

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 PrivateCall класса Object.....	9
3.3 Алгоритм функции main.....	10
3.4 Алгоритм метода Plus класса Object.....	10
3.5 Алгоритм метода PrintProperty класса Object.....	11
3.6 Алгоритм метода ChangeProperty класса Object.....	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 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 `Object` класса `Object` предназначен для Объект, в котором есть доступные и скрытые элементы;
- функция `main` для главная функция программы;
- библиотека `iostream`;
- объект `cin` стандартного потока ввода с клавиатуры;
- объект `cout` стандартного потока вывода на экран;
- пространство имён `std`.

Класс `Object`:

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

о метод `ChangeProperty` — Изменение значений доступного и приватного 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: параметризированный конструктор для создания объекта класса и изменения значений полей с помощью параметра.

Параметры: int dataValue, использующийся для изменения полей класса.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Присвоение значения параметра dataValue полю dataPublic	2
2		Присвоение удвоенного значения параметра dataValue полю dataPrivate	Ø

3.2 Алгоритм метода PrivateCall класса Object

Функционал: Вызов приватного метода ChangePropsPrivate.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода PrivateCall класса Object

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

3.3 Алгоритм функции main

Функционал: Главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной i_data	2
2		Ввод значения переменной i_data	3
3		Создание объекта Object класса Object путём вызова конструктора с передачей значения i_data в параметр dataValue	4
4		Вызов метода PrintProperty объекта Object	5
5		Вызов метода ChangeProperty объекта Object	6
6		Вызов метода PrintProperty объекта Object	7
7		Ввод значения переменной i_data	8
8		Присвоение полю dataPublic щбъекта obj значения поля dataPublic, умноженного на значение переменной i_data	9
9		Вызов метода PrintProperty объекта Object	10
10		Вызов метода PrivateCall объекта Object	11
11		Вызов метода PrintProperty объекта Object	Ø

3.4 Алгоритм метода Plus класса Object

Функционал: Изменение значений доступного и скрытого поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 4.

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

№	Предикат	Действия	№ перехода
1		Увеличить значение переменной <i>dataPublic</i> на 1	2
2		Увеличить значение переменной <i>dataPrivate</i> на 4	Ø

3.5 Алгоритм метода *PrintProperty* класса *Object*

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

Параметры: *int x*.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 5.

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

№	Предикат	Действия	№ перехода
1		Вывод: "Value of the available property "	2
2		Вывод: <<значение <i>dataPublic</i> >>	3
3		Вывод: "; Value of a hidden property "	4
4		Вывод: <<значение <i>dataPrivate</i> >>	5
5	Значение переменной <i>x</i> равно 0	Переход на новую строку	Ø
	Значение переменной <i>x</i> не равно 0		Ø

3.6 Алгоритм метода *ChangeProperty* класса *Object*

Функционал: Изменение значений доступного и приватного поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода *ChangeProperty* класса *Object*

№	Предикат	Действия	№ перехода
1		Увеличить значение переменной dataPublic на 5	2
2		Увеличить значение переменной dataPrivate на 7	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

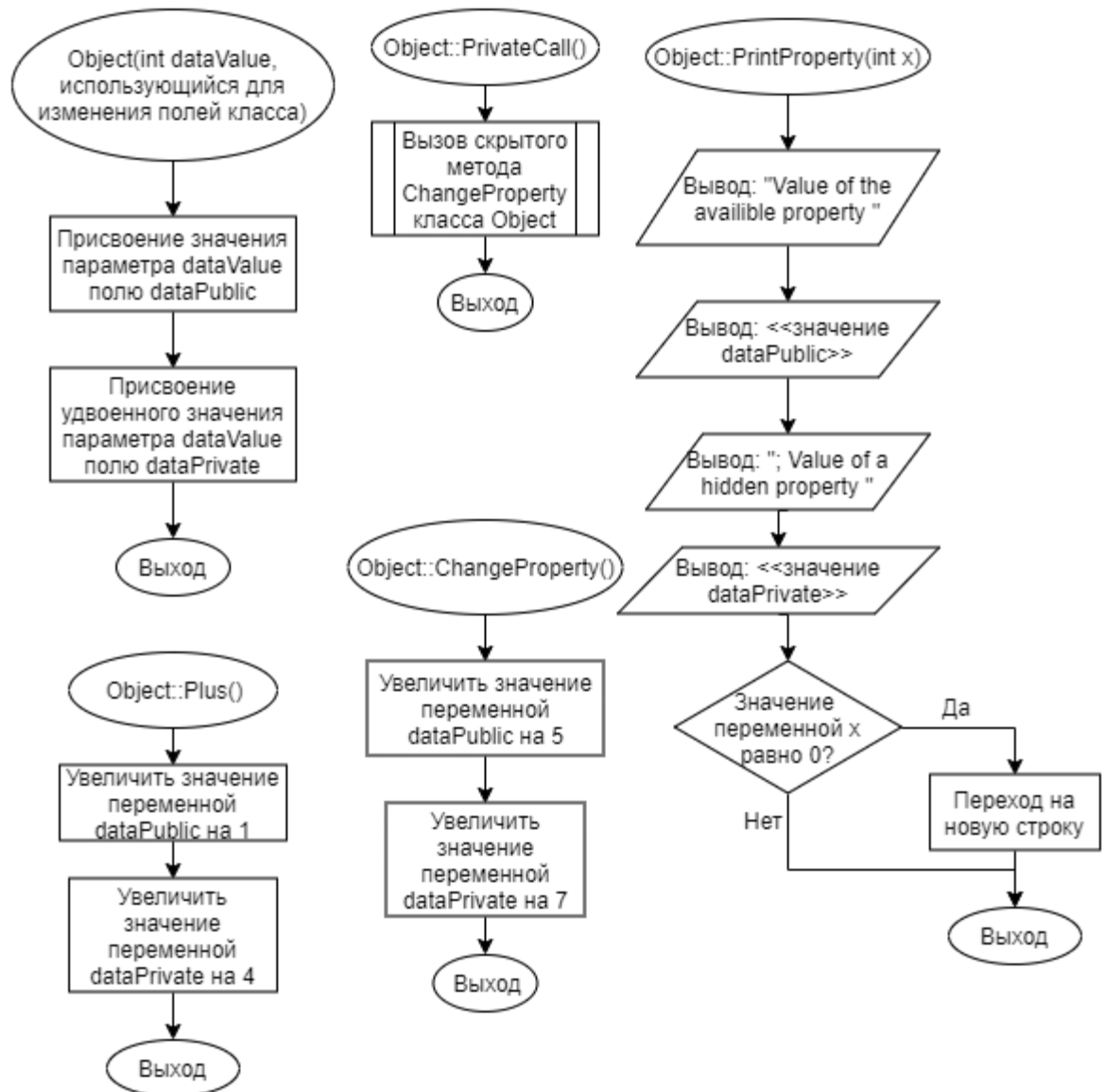


Рисунок 1 – Блок-схема алгоритма

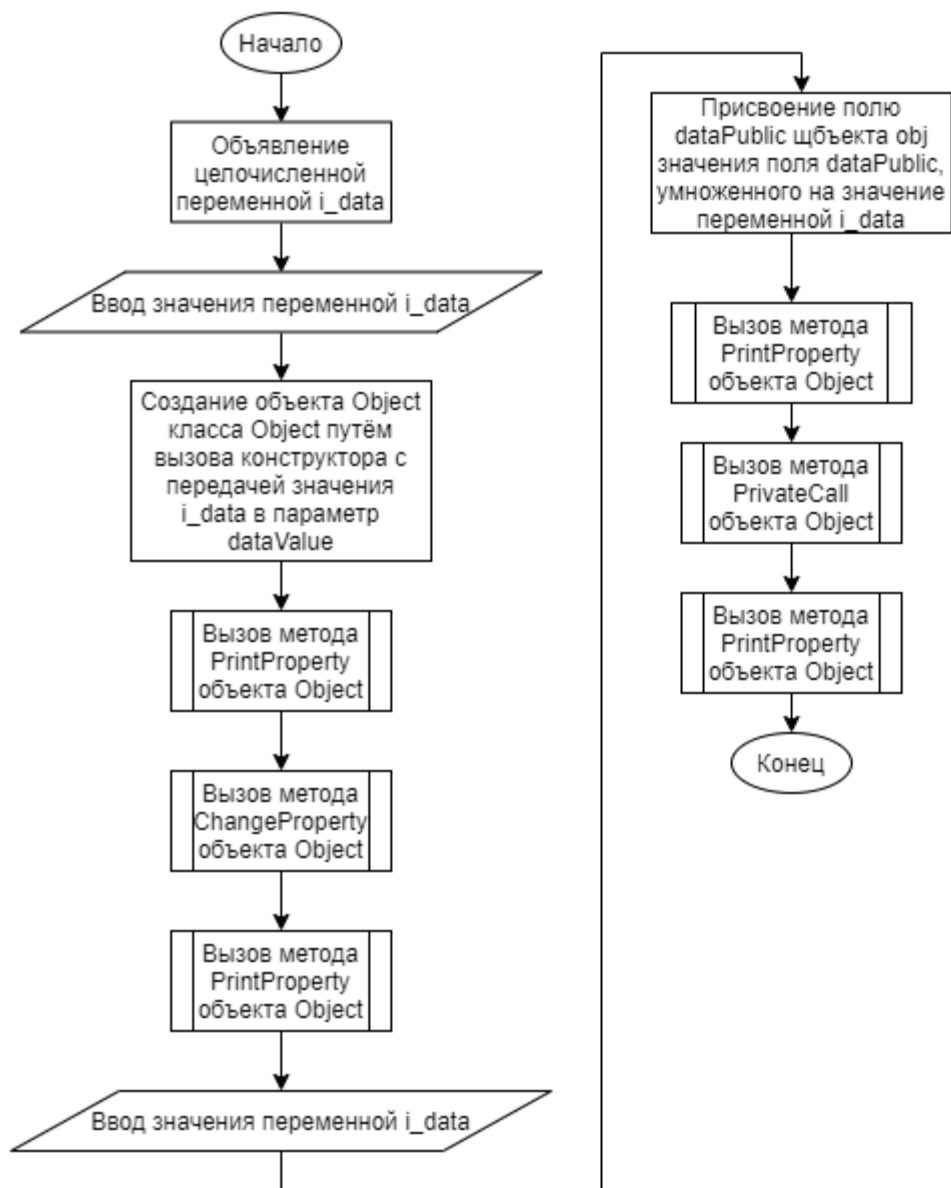


Рисунок 2 – Блок-схема алгоритма

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 Object(i_data);
    Object.PrintProperty(0);
    Object.Plus();
    Object.PrintProperty(0);
    cin >> i_data;
    Object.dataPublic *= i_data;
    Object.PrintProperty(0);
    Object.PrivateCall();
    Object.PrintProperty(0);
    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*2;
};
void Object::Plus()
{
    dataPublic ++;
```

```

        dataPrivate +=4 ;
    }
    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 +=5;
        dataPrivate +=7;
    }

```

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
7 8	Value of the available property 7; Value of a hidden property 14 Value of the available property 8; Value of a hidden property 18 Value of the available property 64; Value of a hidden property 18 Value of the available property 69; Value of a hidden property 25	Value of the available property 7; Value of a hidden property 14 Value of the available property 8; Value of a hidden property 18 Value of the available property 64; Value of a hidden property 18 Value of the available property 69; Value of a hidden property 25
8 1	Value of the available property 8; Value of a hidden property 16 Value of the available property 9; Value of a hidden property 20 Value of the available property 9; Value of a hidden property 20 Value of the available property 14; Value of a hidden property 27	Value of the available property 8; Value of a hidden property 16 Value of the available property 9; Value of a hidden property 20 Value of the available property 9; Value of a hidden property 20 Value of the available property 14; Value of a hidden property 27
2024 1036	Value of the available property 2024; Value of a hidden property 4048 Value of the available property 2025; Value of a hidden property 4052 Value of the available property 2097900; Value of a	Value of the available property 2024; Value of a hidden property 4048 Value of the available property 2025; Value of a hidden property 4052 Value of the available property 2097900; Value of a

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	hidden property 4052 Value of the available property 2097905; Value of a hidden property 4059	hidden property 4052 Value of the available property 2097905; Value of a hidden property 4059
7 7	Value of the available property 7; Value of a hidden property 14 Value of the available property 8; Value of a hidden property 18 Value of the available property 56; Value of a hidden property 18 Value of the available property 61; Value of a hidden property 25	Value of the available property 7; Value of a hidden property 14 Value of the available property 8; Value of a hidden property 18 Value of the available property 56; Value of a hidden property 18 Value of the available property 61; Value of a hidden property 25

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).