

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Постановка задачи.....	5
2 Метод решения.....	8
3 Описание алгоритма.....	11
4 Блок-схема алгоритма.....	12
5 Код программы.....	14
6 Тестирование.....	18
ЗАКЛЮЧЕНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

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:

1. Ввод целочисленного значения переменной `i_data`.
2. Создание объекта посредством оператора функции `new` и использованием указателя на объект, 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. Если значение `i_data` больше, чем значение доступного свойства объекта, то
 - 7.1. Присвоение доступному свойству объекта значение выражения `i_data * 8` и переход к пункту 9.
8. Иначе
 - 8.1. Переход к пункту 9.
9. Вывод текущего состояния объекта.
10. Вызов метода объекта, который вызывает скрытый метод объекта.
11. 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

Оператор функция new;

Оператор функция delete;

Класс Calc:

- Функционал:
 - Метод Change() - прибавляет к доступному свойству 4, а к скрытому свойству прибавляет 1;
 - Метод Private() - прибавляет к доступному свойству 7, а к скрытому свойству прибавляет 5.

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 конструктора класса Calc

Функционал: доступному свойству присваивается значение параметра, а скрытому свойству утроенное значение параметра.

Параметры: int valueData.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Calc

№	Предикат	Действия	№ перехода
1		publicData = valueData	2
2		privateData = valueData * 3	∅

3.2 Алгоритм метода Change класса Calc

Функционал: доступному свойству добавляется 4, скрытому свойству добавляется 1.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода Change класса Calc

№	Предикат	Действия	№ перехода
1		publicData += 4	2

№	Предикат	Действия	№ перехода
2		privateData += 1	∅

3.3 Алгоритм метода Private класса Calc

Функционал: доступному свойству добавляется 7, скрытому свойству добавляется 5.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Private класса Calc

№	Предикат	Действия	№ перехода
1		publicData += 7	2
2		privateData += 5	∅

3.4 Алгоритм функции main

Функционал: 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. 9. Вывод текущего состояния объекта. Вызов метода объекта, который вызывает скрытый метод объекта. Вывод текущего состояния объекта..

Параметры: .

Возвращаемое значение: int - код возврата.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>i_data</i>	2
2		Ввод значения переменной <i>i_data</i>	3
3		Создание объекта класса <i>Calc</i> с аргументом <i>i_data</i> Инициализация указателя <i>Obj</i> адресом этого объекта	4
4		Вызов метода <i>Print(0)</i> объекта по адресу <i>Obj</i>	5
5		Вызов метода <i>Change()</i> объекта по адресу <i>Obj</i>	6
6		Вызов метода <i>Print(0)</i> объекта по адресу <i>Obj</i>	7
7		Ввод значения переменной <i>i_data</i>	8
8	<i>i_data</i> больше значения доступного свойства объекта <i>Obj</i>	Присвоить доступному свойству объекта <i>Obj</i> значение выражения <i>i_data * 8</i>	9
			9
9		Вызов метода <i>Print(0)</i> объекта по адресу <i>Obj</i>	10
10		Вызов метода <i>CallPrivate()</i> объекта по адресу <i>Obj</i>	11
11		Вызов метода <i>Print(1)</i> объекта по адресу <i>Obj</i>	12
12		Удаление объекта по адресу указателя <i>Obj</i> при помощи оператора функции <i>delete</i>	13
13		Возвратить 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

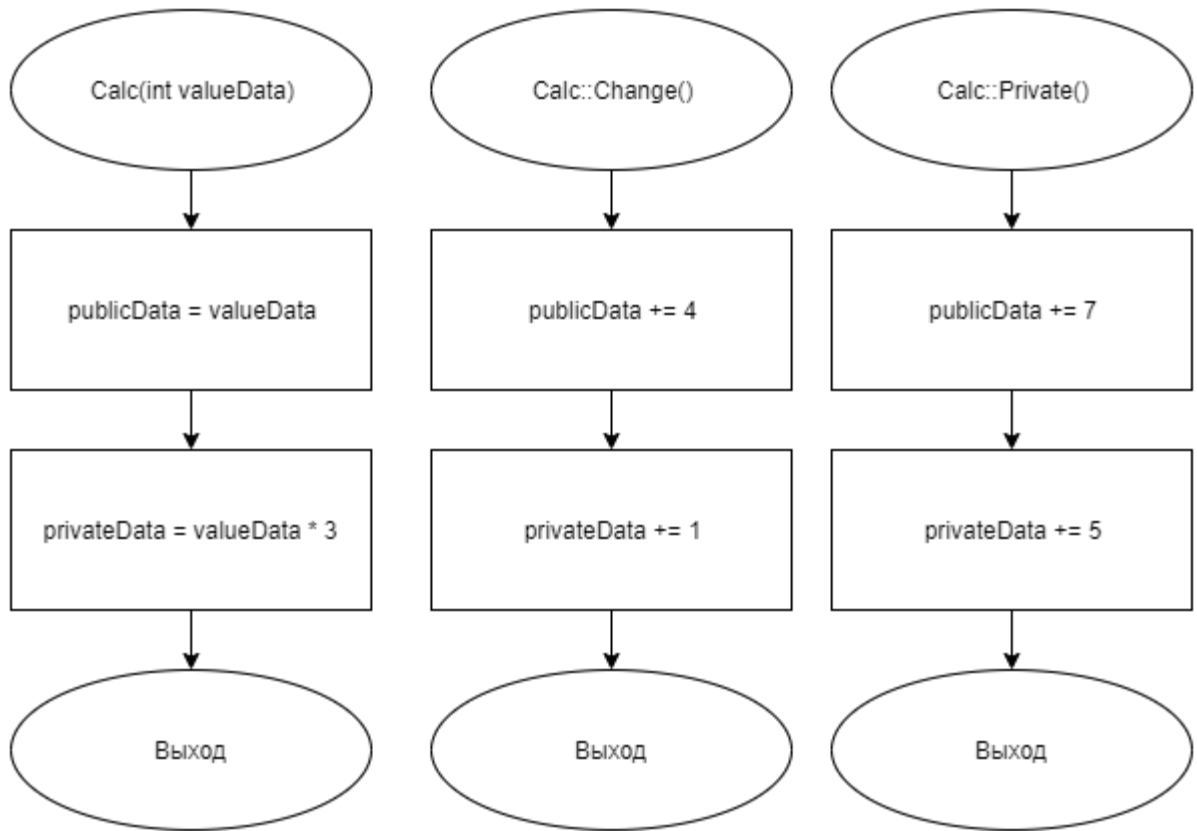


Рисунок 1 – Блок-схема алгоритма

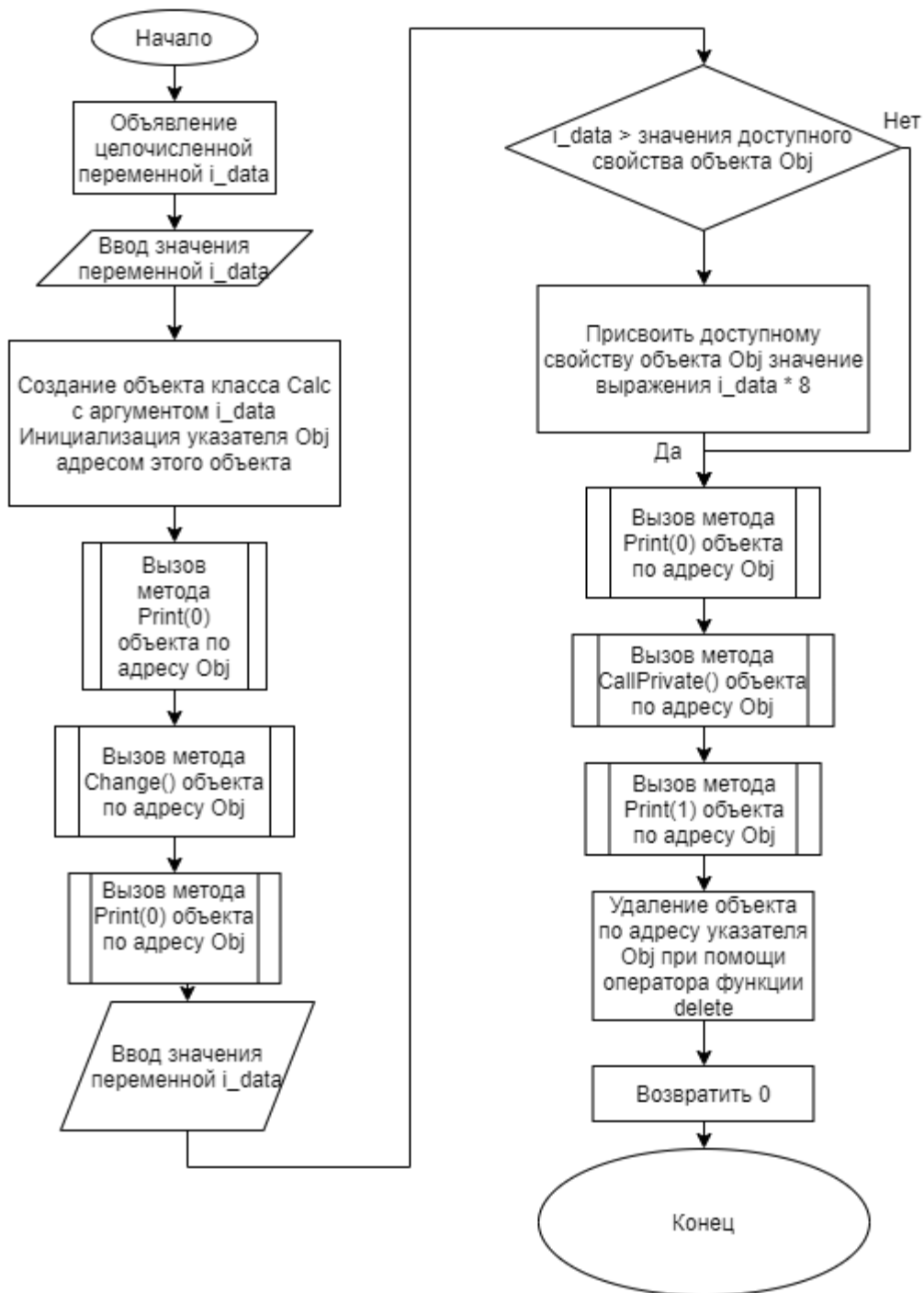


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Calc.cpp

Листинг 1 – Calc.cpp

```
#include <iostream>
#include "Calc.h"
using namespace std;
Calc::Calc(int valueData) {
    publicData = valueData;
    privateData = valueData * 3;
};
void Calc::Change() {
    publicData += 4;
    privateData++;
};
void Calc::CallPrivate() {
    Private();
};
void Calc::Print(int last) {
    cout << "Value of the available property ";
    cout << publicData;
    cout << "; Value of a hidden property ";
    cout << privateData;
    if (last == 0) {
        cout << endl;
    }
};
void Calc::Private(){
    publicData += 7;
    privateData += 5;
};
```

5.2 Файл Calc.h

Листинг 2 – Calc.h

```
#ifndef CALC_H
#define CALC_H
class Calc
{
private:
```

```
        int privateData;
        void Private();
public:
        Calc(int valueData);
        int publicData;
        void Change();
        void CallPrivate();
        void Print(int last);
};
#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <iostream>
#include "Calc.h"
using namespace std;

int main() {
    int i_data;
    cin >> i_data;
    Calc *Obj = new Calc(i_data);
    Obj -> Print(0);
    Obj -> Change();
    Obj -> Print(0);
    cin >> i_data;
    if (i_data > Obj -> publicData) {
        Obj -> publicData = i_data * 8;
    }
    Obj -> Print(0);
    Obj -> CallPrivate();
    Obj -> Print(1);
    delete Obj;
    return 0;
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 8	Value of the available property 5; Value of a hidden property 15 Value of the available property 9; Value of a hidden property 16 Value of the available property 9; Value of a hidden property 16 Value of the available property 16; Value of a hidden property 21	Value of the available property 5; Value of a hidden property 15 Value of the available property 9; Value of a hidden property 16 Value of the available property 9; Value of a hidden property 16 Value of the available property 16; Value of a hidden property 21
5 10	Value of the available property 5; Value of a hidden property 15 Value of the available property 9; Value of a hidden property 16 Value of the available property 80; Value of a hidden property 16 Value of the available property 87; Value of a hidden property 21	Value of the available property 5; Value of a hidden property 15 Value of the available property 9; Value of a hidden property 16 Value of the available property 80; Value of a hidden property 16 Value of the available property 87; Value of a hidden property 21

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-avrora.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-avrora.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).