

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.....	8
3.1 Алгоритм функции main.....	8
3.2 Алгоритм метода summon класса object.....	9
3.3 Алгоритм метода print класса object.....	9
3.4 Алгоритм метода closed_method класса object.....	9
3.5 Алгоритм конструктора класса object.....	10
3.6 Алгоритм деструктора класса object.....	10
3.7 Алгоритм метода plus класса object.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл object.cpp.....	14
5.3 Файл object.h.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18

# 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

Для решения задачи используется:

- объект `object_first` класса `object`;
- `cin` - объект стандартного потока ввода с клавиатуры;
- `cout` - объект стандартного потока вывода на экран.

Класс `object`:

- свойства/поля:
  - поле Открытое поле:
    - наименование — `opened`;
    - тип — `int`;
    - модификатор доступа — `public`;
  - поле Закрытое поле:
    - наименование — `closed`;
    - тип — `int`;
    - модификатор доступа — `private`;
- функционал:
  - метод `plus` — `opened +1, closed +4`;
  - метод `summon` — вызов скрытого метода `closed_method`;
  - метод `print` — вывод на экран значения состояния `opened` и `closed`;
  - метод `closed_method` — `opened +5, closed +7`;
  - метод `object` — параметризованный конструктор;
  - метод `~object` — деструктор.

## 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: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменной i_data типа int	2
2		ввод значения i_data с клавиатуры	3
3		создание объекта object_first класса object с помощью параметризованного конструктора с аргументом i_data	4
4		вызов метода print() объекта object_first	5
5		вызов метода plus() объекта object_first	6
6		вызов метода print() объекта object_first	7
7		ввод значения i_data с клавиатуры	8
8		непосредственное изменение opened объекта object_first посредством умножения его значения на i_data	9
9		вызов метода print() объекта object_first	10
10		вызов метода summon() объекта object_first	11
11		вызов метода print() объекта object_first	12
12		возврат значения 0	Ø

### 3.2 Алгоритм метода `summon` класса `object`

Функционал: вызов скрытого метода `closed_method`.

Параметры: нет.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `summon` класса `object`

№	Предикат	Действия	№ перехода
1		вызов скрытого метода <code>closed_method</code>	Ø

### 3.3 Алгоритм метода `print` класса `object`

Функционал: вывод на экран значения состояния `opened` и `closed`.

Параметры: нет.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода `print` класса `object`

№	Предикат	Действия	№ перехода
1		Вывод на экран "Value of the available property " <code>opened</code>	2
2		Вывод на экран "Value of a hiddenproperty " <code>closed</code>	Ø

### 3.4 Алгоритм метода `closed_method` класса `object`

Функционал: `opened +5`, `closed +7`.

Параметры: нет.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *closed\_method* класса *object*

№	Предикат	Действия	№ перехода
1		Увеличение значения поля <i>opened</i> на 5	2
2		Увеличение значения поля <i>closed</i> на 7	∅

### 3.5 Алгоритм конструктора класса *object*

Функционал: параметризированный конструктор.

Параметры: *int value* - параметр целого типа.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса *object*

№	Предикат	Действия	№ перехода
1		инициализация <i>opened</i> значением <i>value</i>	2
2		инициализация <i>closed</i> значением <i>value</i>	∅

### 3.6 Алгоритм деструктора класса *object*

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 6.

Таблица 6 – Алгоритм деструктора класса *object*

№	Предикат	Действия	№ перехода
1			∅

### 3.7 Алгоритм метода *plus* класса *object*

Функционал: *opened +1, closed +4.*

Параметры: нет.



Возвращаемое значение: void.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *plus* класса *object*

№	Предикат	Действия	№ перехода
1		увеличение значения поля <i>opened</i> на 1	2
2		увеличение значения поля <i>closed</i> на 4	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

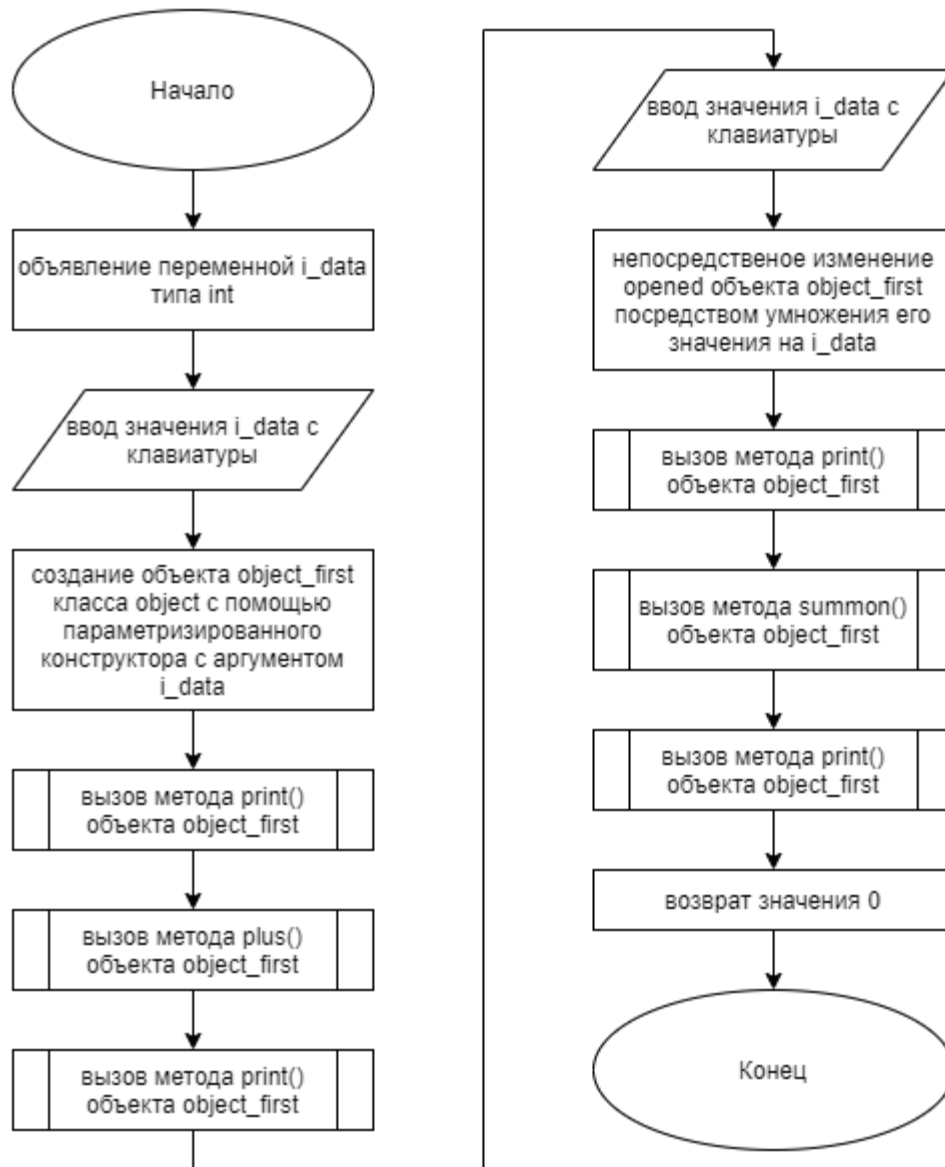


Рисунок 1 – Блок-схема алгоритма

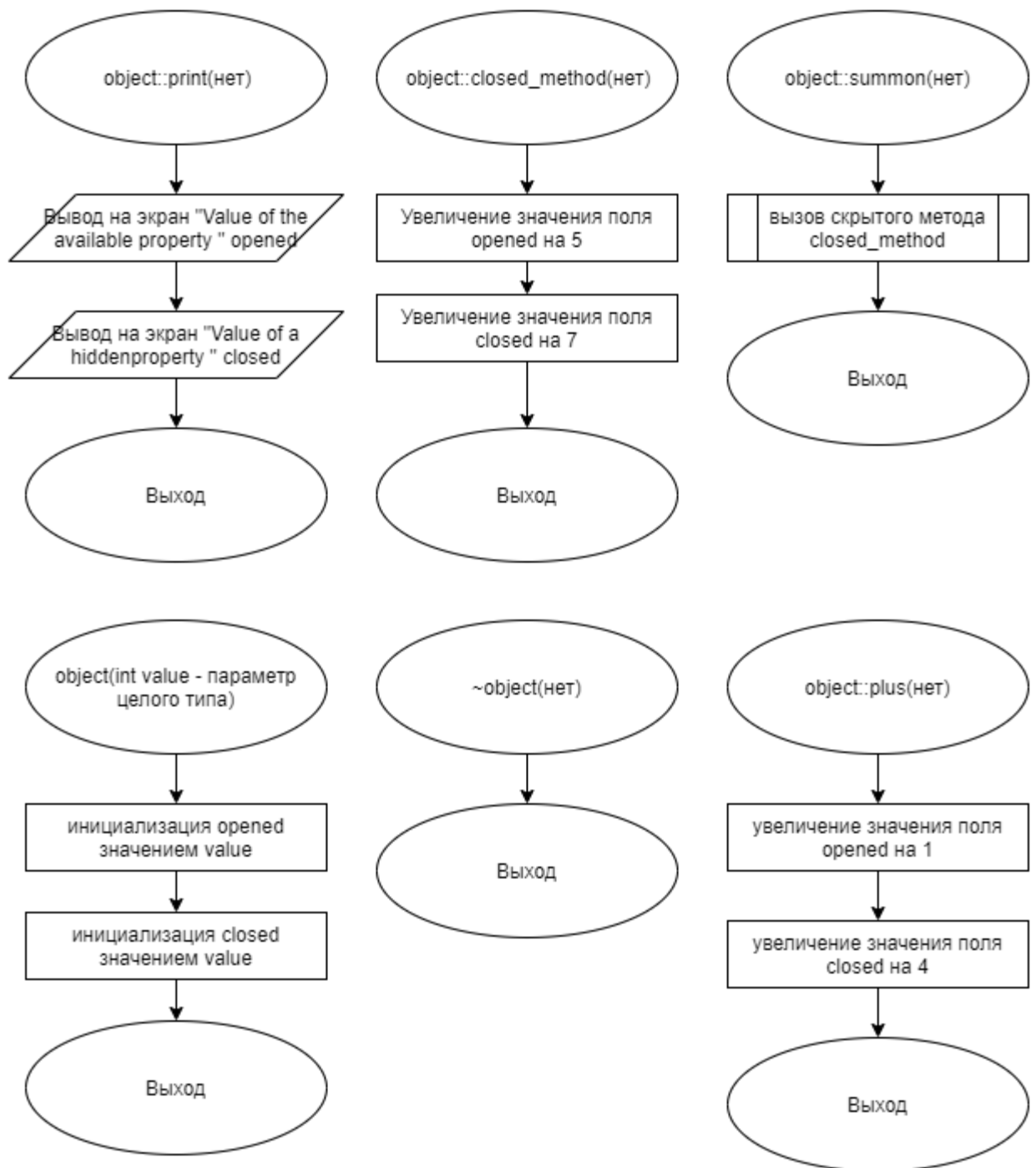


Рисунок 2 – Блок-схема алгоритма

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

int main()
{
    int i_data;
    std::cin >> i_data;
    object object_first(i_data);
    object_first.print();
    object_first.plus();
    object_first.print();
    std::cin >> i_data;
    object_first.opened *= i_data;
    object_first.print();
    object_first.summon();
    object_first.print();
    return(0);
}
```

### 5.2 Файл object.cpp

*Листинг 2 – object.cpp*

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

object::object(int value) {
    opened = value;
    closed = 2 * value;
}

object::~~object() {}
```

```

void object::plus() {
    opened += 1;
    closed += 4;
}

void object::closed_method() {
    opened += 5;
    closed += 7;
}

void object::summon() {
    closed_method();
}

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

```

### 5.3 Файл object.h

*Листинг 3 – object.h*

```

#ifndef __OBJECT__H
#define __OBJECT__H

class object {
private:
    int closed;
    void closed_method();
public:
    object(int);
    ~object();
    int opened;
    void plus();
    void summon();
    void print();
};

#endif

```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 8.

Таблица 8 – 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
3 5	Value of the available property 3; Value of a hidden property 6 Value of the available property 4; Value of a hidden property 10 Value of the available property 20; Value of a hidden property 10 Value of the available property 25; Value of a hidden property 17	Value of the available property 3; Value of a hidden property 6 Value of the available property 4; Value of a hidden property 10 Value of the available property 20; Value of a hidden property 10 Value of the available property 25; Value of a hidden property 17
10 27	Value of the available property 10; Value of a hidden property 20 Value of the available property 11; Value of a hidden property 24 Value of the available property 297; Value of a	Value of the available property 10; Value of a hidden property 20 Value of the available property 11; Value of a hidden property 24 Value of the available property 297; Value of a

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	hidden property 24 Value of the available property 302; Value of a hidden property 31	hidden property 24 Value of the available property 302; Value of a hidden property 31

## 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/methodichescoc\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avvora.ru/student/files/methodichescoc_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](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).