|  |  |
| --- | --- |
| Программа, для распаковки строки на языке C++ | |
| Модульные тесты | |
| Студент | Нагоди М.З. |
| Преподаватель | преп. каф. ПОАС Матюшечкин Д.С. |
| Сдано |  |
| Лабораторная работа №3 |  |

Содержание

[1 Объект испытаний 3](#_Toc75520332)

[2 Цель испытаний 3](#_Toc75520333)

[3 Требования к программе 3](#_Toc75520334)

[4 Требования к программной документации 4](#_Toc75520335)

[5 Средства и порядок испытаний 4](#_Toc75520336)

[6 Методы испытаний 4](#_Toc75520337)

[Приложение 1. Unit-тесты для функции UnpackString 5](#_Toc75520338)

[Приложение 2. Unit-тесты для функции IsStringCorrect 7](#_Toc75520339)

[Приложение 3. Unit-тесты для функции SplitStringToLimit 8](#_Toc75520340)

[Приложение 4. Unit-тесты для функции SplitStringToUnits 9](#_Toc75520341)

# 1 Объект испытаний

Программа будет именоваться: программа для распаковки строки (далее – программа). Латинское название программы: UnpackString.

# 2 Цель испытаний

Проверить соответствие программы требованиям к функциональным характеристикам.

# 3 Требования к программе

Программа должна распаковать строку, в которой повторяющиеся подряд символы были заменены одним символом и количеством повторений этого символа.

В строке могут содержаться следующие символы:

* буквы от «A» до «Z»;
* цифры от «0» до «9».

Входная строка может содержать два вида конструкций nA или A, где n – это целое число от 2 до 99, а A – повторяющийся символ. Во второй конструкции символ указывается без числа повторений – следовательно, он повторяется один раз.

Требования к надёжности программы, к составу и параметрам технических средств, к информационной и программной совместимости описаны в п. 3.2-3.4 документа «Техническое задание».

# 4 Требования к программной документации

В бумажной форме должно быть представлено техническое задание, технический проект в виде описания программы, программа и методика испытаний, и руководство программиста.

В электронной форме должны быть представлены копии всех документов бумажной формы, рабочая документация и текст программы. Вся документация должна быть представлена в соответствии с ГОСТ 19.

# 5 Средства и порядок испытаний

Для запуска тестов необходимы следующие технические средства: процессор Intel Core i3 и 4 Гб оперативной памяти. Программные средства: ОС Windows 10, Microsoft Visual Studio 2019.

# 6 Методы испытаний

Юнит-тесты для функций UnpackString, IsStringCorrect, SplitStringToLimit, SplitStringToUnits описаны в приложениях 1-4.

# Приложение 1.

Unit-тесты для функции UnpackString

Имя функции: UnpackString

Функция, решающая главную задачу. Распаковывает строку.

Функция вызывает:

1) функцию, проверяющую корректность строки checkStringIsCorrect;

2) функцию, разделяющую строку на юниты Split\_String\_Units;

3) функцию, переводящую юниты в распакованную строку Unpack\_String\_InUnits.

Заголовок:

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

Таблица 1. Тесты для функции UnpackString

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Входные данные | Выходные данные |
| 1 | Одиночный символ | A | A |
| 2 | Несколько одиночных символов | ABC | ABC |
| 3 | Один повторяющийся символ | 10B | BBBBBBBBBB |
| 4 | Несколько повторяющихся символов | 10A5B2C | AAAAAAAAAA  BBBBBCC |
| 5 | Смешанная строка | 10AS5BZY2CX | AAAAAAAAAA  SBBBBBZYCCX |
| 6 | Пустая строка |  | Error: String must not exceed size in allowed range of [1; 80] |
| 7 | Слишком длинная строка | AAAAAAAAAAAAAAAAAAAAA  AAAAAAAAAAAAAAAAAAAAA  AAAAAAAAAAAAAAAAAAAAA  AAAAAAAAAAAAAAAAAA | Error: String must not exceed size in allowed range of [1; 80] |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Входные данные | Выходные данные |
| 8 | В строке есть неразрешенные символы | AbC | Error: String contains illegal character! Allowed only upper case Latin letters and digits! |
| 9 | Строка на конце содержит число | ABC10 | Error: String must not contain digit at the end! |
| 10 | Число повторения символа слишком маленькое | 1A | Error: Character repeat is out of range! Its value must be in [2; 99]! |
| 11 | Число повторения символа слишком большое | 100A | Error: Character repeat is out of range! Its value must be in [2; 99]! |

# Приложение 2.

Unit-тесты для функции checkStringIsCorrect

Имя функции: checkStringIsCorrect.

Проверяет корректность данных, предоставленных пользователем.

Заголовок:

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

Таблица 2. Тесты для функции checkStringIsCorrect

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Входные данные | Выходные данные |
| 1 | Нет ошибок | A10B | NoError |
| 2 | Последний символ - цифра | A10B5 | DigitAtTheEnd |
| 3 | Содержит строчные буквы | A10B5ab | IllegalCharacter |
| 4 | Содержит небуквенные символы | A10B5;. | IllegalCharacter |

# Приложение 3.

Unit-тесты для функции Split\_String\_Limit

Имя функции: Split\_String\_Limit.

Разделяет строку на массив строк с указанным лимитом по длине.

Заголовок:

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

Таблица 3. Тесты для функции SplitStringToLimit

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Входные данные | Выходные данные |
| 1 | Строка меньше лимита | UnpackString("39B")  40 | string(39, 'B') |
| 2 | Строка длиной равна лимиту | UnpackString("40B")  40 | string(40, 'B') |
| 3 | Строка кратна лимиту | UnpackString("40B40D40E")  40 | {  std::string(40, 'B'), std::string(40, 'D'), std::string(40, 'E')  } |
| 4 | Строка некратна лимиту | UnpackString("40A40B40C23Z")  40 | {  std::string(40, 'A'), std::string(40, 'B'), std::string(40, 'C'), std::string(23, 'Z')  } |
| 5 | Некорретный лимит | UnpackString("39B")  -42 | Empty Array |
| 6 | Пустая строка | «»  40 | Empty Array |

# Приложение 4.

Unit-тесты для функции Split\_String\_Units

Имя функции: Split\_String\_Units.

Разделяет строку на юниты.

Заголовок:

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

Таблица 4. Тесты для функции Split\_String\_Units

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Входные данные | Выходные данные |
| 1 | Одиночный символ | "B" | {{1, 'B'}} |
| 2 | Один повторяющийся символ | "10D" | {{10, 'D'}} |
| 3 | Несколько одиночных символов | "ABCD" | {{1, 'A'}, {1, 'B'}, {1, 'C'}, {1, 'D'}} |
| 4 | Несколько повторяющихся символов | "10A20B30C40D" | {{10, 'A'}, {20, 'B'}, {30, 'C'}, {40, 'D'}} |
| 5 | Смешанные символы | "A20BC40D" | {{1, 'A'}, {20, 'B'}, {1, 'C'}, {40, 'D'}} |

# 

# Приложение 5.

Unit-тесты для функции Unpack\_String\_InUnits

Имя функции: Unpack\_String\_InUnits.

Распаковывает юниты в конечную строку.

Заголовок:

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

Таблица 5. Тесты для функции UnitsToUnpackedString

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Входные данные | Выходные данные |
| 1 | Пустой массив юнитов | {} | “” |
| 2 | Только одиночные юниты | {{1, 'D'}, {1, 'E'}, {1, 'G'}} | "DEG" |
| 3 | Только повторяющиеся юниты | {{2, 'D'}, {3, 'B'}, {4, 'C'}} | "DDBBBCCCC" |
| 4 | Смешанные юниты | {{2, 'D'}, {1, 'B'}, {4, 'E'}} | "DDBEEEE" |