

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	5
2 МЕТОД РЕШЕНИЯ.....	6
3 ОПИСАНИЕ АЛГОРИТМОВ.....	7
3.1 Алгоритм метода set_n класса obj.....	7
3.2 Алгоритм метода count_digits класса obj.....	7
3.3 Алгоритм метода get_count класса obj.....	8
3.4 Алгоритм метода get_n класса obj.....	8
3.5 Алгоритм функции main.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
5 КОД ПРОГРАММЫ.....	12
5.1 Файл main.cpp.....	12
5.2 Файл MyClass.cpp.....	12
5.3 Файл MyClass.h.....	13
6 ТЕСТИРОВАНИЕ.....	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	15

1 ПОСТАНОВКА ЗАДАЧИ

Создать объект, который обрабатывает переменную целого типа максимальной длины.

У объекта есть закрытое свойство n целого типа максимальной длины.

Объект обладает следующей функциональностью:

- задает значение свойства n ;
- вычисляет количество цифр значения свойства n ;
- возвращает количество цифр значения свойства n .

Написать программу, которая:

1. Создает объект.
2. Вводит значение переменной целого типа.
3. Определяет значение свойства n по значению переменной целого типа.
4. Вычисляет количество цифр свойства n .
5. Выводит значение свойства n .
6. Выводит количество цифр значения свойства n .

1.1 Описание входных данных

Первая строка:

Целое число в десятичном формате.

1.2 Описание выходных данных

Первая строка:

n = «значение свойства n »

Вторая строка:

N = «количество цифр свойства n »

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj класса MyClass;
- функция main для Основная функция;
- Объект стандартного потока ввода/вывода cin/cout;
- Условный оператор;
- Оператор цикла с предусловием.

Класс obj:

- свойства/поля:
 - поле Число целого типа максимальной длины:
 - наименование — n;
 - тип — long long int;
 - модификатор доступа — private;
 - поле Количество цифр в числе:
 - наименование — count;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод set_n — Получение значения n;
 - метод count_digits — Подсчёт количества цифр в числе n;
 - метод get_count — Получение(вывод) количества цифр;
 - метод get_n — Получение(вывод) самого n.

3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

3.1 Алгоритм метода `set_n` класса `obj`

Функционал: Получение значения `n`.

Параметры: `long long int n1`.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `set_n` класса `obj`

№	Предикат	Действия	№ перехода
1		Присвоение переменной <code>n</code> значения переменной <code>n1</code>	Ø

3.2 Алгоритм метода `count_digits` класса `obj`

Функционал: Подсчёт количества цифр в числе `n`.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `count_digits` класса `obj`

№	Предикат	Действия	№ перехода
1		Инициализация переменной целочисленной переменной <code>count = 0</code>	2
2		Объявление и присвоение целочисленной	3

№	Предикат	Действия	№ перехода
		переменной n2 значения перменной n	
3	n == 0	count = 1	∅
			4
4	n != 0	Делим n2 на 10	5
			∅
5		Инкрементируем count	∅

3.3 Алгоритм метода get_count класса obj

Функционал: Получение(вывод) количества цифр.

Параметры: нет.

Возвращаемое значение: Целое.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода get_count класса obj

№	Предикат	Действия	№ перехода
1		Возвращение целочисленной переменной count	∅

3.4 Алгоритм метода get_n класса obj

Функционал: Получение(вывод) самого n.

Параметры: нет.

Возвращаемое значение: Целочисленное максимальной длиной.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода get_n класса obj

№	Предикат	Действия	№ перехода
1		Возвращение переменной n	∅

3.5 Алгоритм функции main

Функционал: Основная функция.

Параметры: Отсутствуют.

Возвращаемое значение: Целое с максимальной длиной, целое.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной n с максимальной длиной	2
2		Ввод n	3
3		Создание объекта obj	4
4		Вызов методов set_n(), count_digits(), get_count(), get_n()	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

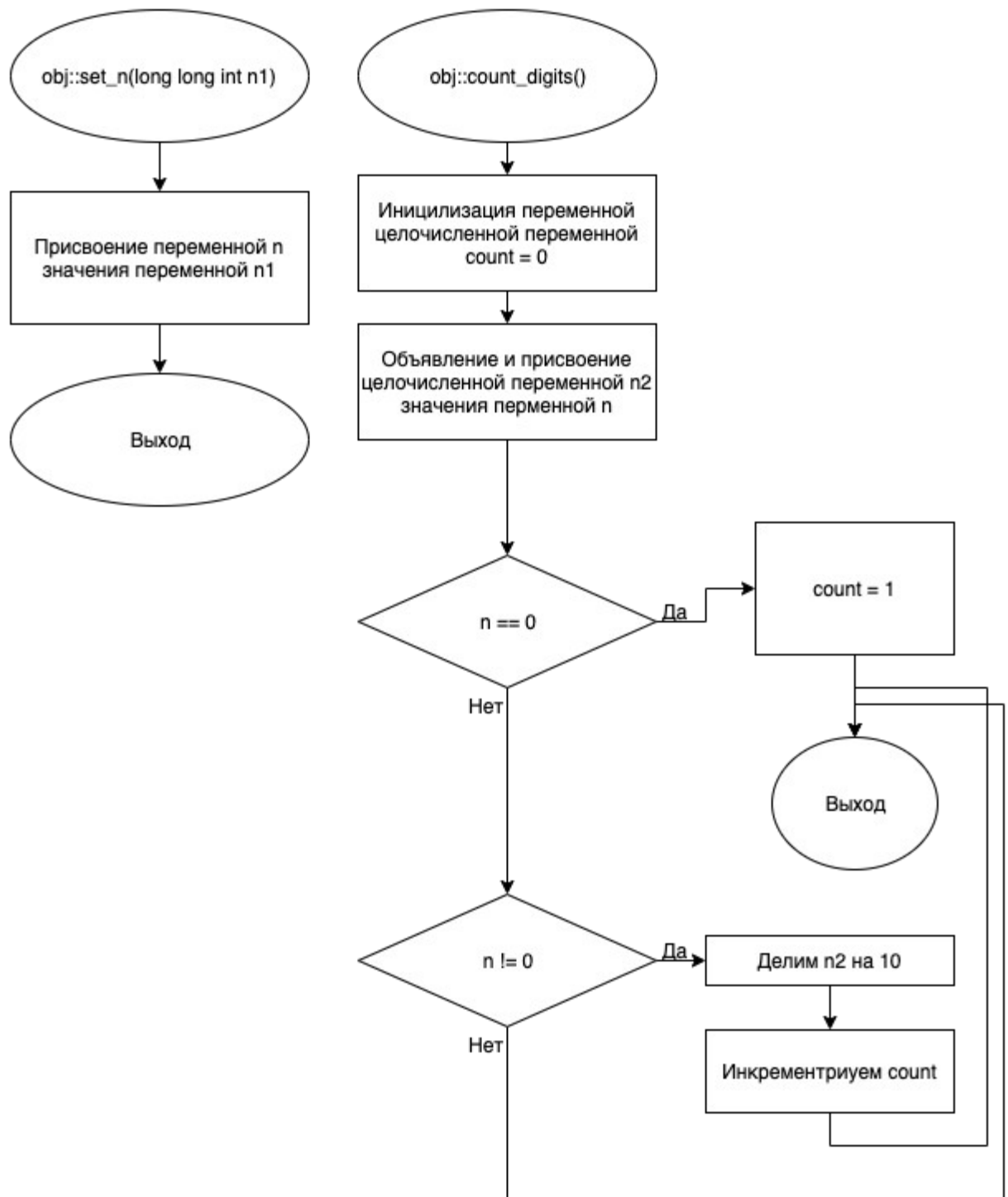


Рисунок 1 – Блок-схема алгоритма

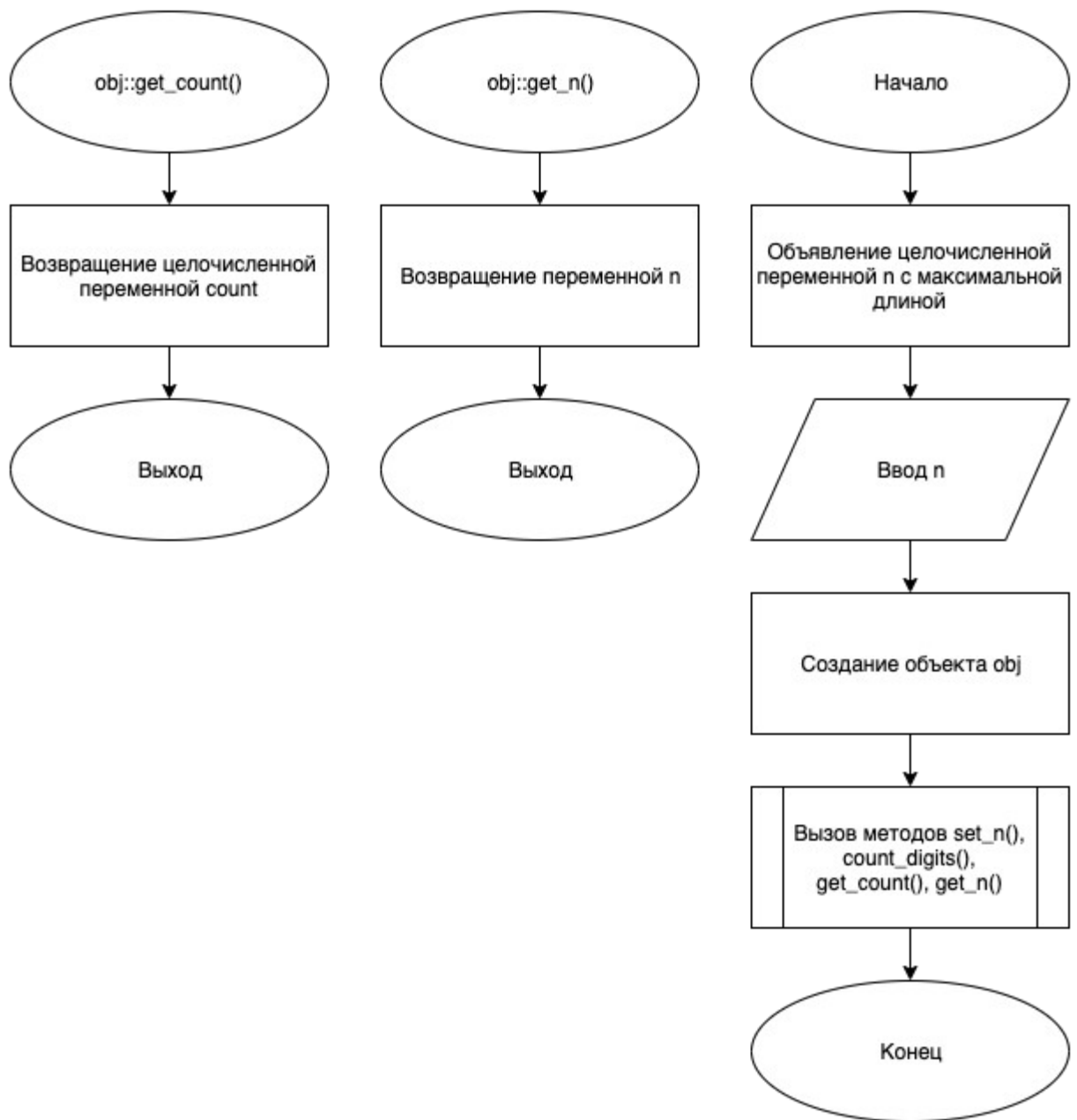


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "MyClass.h"
int main()
{
    long long int n;
    cin >> n;
    MyClass obj;
    obj.set_n(n);
    obj.count_digits();
    cout <<"n = " << obj.get_n() << endl;
    cout <<"N = " << obj.get_count();
    return(0);
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"

void MyClass::set_n(long long int n1)
{
    n = n1;
}
void MyClass::count_digits()
{
    count=0;
    long long int n2=abs(n);
    if(n2 == 0)
    {
        count = 1;
    }
    while(n2 != 0)
```

```

        {
            n2/=10;
            count++;
        }

    }
    int MyClass::get_count()
    {
        return count;
    }

    long long int MyClass::get_n()
    {
        return n;
    }

```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```

#ifndef __MYCLASS__H
#define __MYCLASS__H
#include <iostream>
using namespace std;

class MyClass
{
    long long int n;
    int count;
public:
    void set_n(long long int n1);
    void count_digits();
    int get_count();
    long long int get_n();
};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
-10	n = -10 N = 2	n = -10 N = 2
10	n = 10 N = 2	n = 10 N = 2
1234567890123	n = 1234567890123 N = 13	n = 1234567890123 N = 13

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.
2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avvora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avvora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
4. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).