**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 №2**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-112

Орлов Андрей Витальевич

Проверил:

Старший преподаватель Цыганова Наталия Алексеевна

(должность, ФИО)

Старший преподаватель Никольская Мария Николаевна

(должность, ФИО)

Москва 2021 г.

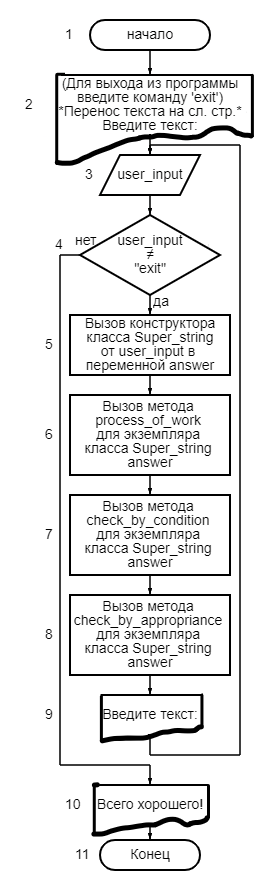
**Задание №2**

Задание: Задан текст, в который входят лишь цифры и буквы. Определить, удовлетворяет ли этот текст следующему условию: текст начинается с цифры, за которой следуют только буквы, и их количество равно числовому значению этой самой цифры.

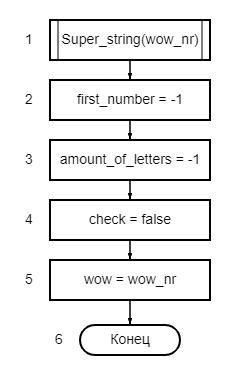
1. Таблица имён:

|  |  |  |
| --- | --- | --- |
| Исходные данные | | |
| user\_input | строковый | Исходная строка, вводимая юзером |
| Рабочие переменные | | |
| first\_number | целочисленный | Переменная для проверки первого символа текста на цифру |
| amount\_of\_letters | целочисленный | Количество букв по всей заданной строчке текста |
| inter\_amount | целочисленный | Длина строки, состоящей только из букв, что следует за первой цифрой |
| wow | строковый | Обрабатываемая строка в классе |
| wow[0], wow[i] | символьный | Элементы обрабатываемой строки |
| wow\_nr | строковый | Строка, задаваемая пользователем |
| с | автоматический | Переменная для пробега строки в цикле |
| answer | Экземпляр класса Super\_string | Объект класса для обработки заданной пользователем строки |
| Результат | | |
| check | логический | Общий флаг, показывающий, удовлетворяет ли строка условию |

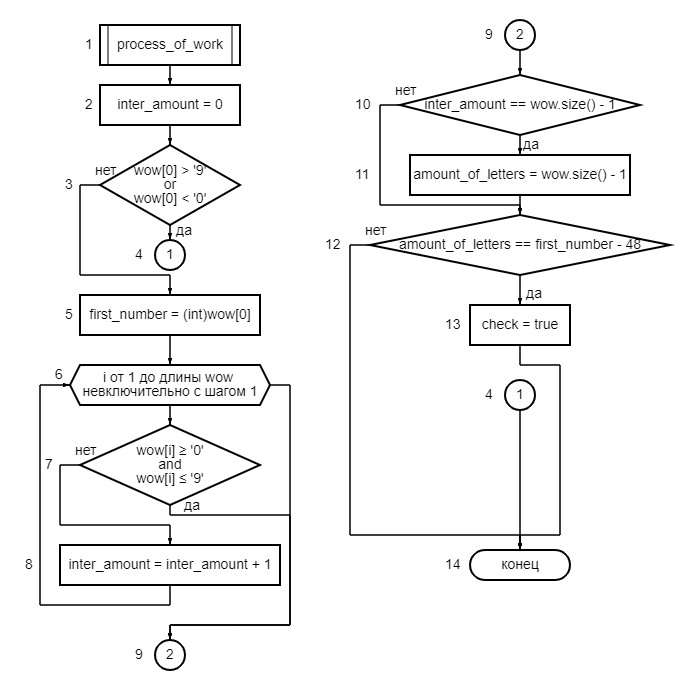
1. Блок-схемы:



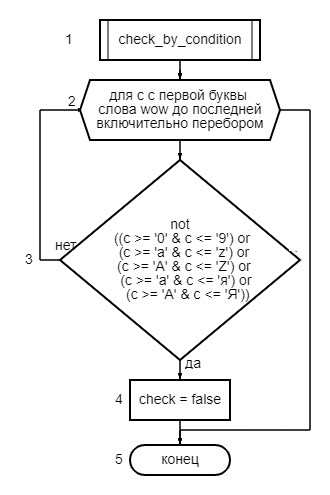
Блок-схема 1: Основной алгоритм



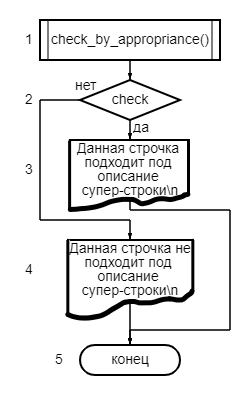
Блок-схема 2: Конструктор класса Super\_string



Блок-схема 3: Метод класса Super\_string: process\_of\_work



Блок-схема 4: Метод класса Super\_string: check\_by\_condition



Блок-схема 5: Метод класса Super\_string: check\_by\_approproance

1. Отладочный пример:

Вариант I

1) Начало  
2) Вывод на экран текста: «(Для выхода из программы введите команду 'exit')\nВведите текст:»  
3) Ввод с клавиатуры пользователем: ‘0’  
4) ‘0’ != “exit”, да, переход к блоку 5  
5) Вызов конструктора класса Super\_string от '0' в переменной answer  
Переход к блок-схеме №2:  
1a) Переприсваивание переменных: wow\_nr = '0'  
2a) Переприсваивание переменных: first\_number = -1   
3a) Переприсваивание переменных: amount\_of\_letters = -1  
3a) Переприсваивание переменных: check = false  
4a) Переприсваивание переменных: wow = '0'  
Конец работы блок-схемы №2, возврат в основную программу  
6) Вызов метода process\_of\_work для экземпляра класса answer  
Переход к блок-схеме №3:   
2б) Переприсваивание переменных: inter\_amount = 0  
3б) Условие: '0' > '9' || '0' < '0', нет, переход к блоку 5б  
5б) Переприсваивание переменных: first\_answer = 48  
6б) Вход в цикл: Переменная i = 1, конец цикла, переход к блоку 9  
9б) Ссылка на блок 10б  
10б) Условие: 0 == 0, да, переход к блоку 11б  
11б) Переприсваивание переменных: amount\_of\_letters = 0  
12б) Условие: 0 == 0, да, переход к блоку 13б   
13б) Переприсваивание переменных: check = true  
Конец работы боок-схемы №3, переход к основной программе  
7) Вызов метода check\_by\_condition для экземпляра класса answer  
Переход к блок-схеме №4:  
2в) Вход в цикл: Переменная c = '0'  
3в) Условие !(('0' >= '0' & '0' <='9')||…) нет, переход к блоку 2в  
2в) Конец цикла, переход к блоку 5  
Конец работы блок-схемы №4, переход к основной части программы  
8) Вызов метода check\_by\_appropriance для экземпляра класса answer  
Переход к блок-схеме №5:  
2г) Условие: true, да, переход к блоку 3г  
3г) Вывод на экран текста “Данная строчка подходит под описание супер-строки\n”  
Конец работы блок-схемы № 5, переход к основной части программы.  
9) Вывод на экран текста: «Введите текст: », переход к блоку 3  
3) Ввод с клавиатуры «exit»  
4) Условие: “exit” != “exit”, нет, переход к блоку 10  
10) Вывод на экран текста: «Всего хорошего!»  
11) Конец

1. Код программы:

#include <string>

#include <iostream>

using namespace std;

class Super\_string {

private:

int first\_number; // Объявления переменных для работы класса

int amount\_of\_letters;

bool check;

string wow;

public:

Super\_string(string wow\_nr) {

first\_number = -1; // Инициализация переменных для работы класса

amount\_of\_letters = -1;

check = false;

wow = wow\_nr;

};

void process\_of\_work() { // метод для проверки: первый символ - число, остальные - не числа

int inter\_amount = 0;

if (wow[0] > '9' || wow[0] < '0')

return;

else

first\_number = (int)wow[0];

for (int i = 1; i < wow.length(); i++) {

if (wow[i] >= '0' && wow[i] <= '9')

break;

inter\_amount++; // Если цикл прервали раньше, значит встретили число

}

if (inter\_amount == wow.size() - 1) // поэтому и сравнение идёт с длиной слова минус первый символ

amount\_of\_letters = wow.size() - 1;

if (amount\_of\_letters == first\_number - 48) // Проверка на равенство кличества букв после числа числу

check = true;

}

// Проверяем, все ли символы являются буквой или цифрой

void check\_by\_condition() {

for (auto c : wow) {

if (!((c >= '0' && c <= '9') ||

(c >= 'a' && c <= 'z') ||

(c >= 'A' && c <= 'Z') ||

(c >= 'а' && c <= 'я') ||

(c >= 'А' && c <= 'Я')))

check = false;

}

}

// Метод для финальной проверки глобальной булевой переменной, показывающая ответ

void check\_by\_appropriance() {

if (check)

cout << "Данная строчка подходит под описание супер-строки.\n";

else

cout << "Данная строчка не подходит под описание супер-строки.\n";

}

};

int main() {

setlocale(LC\_ALL, "rus"); // Подключаем русский язык

system(“chcp 1251 >> null”);

string user\_input;

cout << "(Для выхода из программы введите команду 'exit')\nВведите текст: ";

while (true) { // цикл для многократного тестирования программы

getline(cin, user\_input); // читаем строку

if (user\_input == "exit")

break;

Super\_string answer(user\_input);

answer.process\_of\_work();

answer.check\_by\_condition();

answer.check\_by\_appropriance();

cout << "Введите текст: ";

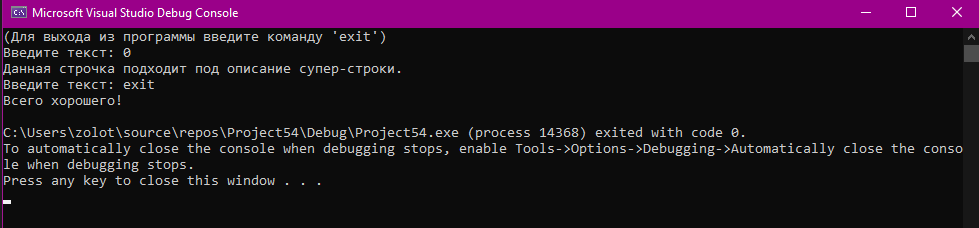
}

cout << "Всего хорошего!\n"; // дружелюбность к пользователю

return 0;

}

1. Результат выполнения работы программы:



Снимок работы программы при отладочном примере №1

1. Вывод:

В ходе выполнения работы были изучены основные способы работы с классами, а именно – создание личного конструктора класса, создание метода класса, разделение полей класса по доступности с помощью ключевых слов private\public.

Было проделано несколько тестирований кода, все выявленные исключения были из программы удалены посредством переписи кода.

На контрольных примерах мы убедились, что код работает стабильно.

Был оформлен комплект документации на программный код.