|  |  |
| --- | --- |
| Программа, для распаковки строки на языке C++ | |
| Внутренная спецификация | |
| Студент | Нагоди М.З. |
| Преподаватель | преп. каф. ПОАС Матюшечкин Д.С. |
| Сдано |  |
| Лабораторная работа №2 |  |

# 1 Общие сведения

Название программы: UnpackString.

Для функционирования программы необходима операционная система

Windows 7 или выше.

Программа написана на языке C++ с использованием Visual Studio 2019.

# 2 Функциональное назначение

Эта программа позволяет распаковать упакованную строку, заменив повторяющиеся символы в строке одним символом и количеством его повторений.

# 3 Описание логической структуры

Функции, используемые в задаче удаленного доступа в Приложении А.

Используемые структуры внешних устройств в приложении Б.

Дерево функций вызова представлено в Приложении B.

Схема потоков данных представлена в Приложении Гп.

Метод решения задачи:

1. Прочитайте входные данные.

2. Проверить правильность введенных данных.

3. Распаковать строку.

4. Разбить строку на массив строк по 40 символов в каждой.

5. Запишите вывод.

# 4 Используемые технические средства

Описание требуемых технических средств содержится в Техническом задании в пункте 3.3.

# 5 Вызов и загрузка

Программа запускается из командной строки. Программа принимает путь к файлу в качестве первого аргумента командной строки.

Пример запуска из командной строки:

UnpackString.exe <<имя\_файла>>.txt

# 6 Входные и выходные данные

Формат входных и выходных данных описан в пункте 3.4 технического задания.

# Приложение А

Описание функций

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

Обеспечивает чтение из файла, вывод ошибок в консоль, если они есть, запись ответа в выходной файл, вызов основной функции, решающей проблему.

Алгоритм работы функции:

1. Проверка наличия входного файла в аргументах командной строки. Если он не существует, выдайте ошибку.
2. Откройте входной файл. Если файл не открывается, выдать ошибку.
3. Считать строку из файла.
4. Запускаем основную функцию программы. Если есть ошибка, напечатайте ее в консоли ошибок и закройте программу.
5. Разбить выходную строку на массив строк по описанию из 40 символов.
6. Записать выходного файла массива

Функция: std::string UnpackString(std::string\_view str)

Распаковывает строку.

Алгоритм работы функции:

1. Проверка размера строки. Если он не соответствует допустимому диапазону, выдайте ошибку.

2. Проверка содержимого строк на корректность. Если есть ошибка, выдайте.

3. Разделите статью на части.

4. Преобразовать единицы в неупакованную строку.

5. Вернуть распакованную строку.

Функция: std::vector<Unit> Split\_String\_Units (std::string\_view str)

Разделяет строку на юниты.

Алгоритм работы функции:

1. Завести массив юнитов.

2. Для каждого символа строки:

2.1. Если текущий символ – буква:

2.1.1. Добавить юнит с текущим символом и повторением = 1.

2.1.2. Продолжить цикл.

2.2. Начало числа повторений – текущий символ.

2.3. Найти первый нецифровой символ – это запоследний символ числа повторений, а также повторяемая буква.

2.4. Конвертировать строку с числом повторением в число.

2.5. Если число повторений не входит в разрешенный диапазон – выдать ошибку.

2.6. Добавить в массив юнит с вычисленным числом повторений и повторяемой буквой.

2.7. Считать текущим символом следующий символ после повторяемой буквы.

3. Вернуть получившийся массив юнитов.

Функция: Code checkStringIsCorrect (std::string\_view str)

Проверяет содержимое строки на корректность.

Алгоритм работы функции:

1. Проверить каждый символ строки. Если найден символ, который не является заглавной латинской буквой или цифрой – вернуть ошибку некорректного символа.

2. Если последний символ является цифрой – вернуть ошибку последней цифры.

3. Ошибок нет – вернуть код без ошибки.

Функция: std::string Unpack\_String\_InUnits (const std::vector<Unit>& units)

Переводит юниты в распакованную строку.

Алгоритм работы функции:

1. Создаём результирующую строку.

2. Для каждого юнита:

2.1. Создаём строку, состоящую из символа из юнита с количеством повторений из юнита.

2.2. Добавляем созданную строку в конец результирующей строки.

3. Вернуть результирующую строку.

Функция: std::vector<std::string> Split\_String\_Limit (const std::string\_view str, int limit)

Разделяет строку на массив строк с длинной равной лимиту.

Алгоритм работы функции:

1. Если лимит меньше либо равен 0, вернуть пустой массив.

2. Создать результирующий массив строк.

3. Для каждой группы символов, длина которой равна лимиту:

3.1. Создать строку из текущей группы символов.

3.2. Добавить строку в результирующий массив.

4. Если в конце строки есть группа символов, длиной меньше лимита, добавить её в результирующий массив.

5. Вернуть результирующий массив.

# Приложение Б

Описание структур данных

Структура Unit содержит следующие поля:

int RepeatNumber – количество повторений символа

char Char – повторяющаяся буква.

Перечисление Code содержит следующие элементы:

NoError – нет ошибки.

IllegalCharacter – ошибка: неразрешенный символ в строке

DigitAtTheEnd – ошибка: в строке последний символ – цифра.

StringSizeOutOfRange – ошибка: размер строки вне разрешенного диапазона.

RepeatNumberIsOutOfRange – ошибка: число повторений буквы вне разрешенного диапазона.

# Приложение В

Дерево вызова функций

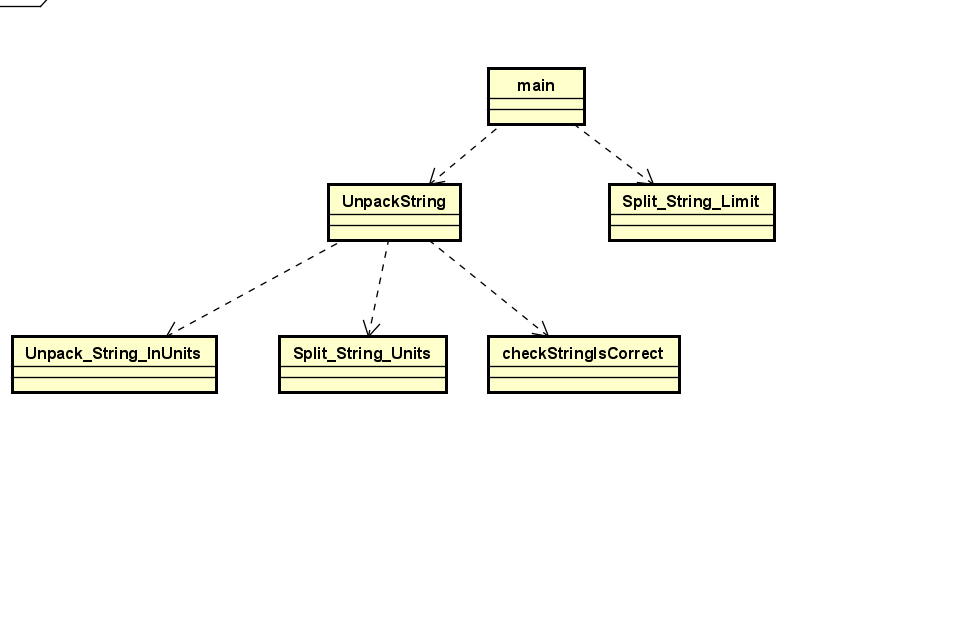


Рис. 1. Дерево вызова

# Приложение Г

Диаграмма потоков данных

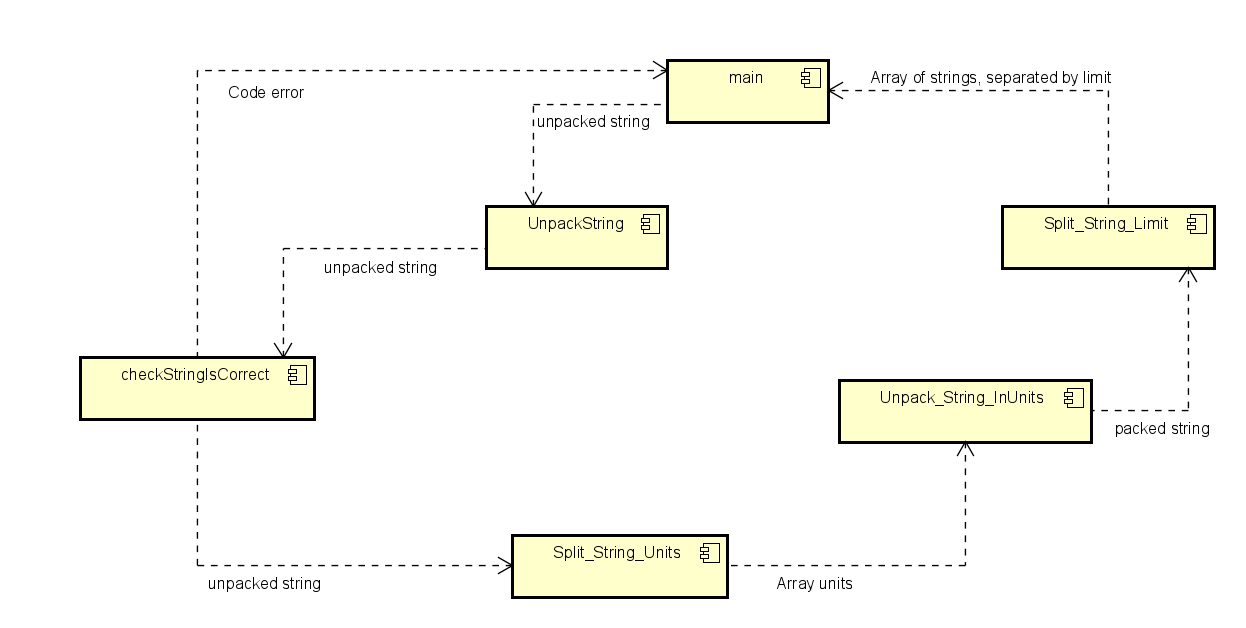


Рис. 2. Диаграмма потоков данных