

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	9
3.1 Алгоритм функции main.....	9
3.2 Алгоритм конструктора класса Object.....	10
3.3 Алгоритм метода add_5_7 класса Object.....	10
3.4 Алгоритм метода add_1_4 класса Object.....	11
3.5 Алгоритм метода call_add_5_7 класса Object.....	11
3.6 Алгоритм метода print класса Object.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	21
5.1 Файл main.cpp.....	21
5.2 Файл Object.cpp.....	22
5.3 Файл Object.h.....	23
6 ТЕСТИРОВАНИЕ.....	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	25

# 1 ПОСТАНОВКА ЗАДАЧИ

Спроектировать объект, в котором есть доступные и скрытые элементы (свойства и методы). Есть одно доступное свойство целого типа и одно скрытое свойство целого типа.

У объекта есть параметризованный конструктор с параметром целого типа. В конструкторе доступному свойству присваивается значение параметра, а скрытому свойству удвоенное значение параметра.

У объекта есть доступные методы со следующим функционалом:

Метод изменения значения доступного и скрытого свойства. Доступному свойству добавляется 1, скрытому свойству добавляется 4;

Метод вызова скрытого метода;

Метод вывода состояния, выводит значение доступного и скрытого свойства.

У объекта есть один скрытый метод со следующим функционалом:

метод доступному свойству добавляет 5, скрытому свойству добавляет 7.

Написать программу, которая состоит из описания класса выше представленного объекта и основной функции, в которой реализован следующий алгоритм:

1. Ввод целочисленного значения переменной `i_data`.
2. Создание объекта, параметризованному конструктору в качестве аргумента передается переменная `i_data`.
3. Вывод исходного состояния объекта.
4. Вызов метода изменения значений свойств объекта.
5. Вывод текущего состояния объекта.
6. Ввод целочисленного значения переменной `i_data`.
7. Непосредственное изменение доступного свойства объекта посредством умножения его значения на `i_data`.

8. Вывод текущего состояния объекта.
9. Вызов метода объекта, который вызывает скрытый метод объекта.
10. Вывод текущего состояния объекта.

## **1.1 Описание входных данных**

Первая строка

«Целочисленное значение»

Вторая строка

«Целочисленное значение»

## **1.2 Описание выходных данных**

Метод вывода состояния, первый вывод делает в первой строке, а далее всегда с новой строки. Шаблон вывода:

Value of the available property «значение доступного свойства»; Value of a hidden property «значение закрытого свойства»

## 2 МЕТОД РЕШЕНИЯ

Используемые инструменты:

объект стандартного потока ввода cin (используется для ввода с клавиатуры);

объект стандартного потока вывода cout (используется для вывода на экран);

объект obj класса Object.

Класс Object:

Свойства (поля):

Открытое поле:

Закрытое поле:

Наименование - closed;

Тип - целочисленный;

Модификатор доступа - закрытый;

Методы:

Конструктор Object:

Функционал - создать объект с открытым полем, равным значению параметра, и закрытым, равным удвоенному значению параметра;

Возвращаемое значение - void (отсутствует);

Модификатор доступа - открытый;

Параметры - целочисленный параметр param;

Метод add\_5\_7:

Функционал - добавить к открытому полю 5, к закрытому - 7;

Возвращаемое значение - void (отсутствует);

Модификатор доступа - закрытый;

Параметры - отсутствуют;

Метод add\_1\_4:

Функционал - добавить к открытому полю 1, к закрытому - 4;

Возвращаемое значение - void (отсутствует);

Модификатор доступа - открытый;

Параметры - отсутствуют;

Метод call\_add\_5\_7:

Функционал - вызвать метод add\_5\_7;

Возвращаемое значение - void (отсутствует);

Модификатор доступа - открытый;

Параметры - отсутствуют;

Метод print:

Функционал - вывести на экран значения открытого и закрытого полей;

Возвращаемое значение - void (отсутствует);

Модификатор доступа - открытый;

Параметры - отсутствуют;

## 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 функции main

Функционал: основной алгоритм программы.

Параметры: .

Возвращаемое значение: целочисленный код завершения работы программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i_data значением 0	2
2		Ввод значения переменной i_data с клавиатуры	3
3		Создание объекта obj класса Object с использованием параметризованного конструктора и переменной i_data в качестве параметра	4
4		Вызов метода print объекта obj	5
5		Вывод "\n" (переход на новую строку)	6
6		Вызов метода add_1_4 объекта obj	7
7		Вызов метода print объекта obj	8
8		Вывод "\n" (переход на новую строку)	9
9		Ввод значения переменной i_data с клавиатуры	10
10		Присваивание полю open объекта obj результата произведения значения поля open объекта obj и значения переменной i_data	11

№	Предикат	Действия	№ перехода
1 1		Вызов метода print объекта obj	12
1 2		Вывод "\n" (переход на новую строку)	13
1 3		Вызов метода call_add_5_7 объекта obj	14
1 4		Вызов метода print объекта obj (вывод текущего состояния объекта на экран)	Ø

### 3.2 Алгоритм конструктора класса Object

Функционал: создать объект с открытым полем, равным значению параметра, и закрытым, равным удвоенному значению параметра.

Параметры: целочисленный параметр param.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Присваивание полю open значения параметра param	2
2		Присваивание полю closed результата произведения значения параметра param и 2	Ø

### 3.3 Алгоритм метода add\_5\_7 класса Object

Функционал: добавить к открытому полю 5, к закрытому - 7.

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 3.



Таблица 3 – Алгоритм метода add\_5\_7 класса Object

№	Предикат	Действия	№ перехода
1		Присваивание полю open суммы значения поля open и 5	2
2		Присваивание полю closed суммы значения поля closed и 7	∅

### 3.4 Алгоритм метода add\_1\_4 класса Object

Функционал: добавить к открытому полю 1, к закрытому - 4.

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода add\_1\_4 класса Object

№	Предикат	Действия	№ перехода
1		Присваивание полю open суммы значения поля open и 1	2
2		Присваивание полю closed суммы значения поля closed и 4	∅

### 3.5 Алгоритм метода call\_add\_5\_7 класса Object

Функционал: вызвать метод add\_5\_7.

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода call\_add\_5\_7 класса Object

№	Предикат	Действия	№ перехода
1		Вызов метода add_5_7	∅

### 3.6 Алгоритм метода print класса Object

Функционал: вывести на экран значения открытого и закрытого полей.

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода print класса Object

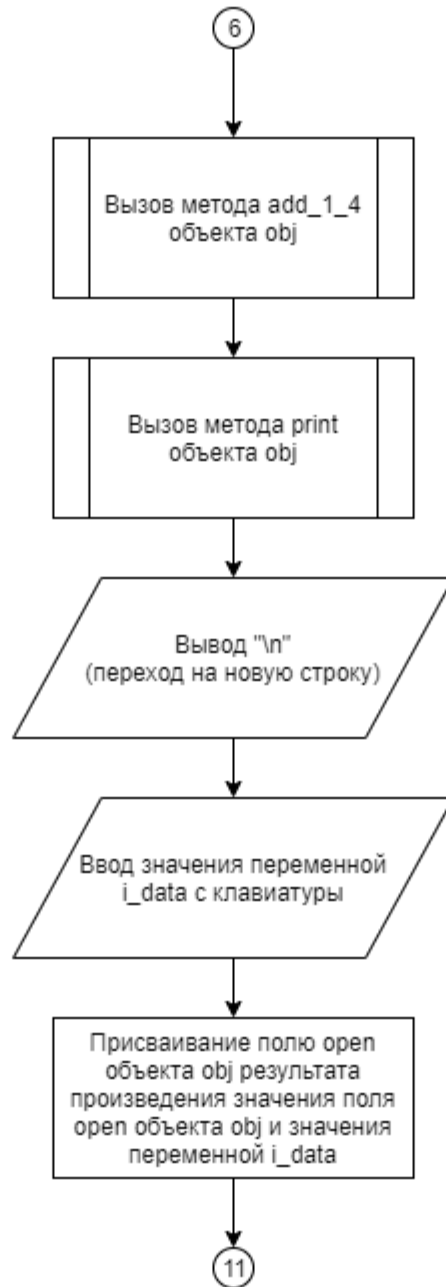
№	Предикат	Действия	№ перехода
1		Вывод сообщения "Value of the available property ", значение поля open, "; Value of a hidden property ", значение поля closed.	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

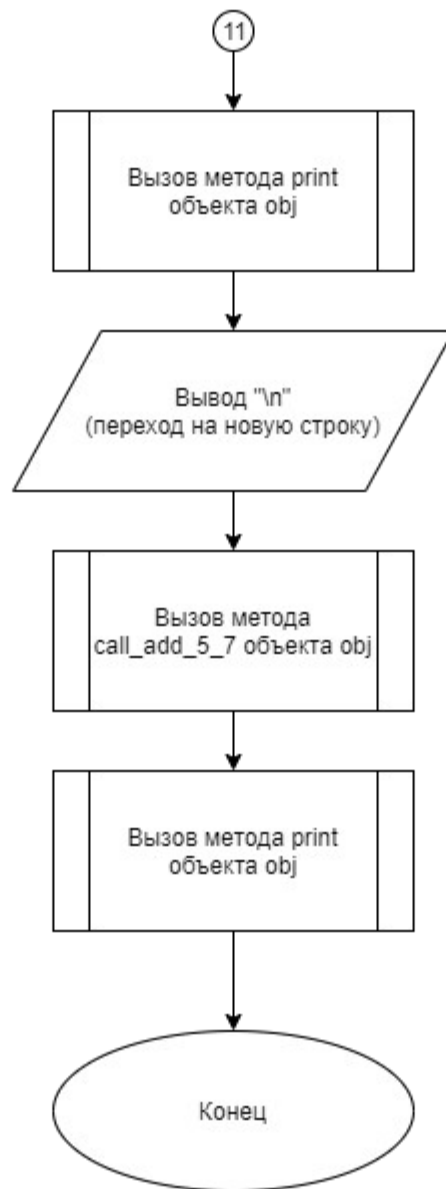
Представим описание алгоритмов в графическом виде на рисунках 1-8.



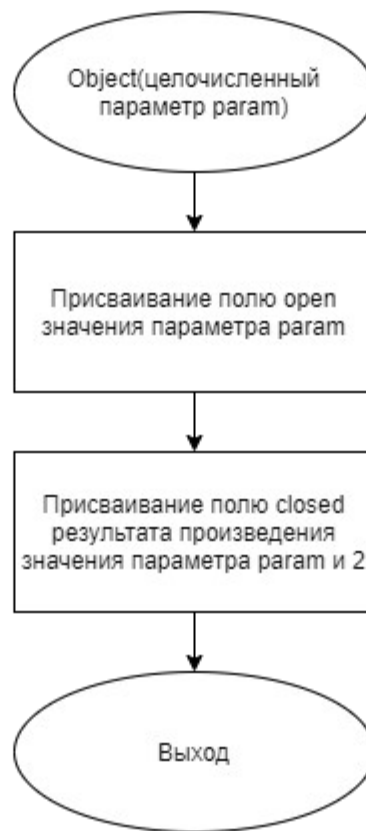
Рисунок 1 – Блок-схема алгоритма



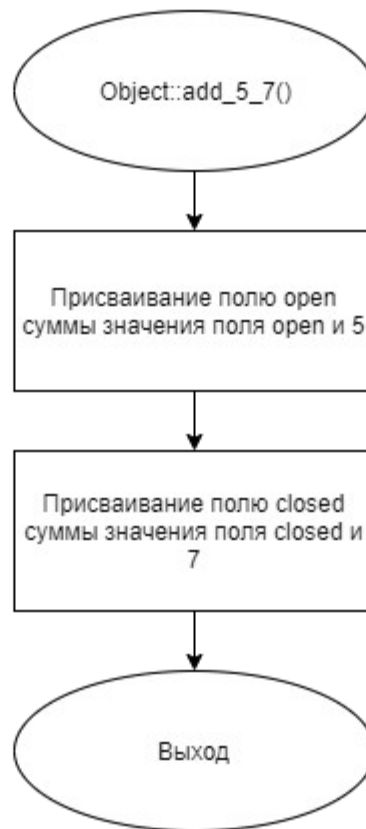
**Рисунок 2 – Блок-схема алгоритма**



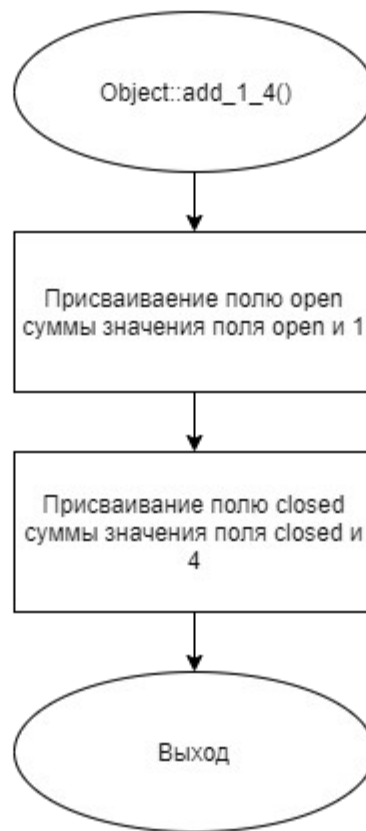
**Рисунок 3 – Блок-схема алгоритма**



**Рисунок 4 – Блок-схема алгоритма**

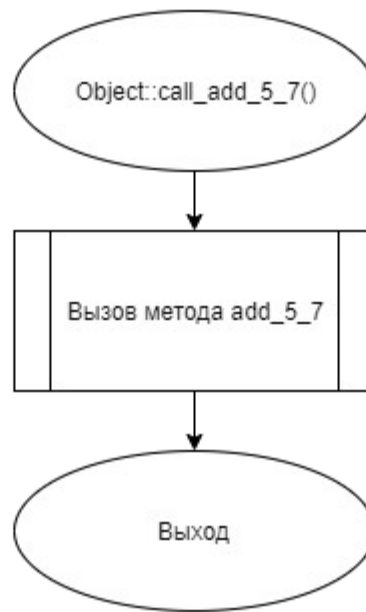


**Рисунок 5 – Блок-схема алгоритма**

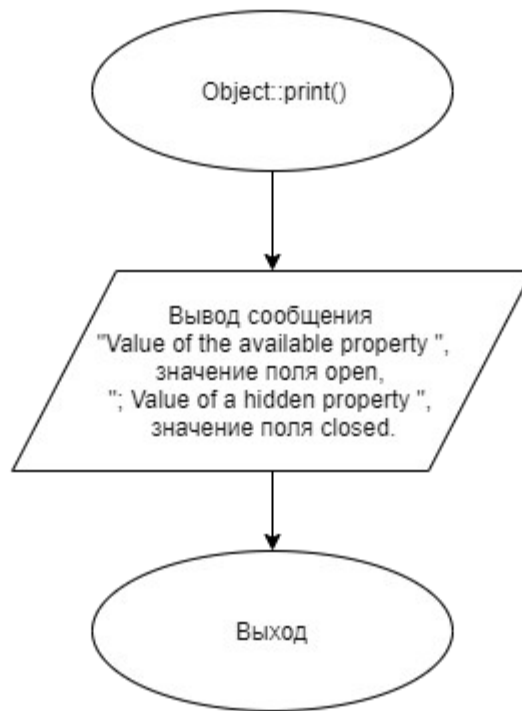


**Рисунок 6 – Блок-схема алгоритма**





**Рисунок 7 – Блок-схема алгоритма**



**Рисунок 8 – Блок-схема алгоритма**

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

```
#include <iostream>
#include "Object.h"

int main()
{
    // Инициализация целочисленной
    // переменной i_data значением 0
    int i_data = 0;
    // Ввод i_data
    std::cin >> i_data;

    // Создание объекта obj
    // типа Object с использованием
    // параметризованного конструктора
    Object obj(i_data);
    // Вызов метода print
    // объекта obj
    obj.print();
    // Переход на новую строку
    std::cout << '\n';

    // Вызов метода add_1_4
    // объекта obj
    obj.add_1_4();
    // Вызов метода print
    // объекта obj
    obj.print();
    // Переход на новую строку
    std::cout << '\n';

    // Ввод i_data
    std::cin >> i_data;
    // Умножение поля open
    // объекта obj на значение i_data
    obj.open = obj.open * i_data;
    // Вызов метода print
    // объекта obj
    obj.print();
    // Переход на новую строку
    std::cout << '\n';
```

```

        // Вызов метода call_add_5_7
        // объекта obj
        obj.call_add_5_7();
        // Вызов метода print
        // объекта obj
        obj.print();

        return 0;
}

```

## 5.2 Файл Object.cpp

*Листинг 2 – Object.cpp*

```

#include "Object.h"
#include <iostream>

void Object::add_5_7()
{
    open += 5;
    closed += 7;
}

Object::Object(int param)
{
    open = param;
    closed = param * 2;
}

void Object::add_1_4()
{
    open += 1;
    closed += 4;
}

void Object::call_add_5_7()
{
    add_5_7();
}

void Object::print()
{
    std::cout << "Value of the available property " << open
               << "; Value of a hidden property " << closed;
}

```

## 5.3 Файл Object.h

*Листинг 3 – Object.h*

```
#ifndef OBJECT_H
#define OBJECT_H

class Object
{
private:
    int closed;

    void add_5_7();

public:
    int open;

    Object(int param);
    void add_1_4();
    void call_add_5_7();
    void print();
};

#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 2	Value of the available property 1; Value of a hidden property 2 Value of the available property 2; Value of a hidden property 6 Value of the available property 4; Value of a hidden property 6 Value of the available property 9; Value of a hidden property 13	Value of the available property 1; Value of a hidden property 2 Value of the available property 2; Value of a hidden property 6 Value of the available property 4; Value of a hidden property 6 Value of the available property 9; Value of a hidden property 13

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] — URL: [https://mirea.aco-avrova.ru/student/files/methodicheskoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avrova.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_rabot_3.pdf) (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: [https://mirea.aco-avrova.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avrova.ru/student/files/Prilozheniye_k_methodichke.pdf) (дата обращения 05.05.2021).
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).