

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 конструктора класса Object.....	9
3.2 Алгоритм метода Plus класса Object.....	9
3.3 Алгоритм метода PrivateCall класса Object.....	10
3.4 Алгоритм метода PrintProperty класса Object.....	10
3.5 Алгоритм метода ChangeProperty класса Object.....	10
3.6 Алгоритм функции main.....	11
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	15
5.1 Файл main.cpp.....	15
5.2 Файл Object.cpp.....	15
5.3 Файл Object.h.....	16
6 ТЕСТИРОВАНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

# 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

Для решения задачи используется:

- объект `Obj_est` класса `Object` предназначен для Объект с доступными и скрытыми элементами;
- функция `main` для Главная функция программы;
- объект `cout` стандартного потока вывода на экран;
- объект `cin` стандартного потока ввода с клавиатуры;
- библиотека `iostream`;
- пространство имён `std`.

Класс `Object`:

- свойства/поля:
  - поле Хранение некоторого значения:
    - наименование — `dataPublic`;
    - тип — `int`;
    - модификатор доступа — `public`;
  - поле Хранение некоторого значения:
    - наименование — `dataPrivate`;
    - тип — `int`;
    - модификатор доступа — `private`;
- функционал:
  - метод `Object` — Параметризированный конструктор для создания объекта класса и изменения значений полей с помощью параметра;
  - метод `Plus` — Изменение значений доступного и скрытого полей;
  - метод `PrivateCall` — Вызов скрытого метода `ChangePropsPrivate`;
  - метод `PrintProperty` — Вывод состояния, а именно значения доступного и скрытого поля;

о метод `ChangeProperty` — Изменение значений доступного и скрытого поля.

## 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 Object

Функционал: Параметризованный конструктор для создания объекта класса и изменения значений полей с помощью параметра.

Параметры: int data, изменение полей класса.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Присвоение значения параметра data полю dataPublic	2
2		Присвоение значения параметра data умноженного на 3 полю dataPrivate	Ø

### 3.2 Алгоритм метода Plus класса Object

Функционал: Изменение значений доступного и скрытого полей.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода Plus класса Object

№	Предикат	Действия	№ перехода
1		Увеличение значения поля dataPublic на 4	2

№	Предикат	Действия	№ перехода
2		Инкремент значения поля dataPrivate	Ø

### 3.3 Алгоритм метода PrivateCall класса Object

Функционал: Вызов скрытого метода ChangeProperty.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода PrivateCall класса Object

№	Предикат	Действия	№ перехода
1		Вызов приватного метода ChangeProperty класса Object	Ø

### 3.4 Алгоритм метода PrintProperty класса Object

Функционал: Вывод состояния, а именно значения доступного и скрытого поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода PrintProperty класса Object

№	Предикат	Действия	№ перехода
1		Вывод: "Value of the available property " <<значение поля dataPublic>> "; Value of a hidden property " <<значение поля dataPrivate>>	Ø

### 3.5 Алгоритм метода ChangeProperty класса Object

Функционал: Скрытый метод, используется для изменения значений



доступного и скрытого поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 5.

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

№	Предикат	Действия	№ перехода
1		Увеличение значения поля dataPublic на 7	2
2		Увеличение значения поля dataPrivate на 5	Ø

### 3.6 Алгоритм функции *main*

Функционал: индикация корректности работы программы.

Параметры: Главная функция программы.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной i_data	2
2		Ввод значения переменной i_data	3
3		Создание объекта obj класса Object путём вызова конструктора с параметром data	4
4		Вызов метода StatusOutput объекта obj	5
5		Вызов метода ChangePropsPublic объекта obj	6
6		Вызов метода StatusOutput объекта obj	7
7		Ввод значения переменной i_data	8
8	Значение переменной i_data больше значения поля dataPublic объекта obj	Присвоение полю dataPublic объекту obj значение переменной i_data, умноженное на 8	9

№	Предикат	Действия	№ перехода
	Значение переменной i_data не больше значения поля dataPublic объекта obj		9
9		Вызов метода StatusOutput объекта obj	10
10		Вызов метода PrivateCall объекта obj	11
11		Вызов метода StatusOutput объекта obj	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

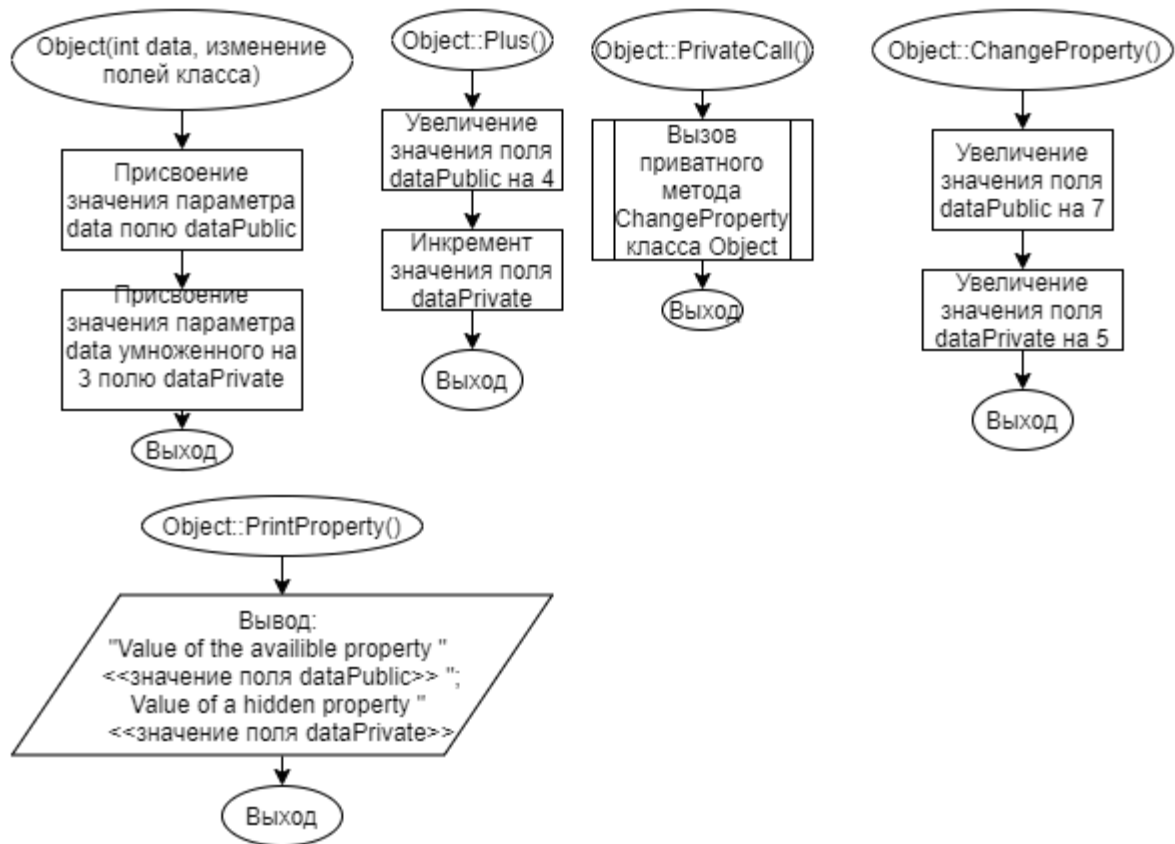


Рисунок 1 – Блок-схема алгоритма

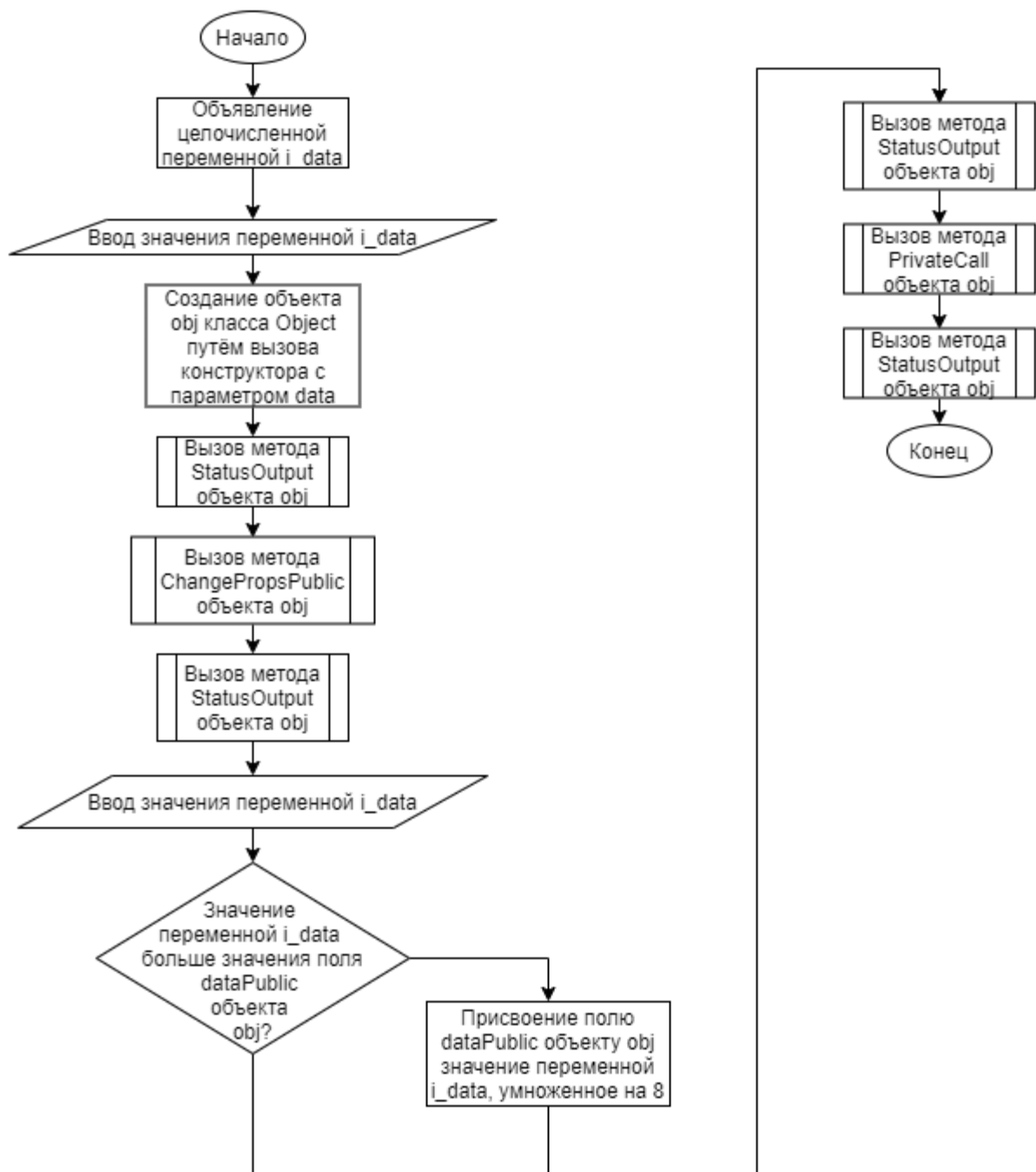


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

```
#include <iostream>
#include "Object.h"
using namespace std;
int main()
{
    int i_data;
    cin >> i_data;
    Object* Obj_ect = new Object(i_data);
    Obj_ect -> PrintProperty(0);
    Obj_ect -> Plus();
    Obj_ect -> PrintProperty(0);
    cin >> i_data;
    if (i_data > Obj_ect -> dataPublic)
    {
        Obj_ect -> dataPublic = i_data*8;
    }
    Obj_ect -> PrintProperty(0);
    Obj_ect -> PrivateCall();
    Obj_ect -> PrintProperty(0);
    delete Obj_ect;
    return(0);
}
```

### 5.2 Файл Object.cpp

*Листинг 2 – Object.cpp*

```
#include <iostream>
#include "Object.h"
using namespace std;
Object::Object(int dataValue)
{
    dataPublic = dataValue;
    dataPrivate = dataValue*3;
}
```

```

};
void Object::Plus()
{
    dataPublic += 4;
    dataPrivate ++ ;
}
void Object::PrivateCall()
{
    ChangeProperty();
}
void Object::PrintProperty(int last)
{
    cout << "Value of the available property " << dataPublic << "; Value of a
hidden property " << dataPrivate;
    if (last == 0)
    {
        cout << endl;
    }
}
void Object::ChangeProperty()
{
    dataPublic += 7;
    dataPrivate += 5;
}

```

## 5.3 Файл Object.h

*Листинг 3 – Object.h*

```

#ifndef OBJECT_H
#define OBJECT_H
class Object
{
public:
    int dataPublic;
    Object(int dataValue);
    void Plus();
    void PrivateCall();
    void PrintProperty(int last);
private:
    int dataPrivate;
    void ChangeProperty();
};
#endif

```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – 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
1 2	Value of the available property 1; Value of a hidden property 3 Value of the available property 5; Value of a hidden property 4 Value of the available property 5; Value of a hidden property 4 Value of the available property 12; Value of a hidden property 9	Value of the available property 1; Value of a hidden property 3 Value of the available property 5; Value of a hidden property 4 Value of the available property 5; Value of a hidden property 4 Value of the available property 12; Value of a hidden property 9
2024 1036	Value of the available property 2024; Value of a hidden property 6072 Value of the available property 2028; Value of a hidden property 6073 Value of the available property 2028; Value of a	Value of the available property 2024; Value of a hidden property 6072 Value of the available property 2028; Value of a hidden property 6073 Value of the available property 2028; Value of a

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	hidden property 6073 Value of the available property 2035; Value of a hidden property 6078	hidden property 6073 Value of the available property 2035; Value of a hidden property 6078
-9 15	Value of the available property -9; Value of a hidden property -27 Value of the available property -5; Value of a hidden property -26 Value of the available property 120; Value of a hidden property -26 Value of the available property 127; Value of a hidden property -21	Value of the available property -9; Value of a hidden property -27 Value of the available property -5; Value of a hidden property -26 Value of the available property 120; Value of a hidden property -26 Value of the available property 127; Value of a hidden property -21



## 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](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](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).