

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 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 `obj` класса `Object` предназначен для изменение свойств, вызов методов, вывод состояния объектов;
- объект `cin` класса потокового ввода предназначен для функционирование системы;
- объект `cout` класса потокового вывода предназначен для функционирование системы.

Класс `Object`:

- свойства/поля:
 - поле хранение доступного значения объекта:
 - наименование — `available_property`;
 - тип — целочисленный;
 - модификатор доступа — `public`;
 - поле хранение значение объекта, которое не должно быть доступно напрямую:
 - наименование — `hidden_property`;
 - тип — целочисленный;
 - модификатор доступа — `private`;
- функционал:
 - метод `Object` — конструктор, инициализирует объект класса `'Object'` с переданным значением;
 - метод `changeProperties` — изменение значений доступного и скрытого свойств объекта;
 - метод `callHiddenMethod` — вызов скрытого метода объекта;
 - метод `printState` — вывод текущего состояния объекта, включая

значения его доступного и скрытого свойств;

- о метод `multiplyAvailableProperty` — изменение доступного свойства объекта путем умножения его значения на заданное значение;

- о метод `hiddenMethod` — выполнение скрытых операций над свойствами объекта, таких как изменения доступного и скрытого свойств.

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: создание объекта класса 'Object'.

Параметры: int value - для инициализации переменной 'available_property'.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		объявление конструктора класса 'Object', который принимает один аргумент типа 'int'	2
2		присвоение значению свойства 'available_property' объекта класса 'Object' значение, переданное в качестве аргумента конструктору	3
3		установка значения скрытого свойства 'hidden_property' объекта класса 'Object', устанавливается в 2 раза больше переданного значения в конструктор	∅

3.2 Алгоритм метода changeProperties класса Object

Функционал: изменение значений доступного и скрытого свойств объекта.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 2.

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

№	Предикат	Действия	№ перехода
1		определение метода 'changeProperties()' класса 'Object'	2
2		увеличение значение свойства 'available_property' на 1	3
3		увеличение значение скрытого свойства 'hidden_property' на 4	Ø

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		определение метода <i>callHiddenMethod()</i> класса 'Object'	2
2		вызов скрытого метода <i>hiddenMethod()</i>	Ø

3.4 Алгоритм метода *printState* класса *Object*

Функционал: вывод текущего состояния объекта.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *printState* класса *Object*

№	Предикат	Действия	№ перехода
1		определение метода <i>printState()</i> класса 'Object'	2
2		вывод на экран "Value of the available property ", значение поля <i>available_property</i> , "; Value of a hidden property", значение поля	Ø

№	Предикат	Действия	№ перехода
		hidden_property	

3.5 Алгоритм метода multiplyAvailableProperty класса Object

Функционал: изменение доступного свойства объекта.

Параметры: int value.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода multiplyAvailableProperty класса Object

№	Предикат	Действия	№ перехода
1		определение метода 'multiplyAvailableProperty()' класса 'Object', принимает параметр value	2
2		значение доступного свойства 'available_property' умножается на значение параметра 'value', результат присваивается обратно свойству 'available_property'	∅

3.6 Алгоритм метода hiddenMethod класса Object

Функционал: выполнение скрытых операций над свойствами объекта.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода hiddenMethod класса Object

№	Предикат	Действия	№ перехода
1		определение метода 'hiddenMethod()' класса 'Object'	2
2		увеличение значение свойства 'available_property' на 5	3

№	Предикат	Действия	№ перехода
3		увеличение значение скрытого свойства 'hidden_property' на 7	Ø

3.7 Алгоритм функции main

Функционал: основной алгоритм работы программы.

Параметры: отсутствуют.

Возвращаемое значение: int - индикатор корректности завершения работы программы.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной i_data	2
2		ввод значения переменной i_data	3
3		создание объекта 'obj' класса 'Object', который инициализируется значением 'i_data'	4
4		вызов метода 'printState()' объекта 'obj', который выводит текущее состояние объекта	5
5		вызов метода 'changeProperties()' объекта 'obj', который изменяет значение доступного и скрытого свойств объекта	6
6		вызов метода 'printState()' объекта 'obj', который выводит текущее состояние объекта после изменения его свойств методом changeProperties()	7
7		ввод значения переменной i_data	8
8		вызов метода "multiplyAvailableProperty()" объекта 'obj', который изменяет значение доступного свойства объекта, умножая его значение на 'i_data'	9
9		вызов метода 'printState()' объекта 'obj', чтобы вывести текущее состояни объекта после изменения его доступного свойства методом	10

№	Предикат	Действия	№ перехода
		'multiplyAvailableProperty()'	
10		вызов метода 'callHiddenMethod()' объекта 'obj', который вызывает скрытый метод 'hiddenMethod()'	11
11		вызов метода вызов метода 'printState()' объекта 'obj', который выводит текущее состояние объекта после выполнения скрытых операция методом 'callHiddenMethod()'	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

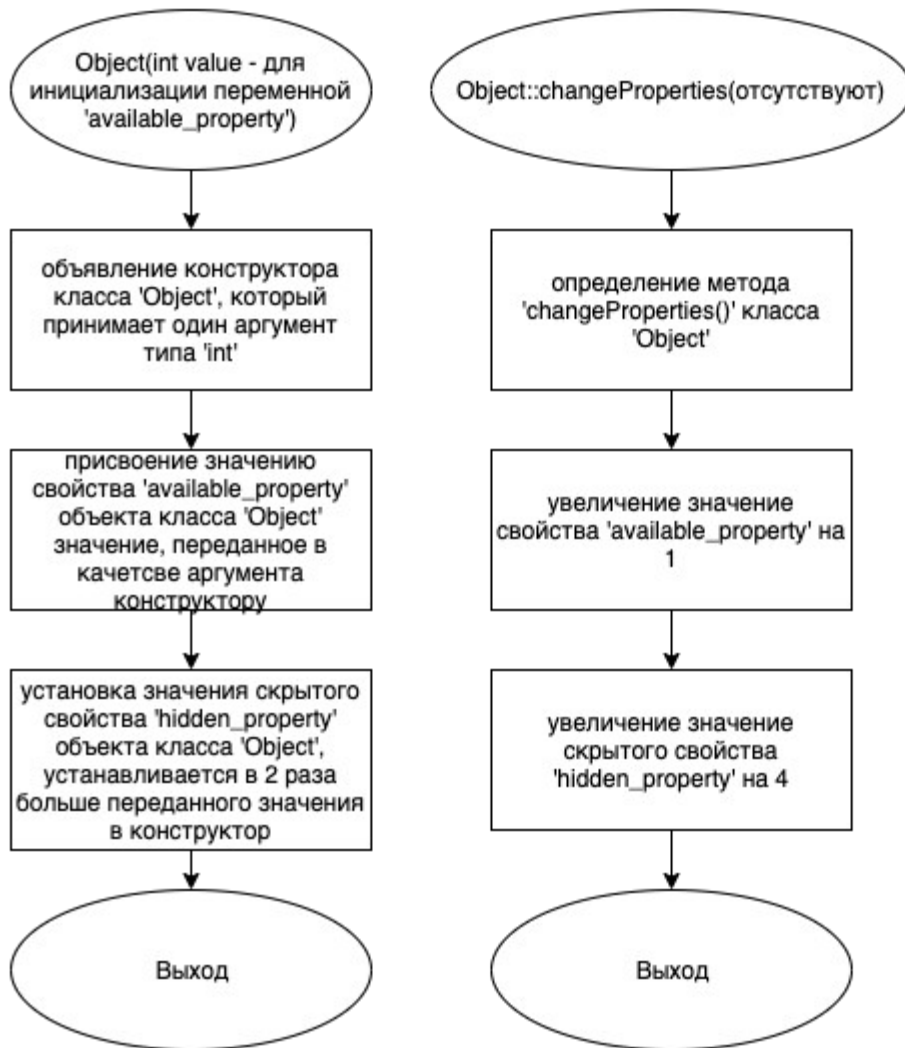


Рисунок 1 – Блок-схема алгоритма

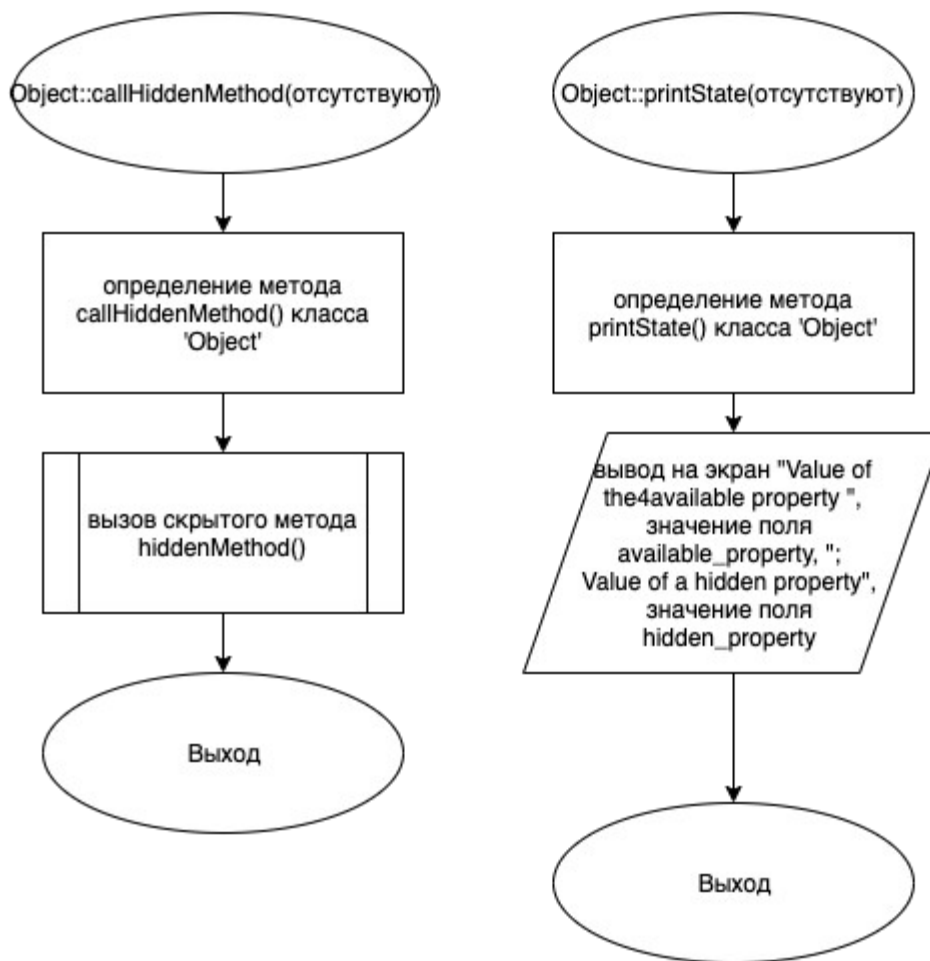


Рисунок 2 – Блок-схема алгоритма

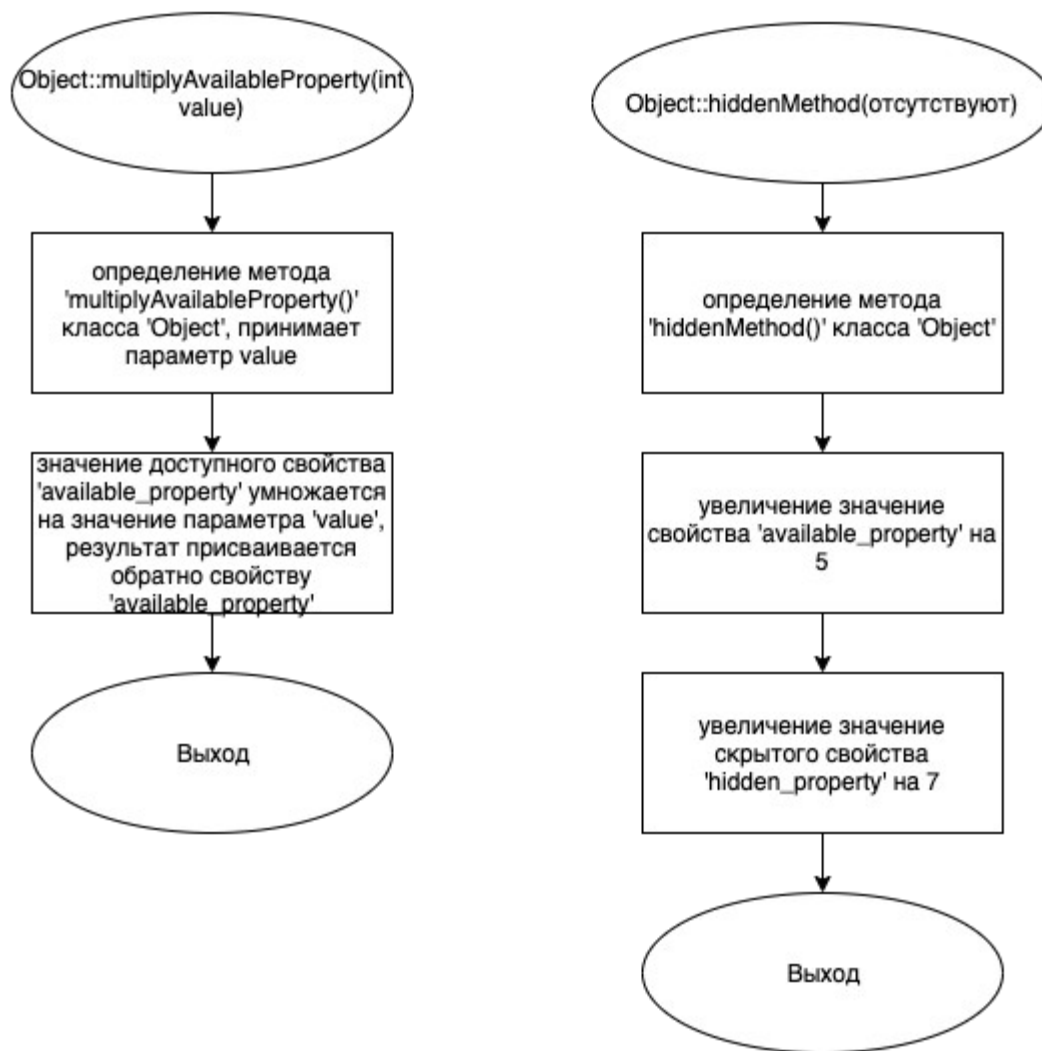


Рисунок 3 – Блок-схема алгоритма

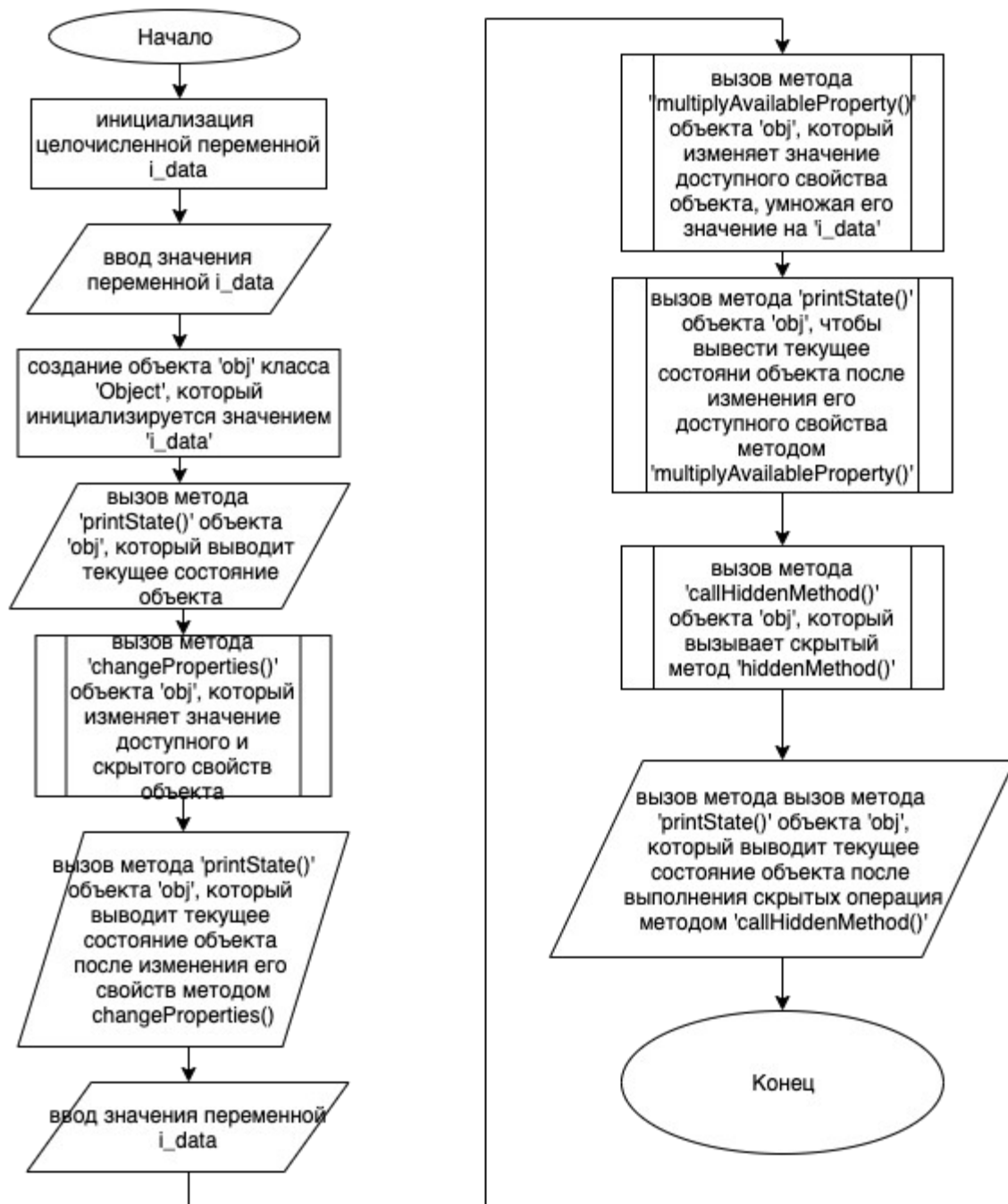


Рисунок 4 – Блок-схема алгоритма

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(i_data);

    obj.printState();

    obj.changeProperties();

    obj.printState();

    cin >> i_data;

    obj.multiplyAvailableProperty(i_data);

    obj.printState();

    obj.callHiddenMethod();

    obj.printState();

    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 = 2 * value;
};

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

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 += 5;
    hidden_property += 7;
}
```

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 8.

Таблица 8 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 10	Value of the available property 5; Value of a hidden property 10 Value of the available property 6; Value of a hidden property 14 Value of the available property 60; Value of a hidden property 14 Value of the available property 65; Value of a hidden property 21	Value of the available property 5; Value of a hidden property 10 Value of the available property 6; Value of a hidden property 14 Value of the available property 60; Value of a hidden property 14 Value of the available property 65; 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 (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 (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).