

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 конструктора класса Object.....	9
3.3 Алгоритм метода add_7_5 класса Object.....	9
3.4 Алгоритм метода add_4_1 класса Object.....	10
3.5 Алгоритм метода call_add_7_5 класса Object.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	19
5.1 Файл main.cpp.....	19
5.2 Файл Object.cpp.....	20
5.3 Файл Object.h.....	21
6 ТЕСТИРОВАНИЕ.....	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	23

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 утроенное значение параметра.

У объекта есть доступные методы со следующим функционалом:

- Метод изменения значения доступного и скрытого свойства. Доступному свойству добавляется 4, скрытому свойству добавляется 1;
- Метод вызова закрытого метода;
- Метод вывода состояния, выводит значение доступного и скрытого свойства.

У объекта есть один скрытый метод со следующим функционалом:

- Метод доступному свойству добавляется 7, скрытому свойству добавляется 5.

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

- Ввод целочисленного значения переменной `i_data`.
- Создание объекта посредством оператора функции `new` и использованием указателя на объект, параметризованному конструктору в качестве аргумента передается переменная `i_data`.
- Вывод исходного состояния объекта.
- Вызов метода изменения значений свойств объекта.
- Вывод текущего состояния объекта.

- Ввод целочисленного значения переменной i_data.
- Если значение i_data больше, чем значение доступного свойства объекта, то
 - Присвоение доступному свойству объекта значение выражения i_data * 8 и переход к пункту 9.
- Иначе
 - Переход к пункту 9.
- Вывод текущего состояния объекта.
- Вызов метода объекта, который вызывает скрытый метод объекта.
- 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 с новой. Шаблон вывода:

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

2 МЕТОД РЕШЕНИЯ

Класс Object:

- Методы:
 - Конструктор Object:
 - Функционал - создать объект с открытым полем, равным значению параметра, и закрытым, равным утроенному значению параметра;
 - Возвращаемое значение - void (отсутствует);
 - Модификатор доступа - открытый;
 - Параметра - целочисленный параметр param;
 - Метод add_7_5:
 - Функционал - добавить к открытому полю 7, а к закрытому - 5;
 - Возвращаемое значение - void (отсутствует);
 - Модификатор доступа - закрытый;
 - Параметры - отсутствуют;
 - Метод add_4_1:
 - Функционал - добавить к открытому полю 4, а к закрытому - 1;
 - Возвращаемое значение - void (отсутствует);
 - Модификатор доступа - открытый;
 - Параметры - отсутствуют;
 - Метод call_add_7_5:
 - Функционал - вызывать метод add_7_5;
 - 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 i_data с клавиатуры	3
3		Инициализация указателя obj адресом нового объекта типа Object созданного с помощью оператора new	4
4		Вызов метода print объекта по адресу obj	5
5		Вывод "\n" (переход на новую строку)	6
6		Вызов метода add_4_1 объекта по адресу obj	7
7		Вызов метода print объекта по адресу obj	8
8		Вывод "\n" (переход на новую строку)	9
9		Ввод значения i_data с клавиатуры	10
10	Значения переменной i_data больше значения поля open	Присваивание полю open объекта по адресу obj результата произведения 8 и значения i_data	11

№	Предикат	Действия	№ перехода
	объекта по адресу obj		
			11
1 1		Вызов метода print объекта по адресу obj	12
1 2		Вывод "\n" (переход на новую строку)	13
1 3		Вызов метода call_add_7_5 объекта по адресу obj	14
1 4		Вызов метода print объекта по адресу obj	15
1 5		Удаление объекта по адресу obj с помощью оператора delete	Ø

3.2 Алгоритм конструктора класса Object

Функционал: создать объект класса Object с открытым полем равным param и закрытым равным утроенному значению param.

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 3 и значения параметра param	Ø

3.3 Алгоритм метода add_7_5 класса Object

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

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода add_7_5 класса Object

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

3.4 Алгоритм метода add_4_1 класса Object

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

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода add_4_1 класса Object

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

3.5 Алгоритм метода call_add_7_5 класса Object

Функционал: вызвать метод add_7_5.

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *call_add_7_5* класса *Object*

№	Предикат	Действия	№ перехода
1		Вызов метода <i>add_7_5</i>	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-7.

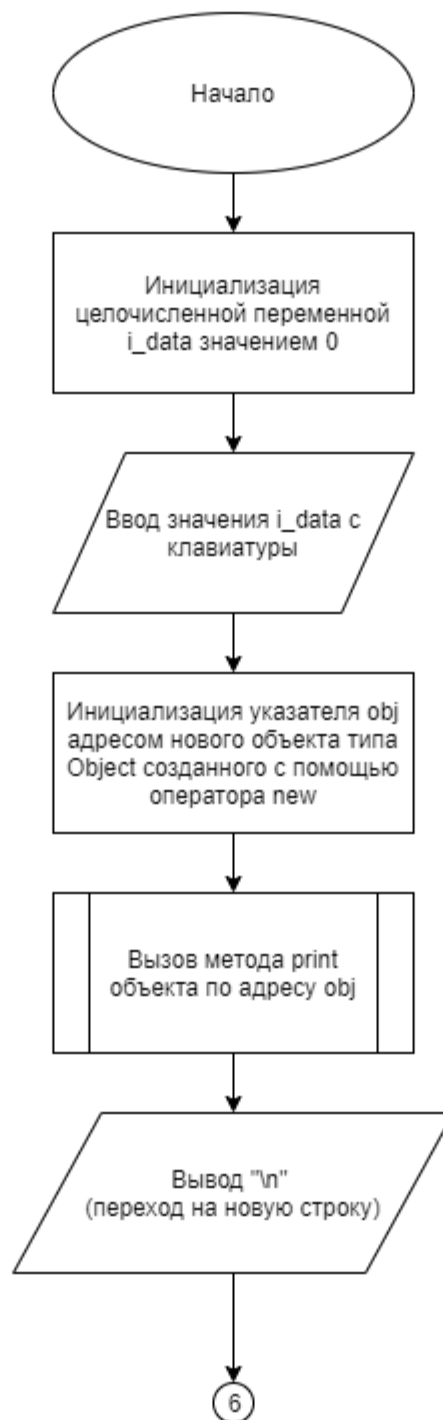


Рисунок 1 – Блок-схема алгоритма

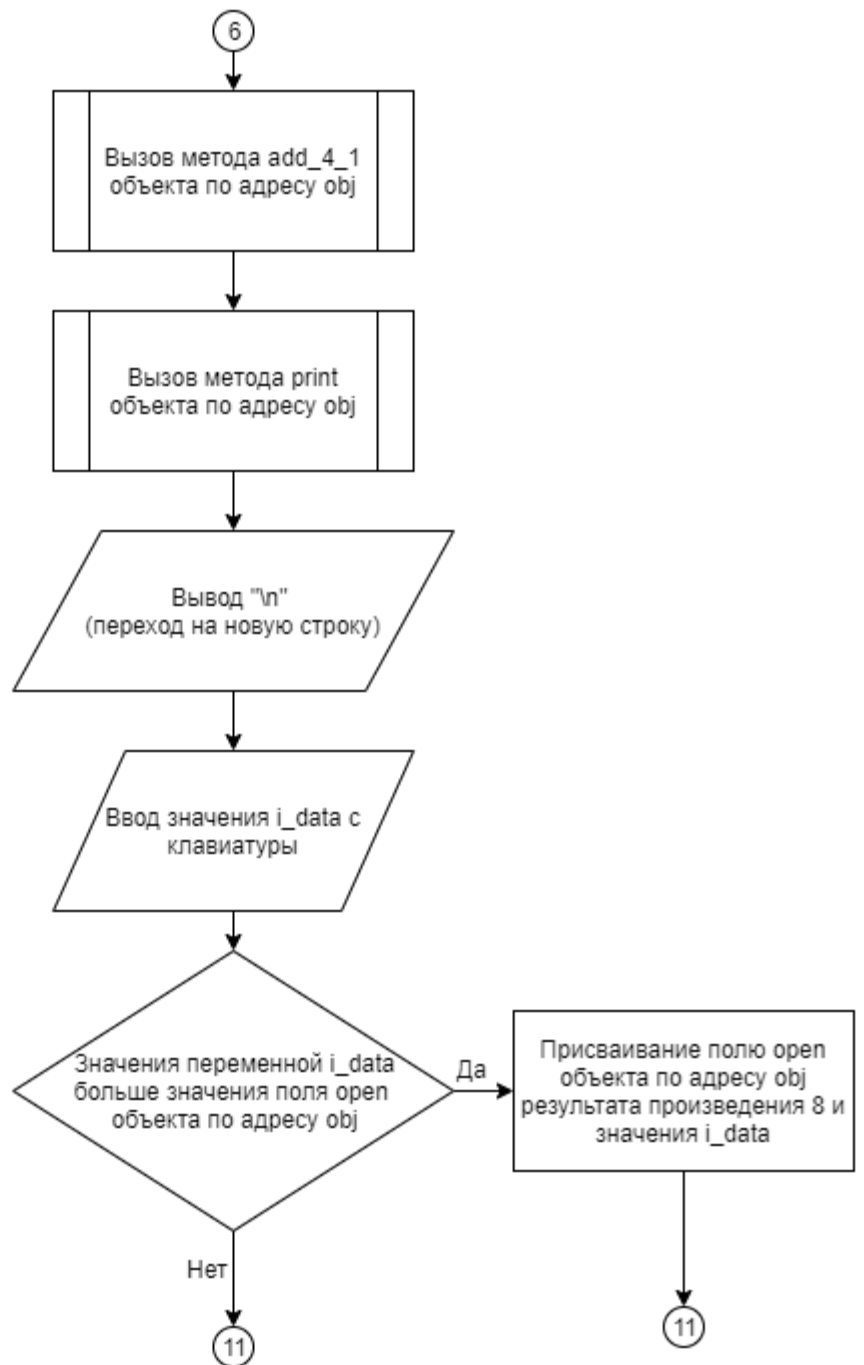


Рисунок 2 – Блок-схема алгоритма

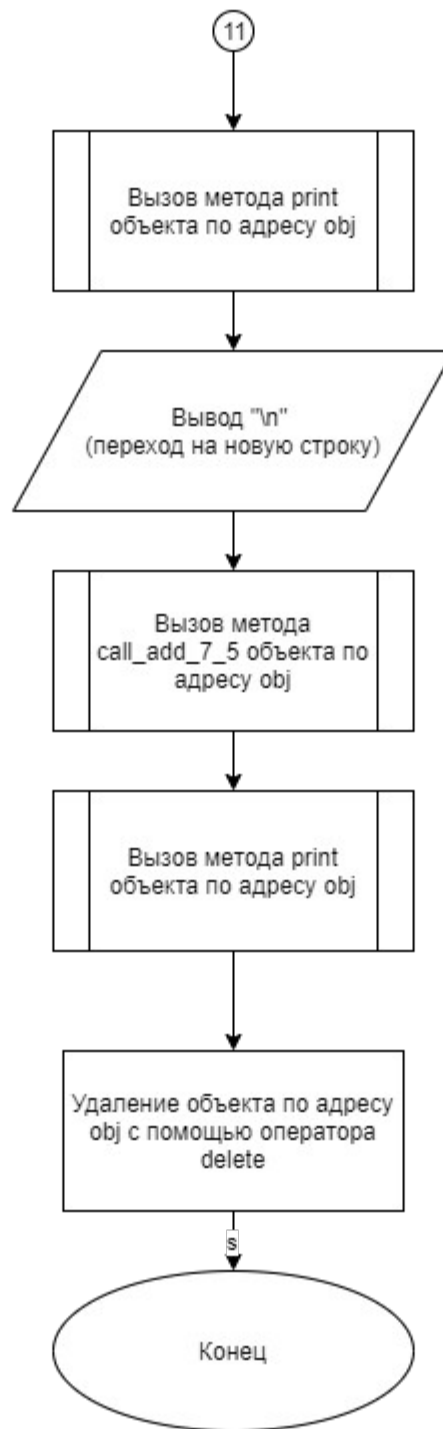


Рисунок 3 – Блок-схема алгоритма

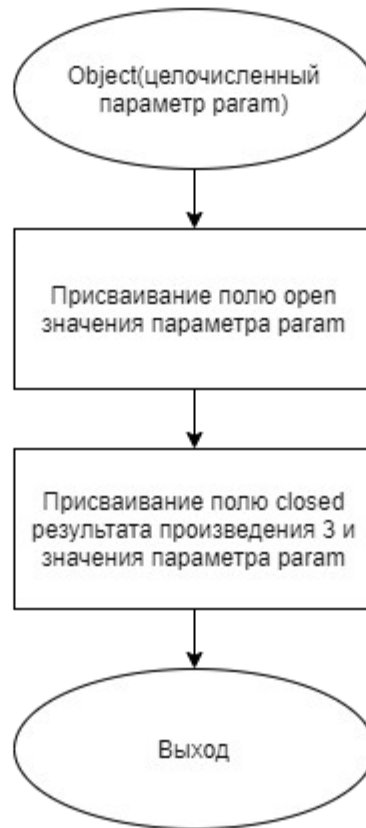


Рисунок 4 – Блок-схема алгоритма

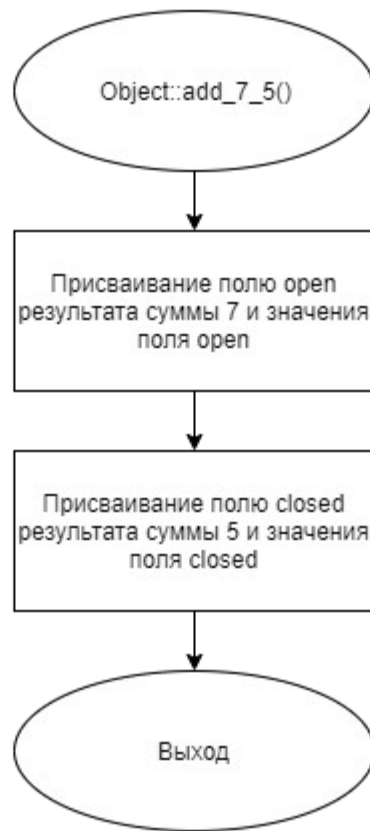


Рисунок 5 – Блок-схема алгоритма



Рисунок 6 – Блок-схема алгоритма

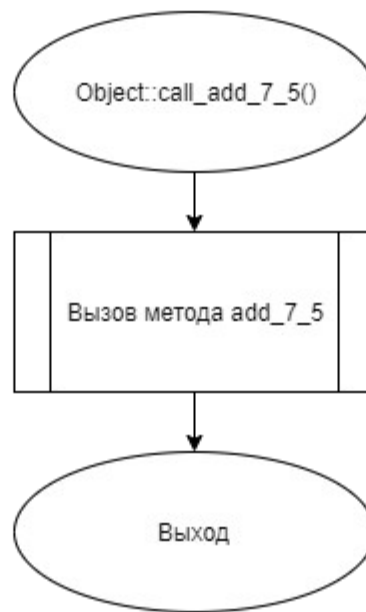


Рисунок 7 – Блок-схема алгоритма

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 = new Object(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_4_1();
    // Вызов метода print
    // объекта obj
    obj->print();
    // Переход на новую строку
    std::cout << '\n';

    // Ввод i_data
    std::cin >> i_data;
    if (i_data > obj->open)
    {
        // Умножение поля open
        // объекта obj на значение i_data
        obj->open = i_data * 8;
    }
    // Вызов метода print
    // объекта obj
    obj->print();
}
```

```

        // Переход на новую строку
        std::cout << '\n';

        // Вызов метода call_add_7_5
        // объекта obj
        obj->call_add_7_5();
        // Вызов метода print
        // объекта obj
        obj->print();

        // Удаление объекта obj
        delete obj;

        return 0;
}

```

5.2 Файл Object.cpp

Листинг 2 – Object.cpp

```

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

void Object::add_7_5()
{
    open += 7;
    closed += 5;
}

Object::Object(int param)
{
    open = param;
    closed = param * 3;
}

void Object::add_4_1()
{
    open += 4;
    closed += 1;
}

void Object::call_add_7_5()
{
    add_7_5();
}

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_7_5();

public:
    int open;

    Object(int param);
    void add_4_1();
    void call_add_7_5();
    void print();
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1	Value of the available	Value of the available
2	property 1; Value of a	property 1; Value of a
	hidden property 3	hidden property 3
	Value of the available	Value of the available
	property 5; Value of a	property 5; Value of a
	hidden property 4	hidden property 4
	Value of the available	Value of the available
	property 5; Value of a	property 5; Value of a
	hidden property 4	hidden property 4
	Value of the available	Value of the available
	property 12; Value of a	property 12; Value of a
	hidden property 9	hidden property 9

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-avroora.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-avroora.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).