

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 changeProperties класса Object.....	9
3.3 Алгоритм метода callHiddenMethod класса Object.....	10
3.4 Алгоритм метода printState класса Object.....	10
3.5 Алгоритм метода hiddenMethod класса Object.....	10
3.6 Алгоритм функции main.....	11
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	17
5.1 Файл main.cpp.....	17
5.2 Файл Object.cpp.....	18
5.3 Файл Object.h.....	18
6 ТЕСТИРОВАНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

# 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` класса `Object` предназначен для изменение свойств, вызов скрытого метода и вывод текущего состояния объекта;
- объект `cin` класса потокового ввода предназначен для предназначен для функционирования системы;
- объект `cout` класса потокового вывода предназначен для предназначен для функционирования системы;
- оператор освобождения памяти `delete`;
- оператор динамического выделения памяти `new`.

Класс `Object`:

- свойства/поля:
  - поле приватное свойство, предназначенное для хранения значения, которое не должно быть доступно извне класса:
    - наименование — `hidden_property`;
    - тип — целочисленный;
    - модификатор доступа — `private`;
  - поле публичное свойство, предназначенное хранение доступного значения:
    - наименование — `available_property`;
    - тип — целочисленный;
    - модификатор доступа — `public`;
- функционал:
  - метод `Object` — конструктор;
  - метод `changeProperties` — изменение значений свойств объекта;
  - метод `callHiddenMethod` — вызов скрытого метода `'hiddenMethod()'`;

- о метод `printState` — вывод текущего состояния объекта, отображая значения свойств `'available_property'` и `'hidden_property'`;
- о метод `hiddenMethod` — увеличение значений свойств `'available_property'` на 5 и `'hidden_property'` на 7.

## 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: конструктор.

Параметры: int value - инициализация свойств объекта класса 'Object'.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *Object*

№	Предикат	Действия	№ перехода
1		присвоение значения параметра 'value' значению свойства 'available_property' объекта класса 'Object'	2
2		установка значения свойства 'hidden_property' объекта класса 'Object' равным утроенному значению параметра 'value'	∅

### 3.2 Алгоритм метода **changeProperties** класса **Object**

Функционал: изменение значений свойств объекта.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *changeProperties* класса *Object*

№	Предикат	Действия	№ перехода
1		значение свойства 'available_property' увеличивается на 4	2
2		значение скрытого свойства 'hidden_property' увеличивается на 1	∅

### 3.3 Алгоритм метода callHiddenMethod класса Object

Функционал: вызов скрытого метода 'hiddenMethod()'.  
Параметры: отсутствуют.  
Возвращаемое значение: отсутствует.  
Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода callHiddenMethod класса Object

№	Предикат	Действия	№ перехода
1		вызов скрытого метода 'hiddenMethod()' объекта класса 'Object'	Ø

### 3.4 Алгоритм метода printState класса Object

Функционал: вывод текущего состояния объекта, отображая значения свойств 'available\_property' и 'hidden\_property'.  
Параметры: отсутствуют.  
Возвращаемое значение: отсутствует.  
Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода printState класса Object

№	Предикат	Действия	№ перехода
1		вывод на экран "Value of the available property ", значение available_property , " Value of a hidden property ", значение hidden_property	Ø

### 3.5 Алгоритм метода hiddenMethod класса Object

Функционал: увеличение значений свойств 'available\_property' на 5 и 'hidden\_property' на 7.  
Параметры: отсутствуют.



Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 5.

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

№	Предикат	Действия	№ перехода
1		значение свойства 'available_property' увеличивается на 7	2
2		значение скрытого свойства 'hidden_property' увеличивается на 5	Ø

### 3.6 Алгоритм функции *main*

Функционал: основной алгоритм работы программы.

Параметры: отсутствуют.

Возвращаемое значение: *int* - индикатор корректности завершения работы программы.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной <i>i_data</i>	2
2		ввод значения переменной <i>i_data</i>	3
3		создание нового объекта типа 'Object' с использованием оператора <i>new</i> , инициализация конструктор класса 'Object' который принимает параметр ' <i>i_data</i> ', возвращение указателя на этот объект	4
4		вызов метода ' <i>printState()</i> ' для объекта, на который указывает указатель ' <i>obj</i> '	5
5		вызов метода ' <i>changeProperties()</i> ' для объекта, на который указывает указатель ' <i>obj</i> '	6
6		вызов метода ' <i>printState()</i> ' для объекта, на который	7

№	Предикат	Действия	№ перехода
		указывает указатель 'obj'	
7		ввод значения переменной i_data	8
8	i_data > obj->available_property	значение свойство 'available_property' объекта, на который указывает указатель 'obj', устанавливается равным значению переменной 'i_data, умноженному на 8	9
			9
9		вызов метода 'printStats()' для объекта, на который указывает указатель 'obj'	10
10		вызов метода 'callHiddenMethod()' для объекта, на который указывает указатель 'obj'	11
11		вызов метода 'printStats()' для объекта, на который указывает указатель 'obj'	12
12		освобождение памяти, выделенная для объекта, на который указывает указатель 'obj' , с помощью оператора 'delete	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

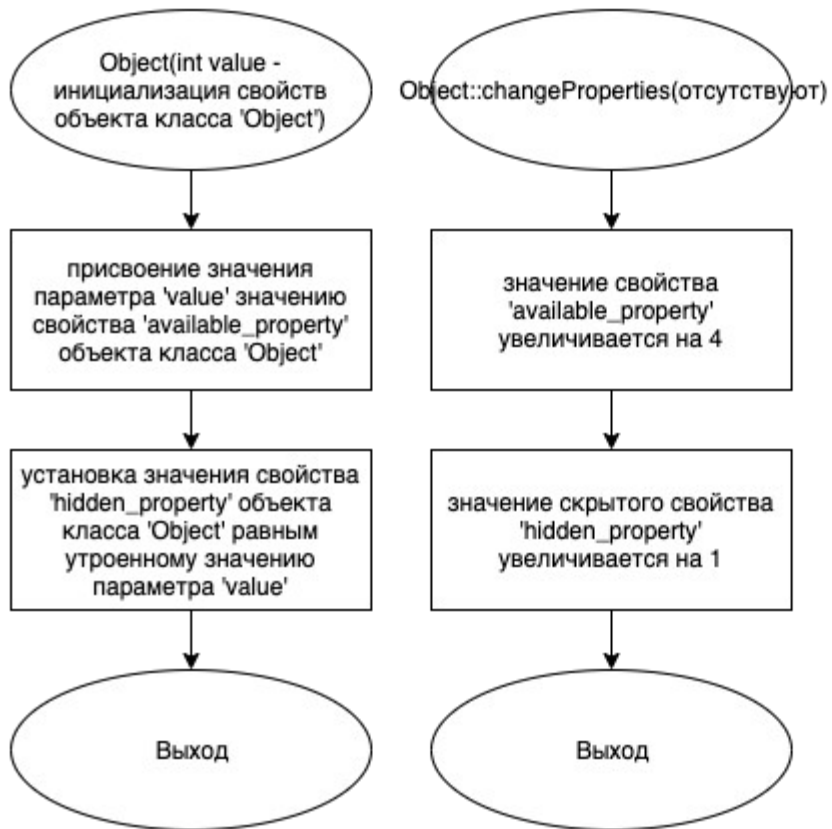
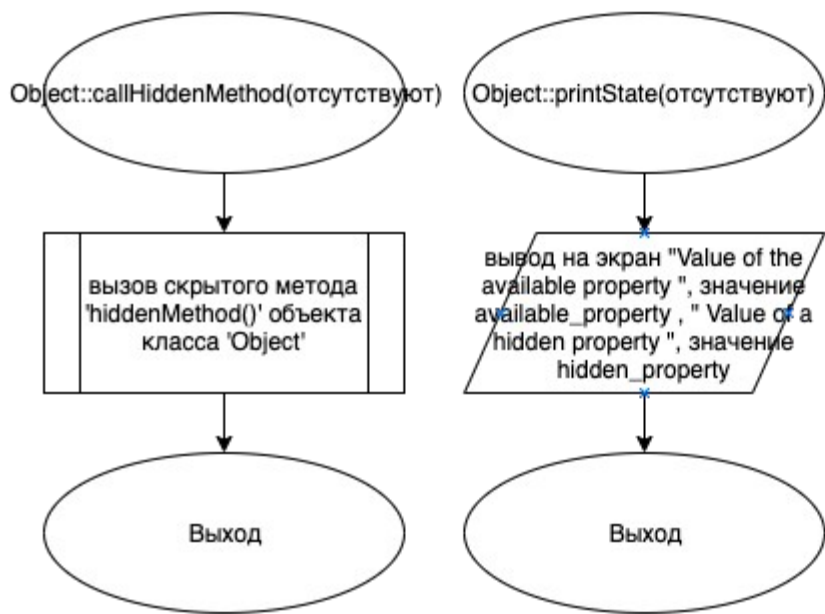


Рисунок 1 – Блок-схема алгоритма



**Рисунок 2 – Блок-схема алгоритма**



**Рисунок 3 – Блок-схема алгоритма**

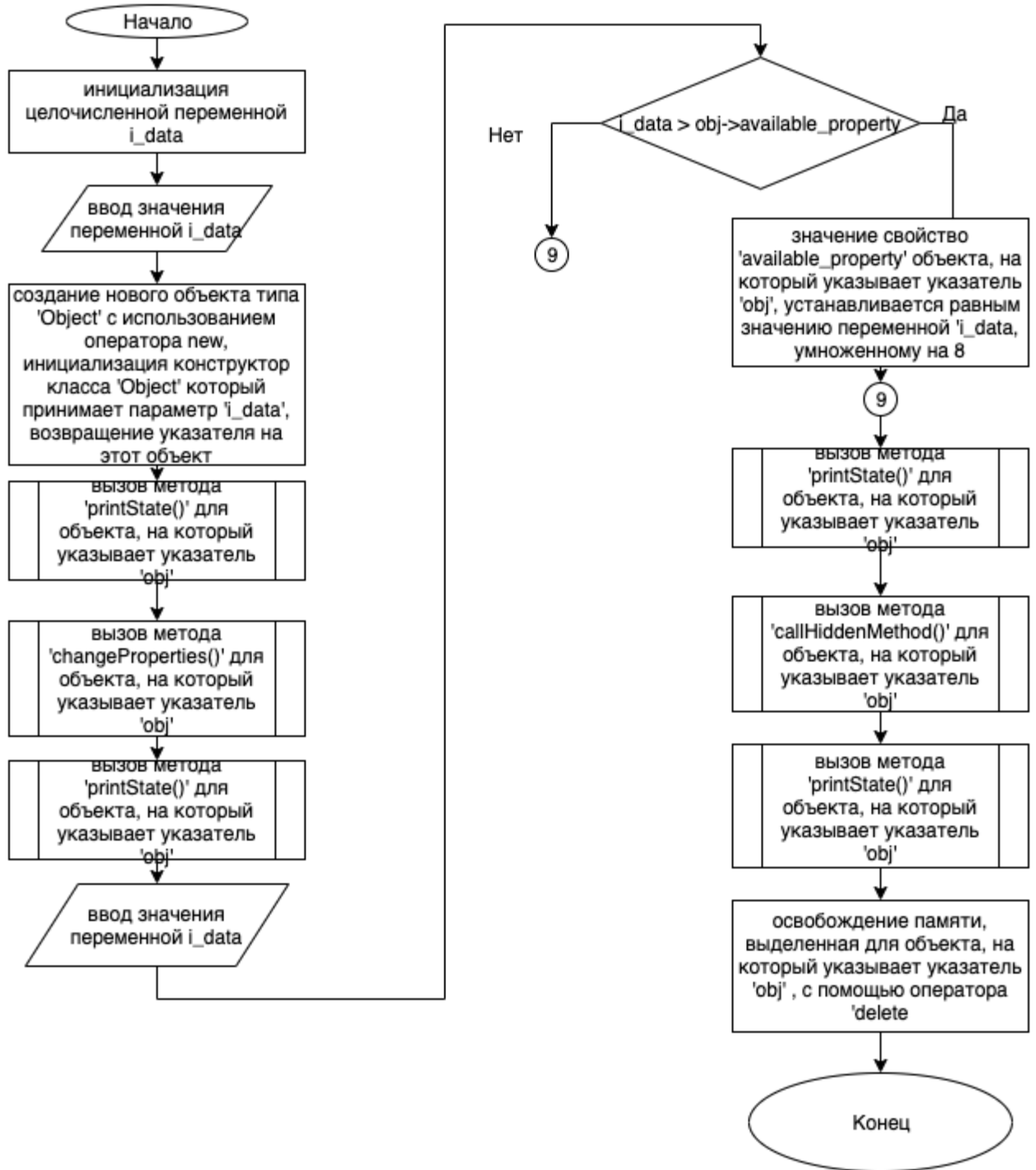


Рисунок 4 – Блок-схема алгоритма

## 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"

using namespace std;

int main() {
    int i_data;
    cin >> i_data;

    Object *obj = new Object(i_data);

    obj->printState();

    obj->changeProperties();

    obj->printState();

    cin >> i_data;

    if(i_data > obj->available_property) {
        obj->available_property = i_data * 8;
    }

    obj->printState();
    obj->callHiddenMethod();

    obj->printState();

    delete obj;

    return(0);
}
```

## 5.2 Файл Object.cpp

Листинг 2 – Object.cpp

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

using namespace std;

Object::Object(int value) {
    available_property = value;
    hidden_property = 3 * value;
};

void Object::changeProperties() {
    available_property += 4;
    hidden_property += 1;
}

void Object::callHiddenMethod() {
    hiddenMethod();
}

void Object::printState() {
    cout << "Value of the available property " << available_property << " ";
    cout << "Value of a hidden property " << hidden_property << endl;
}

//void Object::multiplyAvailableProperty(int value) {
//    available_property *= value;
//}

void Object::hiddenMethod() {
    available_property += 7;
    hidden_property += 5;
}
```

## 5.3 Файл Object.h

Листинг 3 – Object.h

```
#ifndef __OBJECT__H
#define __OBJECT__H

class Object {
private:
    int hidden_property;
};
```



```
        void hiddenMethod();
public:
    int available_property;

    Object(int value);

    void changeProperties();

    void callHiddenMethod();

    void printState();

    //void multiplyAvailableProperty(int value);
};

#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 4	Value of the available property 2; Value of a hidden property 6 Value of the available property 6; Value of a hidden property 7 Value of the available property 6; Value of a hidden property 7 Value of the available property 13; Value of a hidden property 12	Value of the available property 2; Value of a hidden property 6 Value of the available property 6; Value of a hidden property 7 Value of the available property 6; Value of a hidden property 7 Value of the available property 13; Value of a hidden property 12
5 7	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 1	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	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

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	property 4 Value of the available property 12; Value of a hidden property 9	property 4 Value of the available property 12; Value of a hidden property 9
3 5	Value of the available property 3; Value of a hidden property 9 Value of the available property 7; Value of a hidden property 10 Value of the available property 7; Value of a hidden property 10 Value of the available property 14; Value of a hidden property 15	Value of the available property 3; Value of a hidden property 9 Value of the available property 7; Value of a hidden property 10 Value of the available property 7; Value of a hidden property 10 Value of the available property 14; Value of a hidden property 15

## 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).