

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное образовательное
учреждение высшего образования
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
РУТ (МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**ОТЧЕТ ПО ДИСЦИПЛИНЕ
«АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»
ЛАБОРАТОРНАЯ РАБОТА №9**

Направление: 10.03.01 Информационная безопасность

Профиль: Безопасность компьютерных систем

Выполнил:
студент группы УИБ-115
Шевченко Димитрий

Проверил:

(должность, ФИО)

(должность, ФИО)

Москва 2021 г.

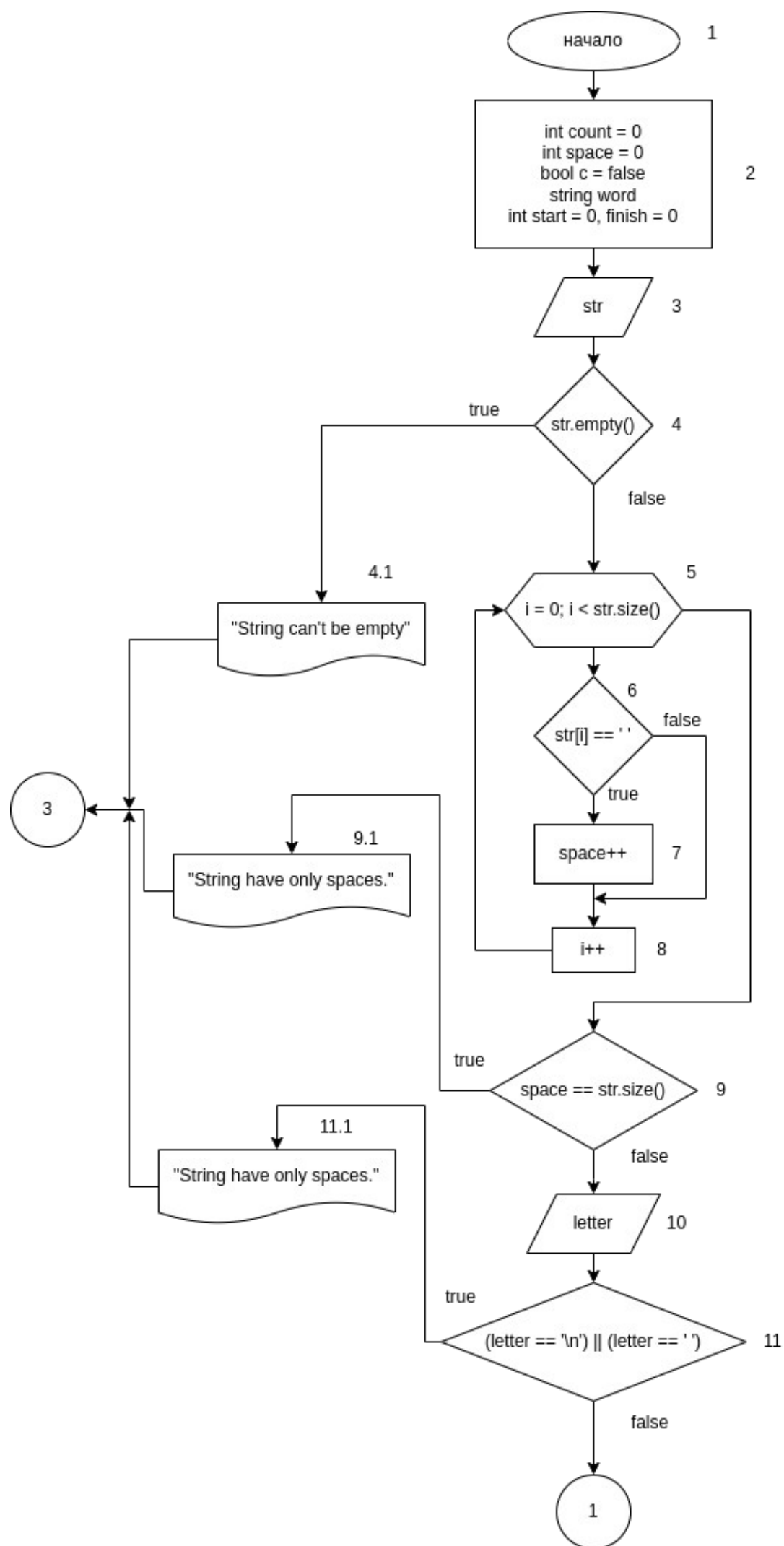
Задание №9

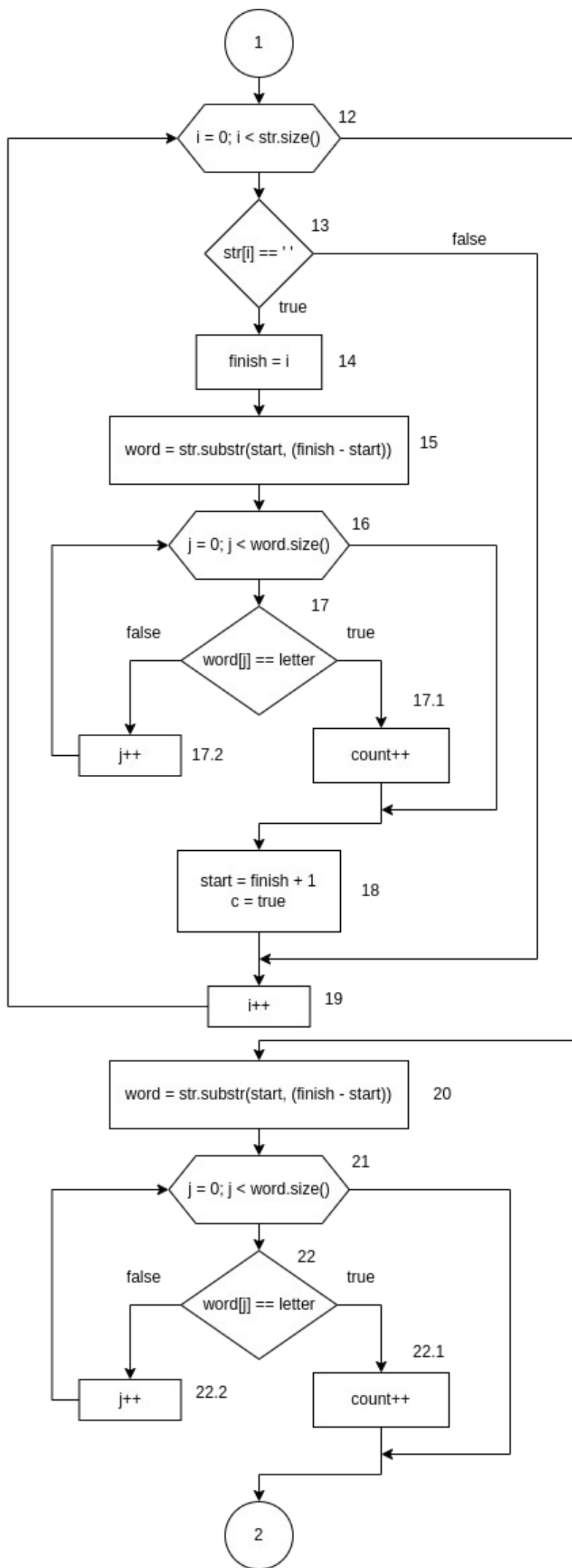
Задание: Задан текст. Определить количество слов, в которых встречается буква вводимая с клавиатуры.

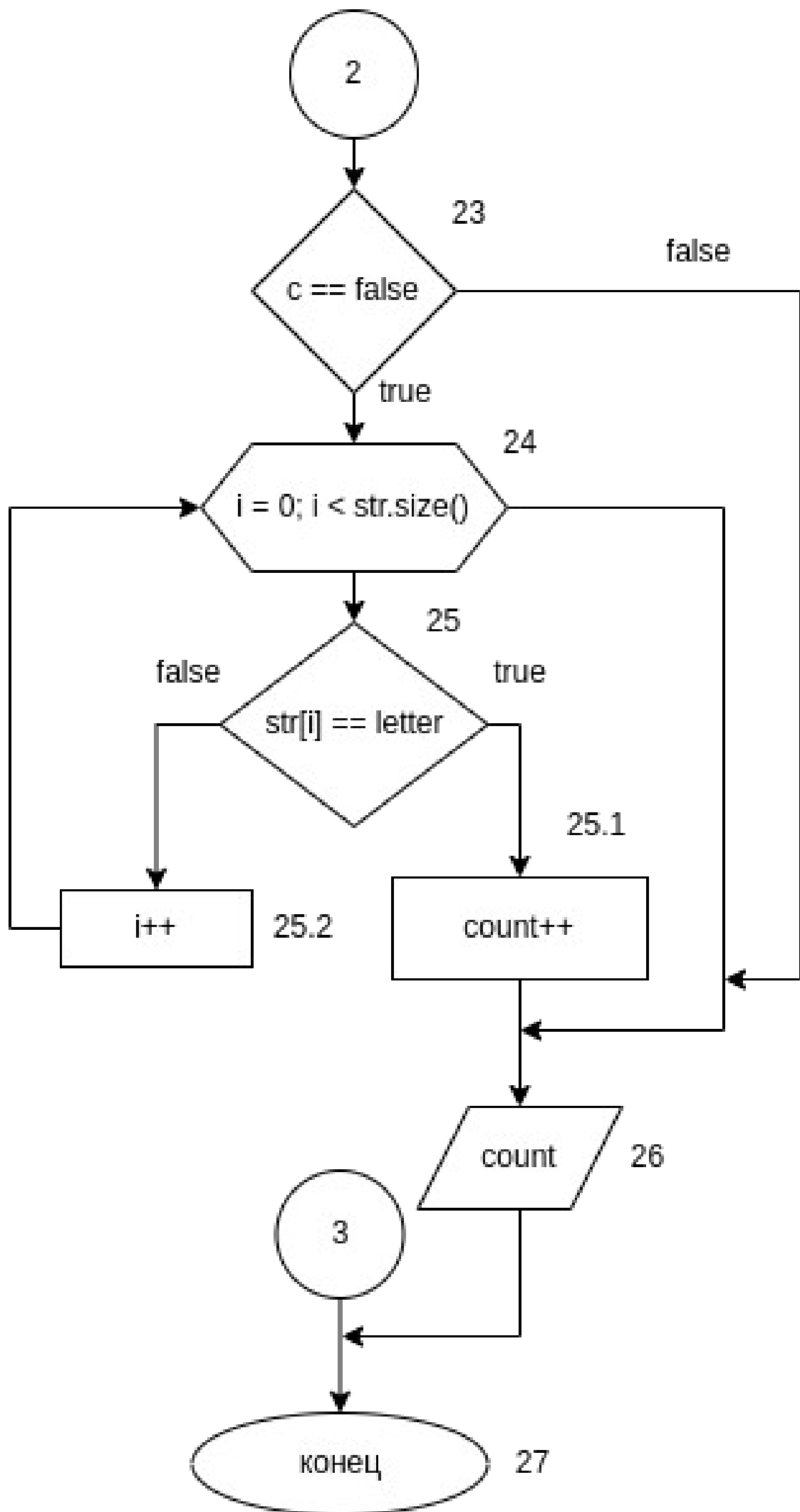
1 Таблица имён:

Исходные данные		
str	String (строка)	Вводимая строка
letter	Char (символ)	Искомая буква
Рабочие переменные		
space	Int (целочисленное)	Счётчик для проверки на пробелы в строке
c	Bool (логическое)	Переменная проверки на разделитель
word	String (строка)	Строка для слова
start	Int (целочисленное)	Значение начала слова
finish	Int (целочисленное)	Значение конца слова
i	Int (целочисленное)	Счётчик цикла
j	Int (целочисленное)	Счётчик цикла
Результат		
count	Int (целочисленное)	Счётчик для количества подсчитанных слов

2 Блок-схема:







3 Отладочные примеры:

Вариант I

1) Начало

2) `int count = 0; int space = 0; bool c = false; string word; int start = 0, finish = 0;`

3) Ввод str

Пустая строка

4) Так как str пуста, то

4.1) Вывод "String can't be empty"

27) Конец

Вариант II

1) Начало

2) `int count = 0; int space = 0; bool c = false; string word; int start = 0, finish = 0;`

3) Ввод str

Ряд из 10 пробелов

4) Так как str не пуста, то переходим к шагу 5

5) Цикл от 0 до 10 с переменной i

6) Если `str[i]` равна пробелу, то

7) Увеличиваем space на 1

8) Увеличиваем i на 1

9) Так как space (10) равен 10, значит, что все символы - пробелы. Тогда

9.1) Вывод "String have only spaces."

27) Конец

Вариант III

- 1) Начало
- 2) `int count = 0; int space = 0; bool c = false; string word; int start = 0, finish = 0;`
- 3) Ввод `str`
“fail or not fail.”
- 4) Так как `str` не пуста, то переходим к следующему шагу
- 5) Цикл от 0 до 17 с переменной `i`
 - 6) Если `str[i]` равна пробелу, то
 - 7) Увеличиваем `space` на 1
 - 8) Увеличиваем `i` на 1
- 9) Так как `space` (3) не равно 17, значит, что не все символы – пробелы.
- 10) Ввод `letter`
Пустой ввод
- 11) Так как `letter` пуста, то
 - 11.1) Вывод “Letter can't be empty”
- 27) Конец

Вариант IV

- 1) Начало
- 2) `int count = 0; int space = 0; bool c = false; string word; int start = 0, finish = 0;`
- 3) Ввод `str`
“God, bless "invalid input detected".”
- 4) Так как `str` не пуста, то переходим к шагу 5
- 5) Цикл от 0 до 36 с переменной `i`
 - 6) Если `str[i]` равна пробелу, то
 - 7) Увеличиваем `space` на 1
 - 8) Увеличиваем `i` на 1
- 9) Так как `space` (4) не равно 36, значит, что не все символы – пробелы.
- 10) Ввод `letter`
`d`
- 11) Так как `letter` не пуста, то переходим к следующему шагу.

- 12) Цикл от нуля до 36 с переменной i
- 13) Если str[i] является пробелом, то
- 14) finish = i
- 15) word = str.substr(start, (finish – start))
- 16) Цикл от 0 до word.size() с переменной j
- 17) Если word[j] равен letter, то
- 17.1) Увеличиваем count на 1 и останавливаем цикл
- 17.2) Увеличиваем j на 1
- 18) start = finish + 1; c = true
- 19) Увеличиваем i на 1
- 20) word = str.substr(start, (finish – start))
- 21) Цикл от 0 до word.size() с переменной j
- 22) Если word[j] равен letter, то
- 22.1) Увеличиваем count на 1 и останавливаем цикл
- 22.2) Увеличиваем j на 1
- 23) Так как c равна true, то переходим к шагу 26
- 26) Вывод count
- 3
- 27) Конец

Вариант V

- 1) Начало
- 2) int count = 0; int space = 0; bool c = false; string word; int start = 0, finish = 0;
- 3) Ввод str
- “Londinium”
- 4) Так как str не пуста, то переходим к шагу 5
- 5) Цикл от 0 до 9 с переменной i
- 6) Если str[i] равна пробелу, то
- 7) Увеличиваем space на 1
- 8) Увеличиваем i на 1
- 9) Так как space (0) не равно 9, значит, что не все символы – пробелы.
- 10) Ввод letter
- i
- 11) Так как letter не пуста, то переходим к следующему шагу.

- 12) Цикл от нуля до 9 с переменной i
- 13) Если str[i] является пробелом, то
- 14) finish = i
- 15) word = str.substr(start, (finish – start))
- 16) Цикл от 0 до word.size() с переменной j
- 17) Если word[j] равен letter, то
- 17.1) Увеличиваем count на 1 и останавливаем цикл
- 17.2) Увеличиваем j на 1
- 18) start = finish + 1; c = true
- 19) Увеличиваем i на 1
- 20) word = str.substr(start, (finish – start))
- 21) Цикл от 0 до word.size() с переменной j
- 22) Если word[j] равен letter, то
- 22.1) Увеличиваем count на 1 и останавливаем цикл
- 22.2) Увеличиваем j на 1
- 23) Так как c равна false, то
- 24) Цикл от нуля до 9 с переменной i
- 25) Если word[j] равен letter, то
- 25.1) Увеличиваем count на 1 и останавливаем цикл
- 25.2) Увеличиваем j на 1
- 26) Вывод count
- 1
- 27) Конец

4 Код программы:

```
#include <iostream> //подключение библиотеки функции ввода-вывода
#include <string> // подключение библиотеки для работы со строками
using namespace std; //подключение пространства имён std

int main(){ //точка входа в программу

    string str; // вводимая строка
    char letter; // искомая буква
    int count = 0; // счётчик для количества подсчитанных слов
    int space = 0; // счётчик для проверки на пробелы в строке
    bool c = false; // переменная проверки на разделитель
    string word; // строка для слова
    int start = 0, finish = 0; // значения начала и конца слова


    cout << "Input string: " << endl;
    getline(cin, str); // ввод строки


    // проверка на пустоту строки
    if (str.empty()){ // если строка пуста, то
        cout << "String can't be empty" << endl; // вывод сообщения об
        ошибке
        exit(0); // завершение программы
    }
    else { // иначе проверка на пробелы
        for (int i = 0; i < str.size(); i++){ // цикл на всю строку
            if (str[i] == ' '){ // если символ является пробелом, то
                space++; // увеличиваем счётчик
            }
        }
        if (space == str.size()) { // если количество пробелов равно размеру
        строки, то
```

```

        cout << "String have only spaces." << endl; // вывод сообщения
        об ошибке
        exit(0); // завершение программы
    }
}

cout << "Input letter: " << endl;
letter = getchar(); // ввод буквы

// проверка на пустоту вводимого символа
if ((letter == '\n') || (letter == ' ')){ // если символ является переходом на
новую строку или пробелом, то
    cout << "Letter can't be empty" << endl; // вывод сообщения об
    ошибке
    exit(0); // завершение программы
}

for (int i = 0; i < str.size(); i++){ // цикл на полную строку
    if (str[i] == ' '){ // если текущий символ является пробелом, то
        finish = i; // объявляем концом слова значение i
        word = str.substr(start, (finish - start)); // присваиваем символы
        от start до finish переменной word
        for (int j = 0; j < word.size(); j++){ // цикл на слово
            if (word[j] == letter){ // если в слове есть искомая
                буква, то
                    count++; // увеличиваем счётчик
                    break; // останавливаем цикл по слову
            }
        }
    }
}

```

```

        start = finish + 1; // изменяем значение start на finish + 1 для
        пропуска пробела
        c = true; // меняем значение c на true, так как в строке есть
        пробелы
    }
}

// проверка последнего слова, которое не вошло в основной цикл
word = str.substr(start, (finish - start));
for (int j = 0; j < word.size(); j++){
    if (word[j] == letter){
        count++;
        break;
    }
}

if (c == false){ // если в строке нет пробелов, то
    for (int i = 0; i < str.size(); i++){ // цикл на строку
        if (str[i] == letter){ // если символ является искомой буквой
            count++; // увеличиваем счётчик
            break; // останавливаем цикл по слову
        }
    }
}

cout << "Count of words: " << count << endl; // вывод счётчика

return 0; //успешное завершение программы
}

```

5 Результат выполнения работы программы:

1)

```
Input string:  
String can't be empty
```

2)

```
Input string:  
String have only spaces.
```

3)

```
Input string:  
fail or not fail.  
Input letter:  
Letter can't be empty
```

4)

```
Input string:  
God, bless "invalid input detected".  
Input letter:  
d  
Count of words: 3
```

5)

```
Input string:
Londinium
Input letter:
i
Count of words: 1
```

6 Вывод:

В ходе выполнения работы были изучены строки, методы работы с ними, алгоритм разбиения строки, а также способ подсчёта слов по заданной букве.

Была проделана работа по созданию UI, алгоритма разбиения строки и проверки символов, а также выводу строки.

На контрольных примерах мы убедились, что программа работает корректно и отвечает заданным в ТЗ требованиям.

Был оформлен комплект документации на программный код.