |  |

# 1 Общие сведения

Наименование программы – «Программа для распаковки строки».

Для функционирования программы необходима операционная система

Windows 8 или выше.

Программа написана на языке C++.

# 2 Описание логической структуры программы

2.1 Алгоритм программы

1. Считать входные данные из входного файла.
2. Проверить корректность входных данных.
3. Считать последний символ каждой строки, определяющий число символов в группе
4. Записать группы символов в обратном порядке
5. Вывести в файл исправленные строки

## 2.2 Декомпозиция программы

Выделенные подпрограммы (функции) описаны в приложении А.

Иерархия вызовов подпрограмм представлена в приложении Б.

Диаграмма потоков данных представлена в приложении В.

# 

# Приложение А

Описание функций

Функция: **int main(const int argc, char\*\* argv)**

Обеспечить считывание из файла, вывод в консоль ошибок, если они есть, запись ответа в выходной файл, вызов главной функции, решающей задачу.

Алгоритм работы функции (псевдокод):

Выдать ошибку, если входной файл не указан в аргументах командной строки

Выдать ошибку, если входной файл невозможно открыть

Считать массив строк из файла…

Если указано другое расширение файла

{

Распечатать ошибку в консоль ошибок

Завершить работу программы

}

Если количество строк больше 100

{

Распечатать ошибку в консоль ошибок

Завершить работу программы

}

Если количество символов в строке больше 100

{

Распечатать ошибку в консоль ошибок

Завершить работу программы

}

Вызвать функцию декодирования строк

Выдать ошибку, если выходной файл не указан в аргументах командной строки

Выдать ошибку, если выходной файл невозможно открыть

Каждую строку записать в выходной файл

**void decodingAllStrings(std::vector<std::string>& arrayStrings)**

Алгоритм работы функции (псевдокод):

Для каждой строки вызывать функцию, декодирующую строку

Если в строке нет символа, отражающего группы строк

{

Распечатать ошибку в консоль ошибок

Завершить работу программы

}

Функция:

**bool decodingString(std::string& arrayStrings)**

Алгоритм работы функции (псевдокод):

**{**

Определить число символов в группе

Вернуть false, если нет символа, отражающего число символов в группе

Удалить последний символ строки

Определить число групп в строке

Вызвать функцию, разбивающую строку на группы

Вызвать функцию, записывающую строку в обратном порядке

Вернуть true, если декодинг строки прошел успешно

}

Функция:

**int** **numbSymbolsInGroups(std::string& stringForDetermNumbGroups)**

Алгоритм работы функции (псевдокод):

{

Определить размер строки

Считать последний символ, определяющий число символов в одной группе

Определить число символов в группе, исходя из последнего символа

Вернуть это число

}

Функция:

**std::vector<std::string> writeGroupsToArrayStrings(std::string& stringForReverseGroups, int numbSymbolsInGroup, int numbGroups)**

Алгоритм работы функции (псевдокод):

**{**

Объявить вектор строк

Заполнить строки группами символов

Вернуть вектор строк

}

Функция:

**std::string writeReverseGroups(std::vector<std::string>& groupsOfStrings)**

Алгоритм работы функции (псевдокод):

**{**

Определить число групп

Присвоить каждой строке нового вектора строку старого вектора с конца.

Объявляем строку, которая будет иметь конечный вид строки

Конкатанируем перевернутые группы строк в одну строку и присвоим объявленной строке

Вернем эту строку

}

# 

# 

# Приложение Б

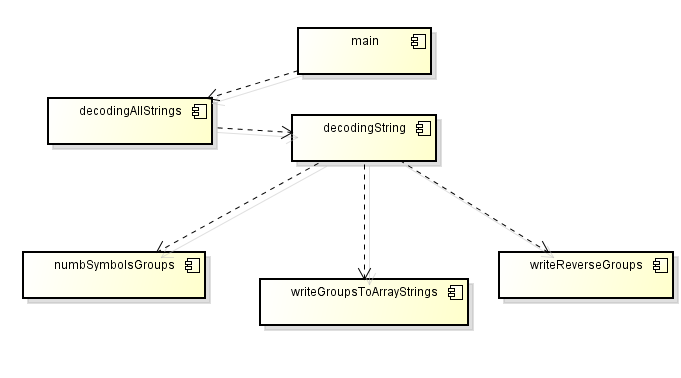
Иерархия вызовов подпрограмм

Рис. 1. Иерархия вызовов функций.

# Приложение В

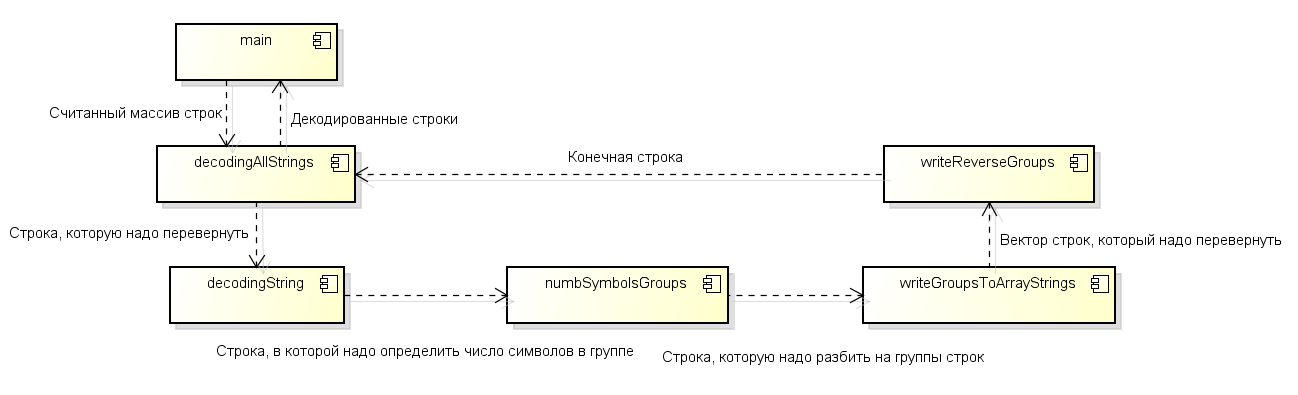
Диаграмма потоков данных

Рис. 2. Диаграмма потоков данных