

Программа для удаления из программы на языке Си многострочных комментариев	
Внутренняя спецификация	
Студент	Елисеева В. Д.
Преподаватель	Матюшечкин Д. С.
Сдано	
Лабораторная работа №2	

## 1 Общие сведения

Наименование программы – «Программа для удаления из программы на языке Си многострочных комментариев».

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. Удалить из синтаксически верного текста программы на языке Си многострочные комментарии
4. Записать выходные данные в выходной файл.

### 2.2 Декомпозиция программы

Выделенные подпрограммы (функции) описаны в приложении А.

Основные типы и структуры данных программы описаны в приложении Б.

Иерархия вызовов подпрограмм представлена в приложении В.

Диаграмма потоков данных представлена в приложении Г

## 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 выходной файл не указан в аргументах командной строки

Выдать ошибку, если входной файл невозможно открыть

Считать содержимое файла

Если есть ошибка

{

    Вывести информацию об ошибке

    Завершить работу программы

}

Удалить из текста программы на языке Си многострочные комментарии

Записать измененный текст программы на языке Си в выходной файл

Функция: `void delComments(char*text[MAXSIZE][STR_MAXSIZE])`

Удалить многострочные комментарии в тексте, если они есть

Алгоритм работы функции (псевдокод):

..Считать индекс строки начала комментария равным 0

..Считать индекс строки конца комментария равным 0

..Считать позицию начала комментария в строке равной -1

..Считать позицию конца комментария в строке равной -1

..Считать счетчик обращений к функции поиска начала равным 0

..Считать счетчик обращений к функции поиска конца равным 0

Пока комментарии находятся

{

Найти начало многострочного комментария

Найти конец многострочного комментария

Если найдены позиции начала и конца комментария

{

Если начало и конец не на одной строке

{

Если начало раньше конца

{

Удалить начало многострочного комментария

Заменить строку 0 , если образовалась пустая строка  
вместо начала

Удалить тело многострочного комментария, заменяя

Пустые строки 0

Удалить конец многострочного комментария

Заменить строку 0 , если образовалась пустая строка  
вместо начала

} Иначе Выбросить исключение

}Иначе Позиция начала становится позицией конца

}Иначе

Если многострочный комментарий не найден и к функциям  
поиска обращались единожды

{

Выбросить исключение

}

Иначе Если не найдено начало многострочного комментария или  
конец

{

```
        Выбросить исключение
    }
    Блок обработки исключений
```

Функция void findBeginComment(char\*\* beginComment, char\* text[MAXSIZE][STR\_MAXSIZE], int numberString, int\*indexRowBegin)

Найти начало комментария в тексте

Алгоритм работы функции (псевдокод):

..Считать позицию начала комментария в строке равной -1

..Считать переменную состояния равной 0

Пока строки в тексте не закончились

```
{
```

```
    Пока не закончились символы в строке
```

```
    {
```

```
        Изменить статус на «в строковой константе», если текущий
        символ – двойные кавычки и статус равен 0
```

```
        Изменить статус на «в тексте», если текущий символ –
        двойная кавычка и статус «в строковой константе»
```

```
        Изменить статус на «в однострочном комментарии», если
        текущий символ не предпоследний в строке и если текущий символ и
        следующий за ним «/» и статус «в тексте»
```

```
        Если текущий элемент не предпоследний
```

```
        {
```

```
            Если текущий символ «/», а следующий за ним «*» и
            статус «в тексте» и многострочный комментарий еще не найден
```

```
            {
```

```
                Если текущая строка не равна строке с которой
                начинается поиск
```

```
                {
```



Изменить статус на «в однострочном комментарии», если текущий символ не предпоследний в строке и если текущий символ и следующий за ним «/» и статус «в тексте»

Если текущий элемент не предпоследний

{

Если текущий символ «\*», а следующий за ним «/» и статус «в тексте» и многострочный комментарий еще не найден

{

Если текущая строка не равна строке с которой начинается поиск

{

Запомнить индекс текущий строки

Запомнить номер текущей позиции

} Иначе Если текущая позиция не равна позиции с которой начинается поиск

{

Запомнить индекс текущий строки

Запомнить номер текущей позиции

}

}

}

}

Меняем статус на «в тексте», если статус «в однострочном комментарии»

}

Счетчик обращений к функции поиска увеличивается на 1

Функция void saveResult (char\*text[MAXSIZE][STR\_MAXSIZE], int numberString)

Сохранить измененный текст программы на языке Си в выходной файл

Алгоритм работы функции(псевдокод):

Открыть файл

Пока не закончились строки в тексте

{

    Если длина текущей строки равна 1

    {

        Записать текущую строку в выходной файл, если в текущей строке находится не 0

    }Иначе Записать текущую строку в выходной файл

}

Закрыть файл



Основные типы и структуры данных программы

`char * text[MAXSIZE][STR_MAXSIZE]` – текст программы на языке Си, в котором нужно удалить многострочные комментарии

`int indexRowBegin` – индекс строки, в которой был найден символ начала многострочного комментария

`int indexRowEnd` – индекс строки, в которой был найден символ конца многострочного комментария

`int countFindBegin` -счетчик обращений к функции поиска начала многострочного комментария

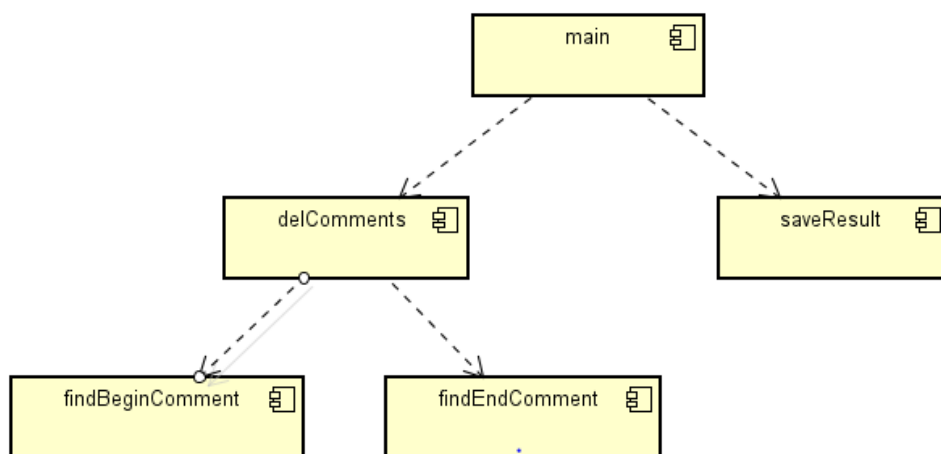
`int countFindEnd` – счетчик обращений к функции поиска конца многострочного комментария

`int posBegin` – позиция начала многострочного комментария в строке

`int posEnd` – позиция конца многострочного комментария в строке

`int status = 0;` //переменная, отвечающая за текущее положение(0-в тексте, 1-в строковой константе, 2- в однострочном комментарии)

Иерархия вызовов подпрограмм



# Диаграмма потоков данных

