

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.....	8
3.1 Алгоритм функции main.....	8
3.2 Алгоритм метода first класса MyClass.....	8
3.3 Алгоритм метода second класса MyClass.....	9
3.4 Алгоритм метода hidden класса MyClass.....	9
3.5 Алгоритм метода res класса MyClass.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл MyClass.cpp.....	14
5.3 Файл MyClass.h.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

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 стандартного ввода `cin`;
- Объект стандартного вывода `cout`.

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 функции main

Функционал: основная программа.

Параметры: нет.

Возвращаемое значение: целое, код успеха.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление i_data	2
2		Ввод i_data	3
3		Объявление obj класса MyClass	4
4		Вызов метода res	5
5		Вызов метода first	6
6		Вызов метода res	7
7		Ввод i_data	8
8		Изменение значения parametr_private вследствие умножение его на i_data	9
9		Вызов метода res	10
10		Вызов метода second	11
11		Вызов метода res	Ø

3.2 Алгоритм метода first класса MyClass

Функционал: Изменяет значение parametr_private.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода first класса MyClass

№	Предикат	Действия	№ перехода
1		parametr_public += 1	2
2		parametr_private += 4	Ø

3.3 Алгоритм метода second класса MyClass

Функционал: Вызывает скрытый метод hidden.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода second класса MyClass

№	Предикат	Действия	№ перехода
1		Вызов скрытого метода hidden	Ø

3.4 Алгоритм метода hidden класса MyClass

Функционал: Изменяет значение parametr_private и parametr_public.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *hidden* класса *MyClass*

№	Предикат	Действия	№ перехода
1		parametr_public += 5	2
2		parametr_private +=7	Ø

3.5 Алгоритм метода *res* класса *MyClass*

Функционал: Вывод parametr_public и parametr_private.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *res* класса *MyClass*

№	Предикат	Действия	№ перехода
1		Вывод текста Value of the available property и значения parametr_public и текста Value of the hidden и значения parametr_private	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

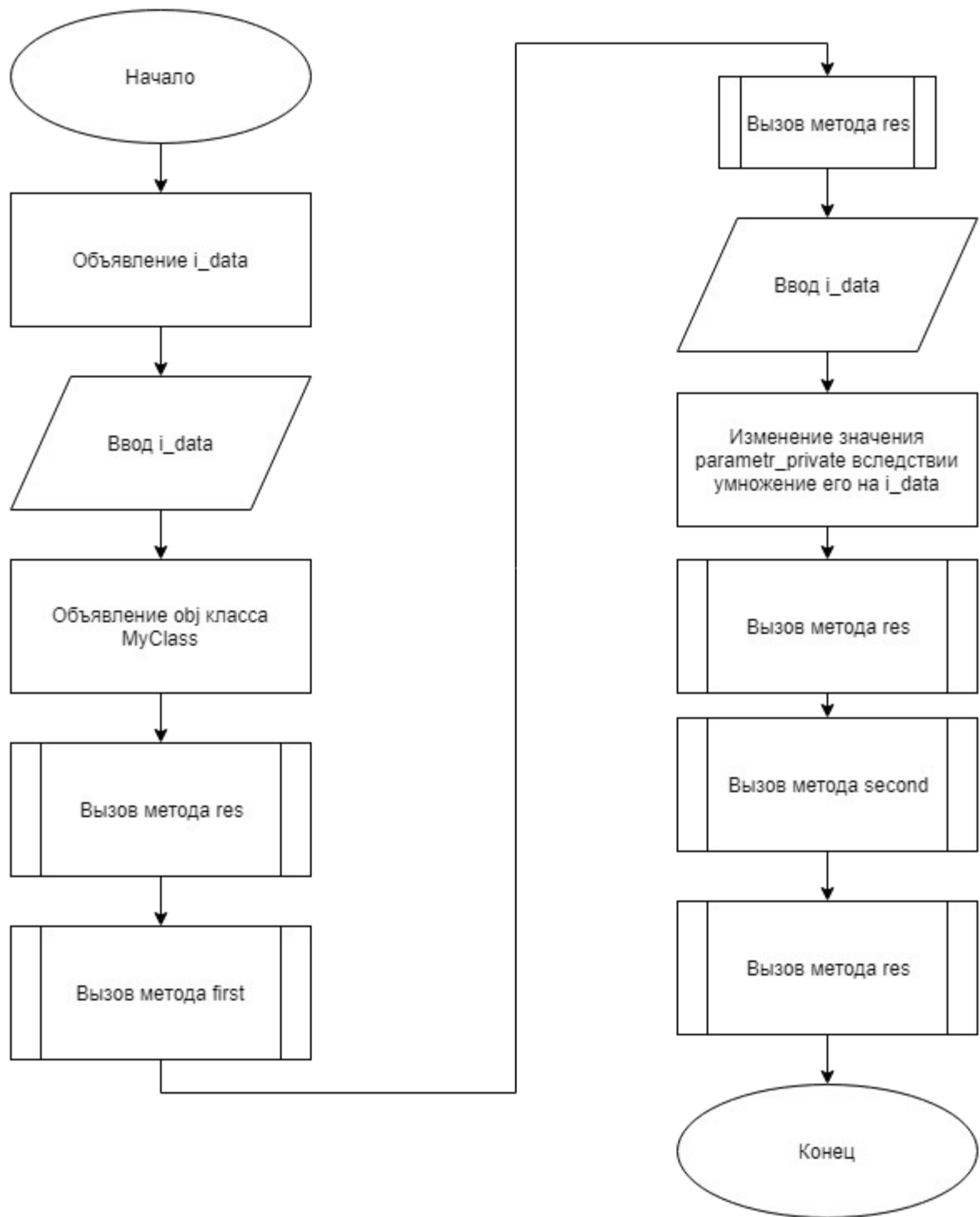


Рисунок 1 – Блок-схема алгоритма

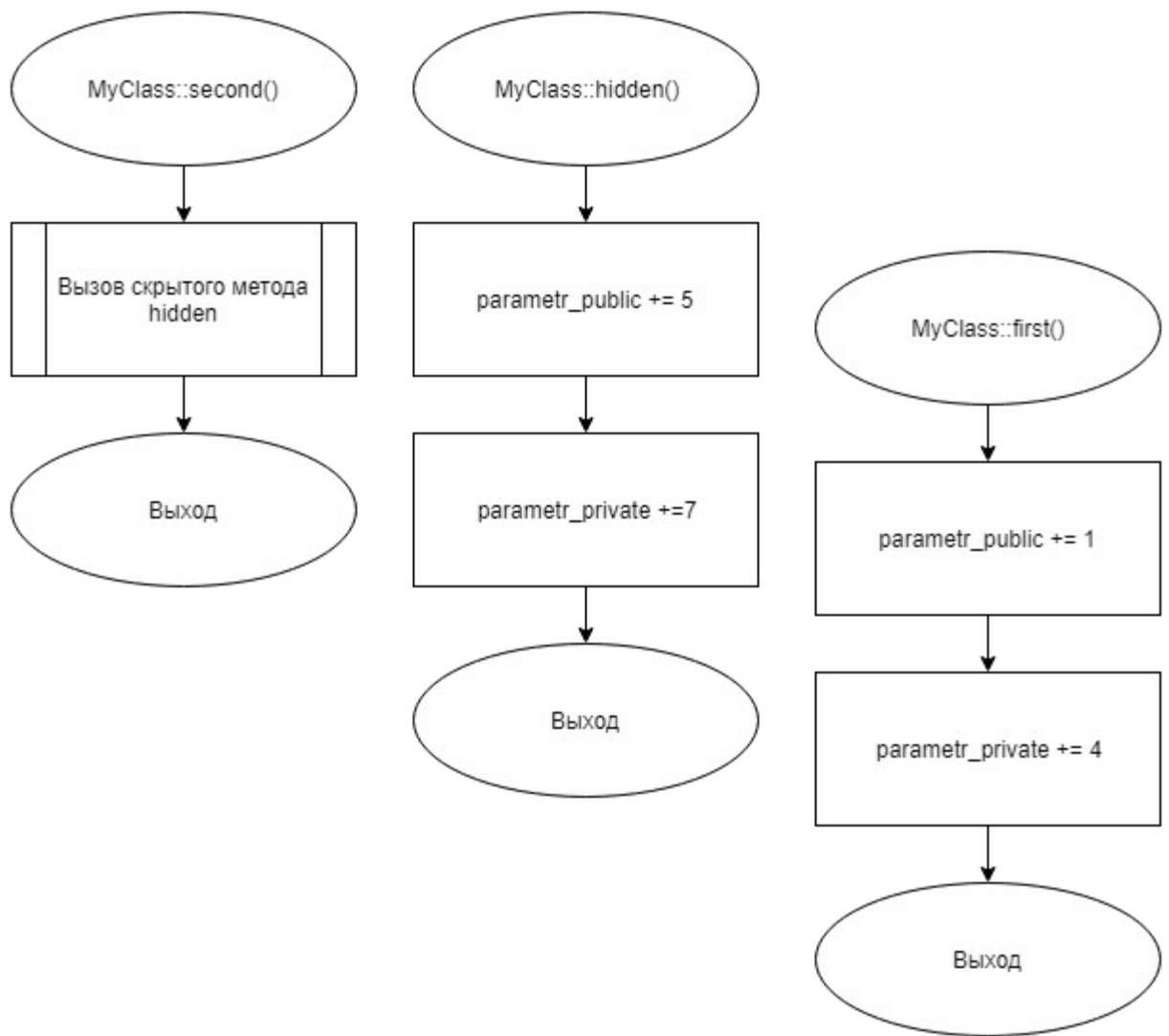


Рисунок 2 – Блок-схема алгоритма

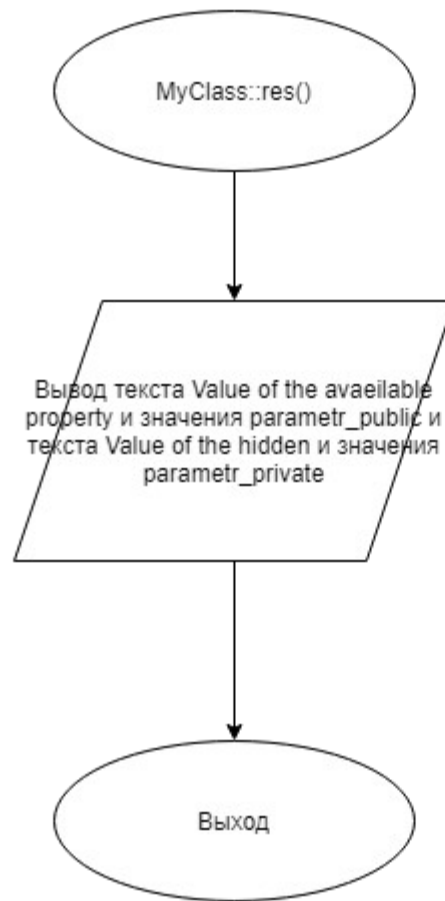


Рисунок 3 – Блок-схема алгоритма

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 "MyClass.h"
using namespace std;
int main()
{
    int i_data;
    cin >> i_data;
    MyClass obj(i_data);
    obj.res();
    cout << "\n";
    obj.first();
    obj.res();
    cout << "\n";
    cin >> i_data;
    obj.parametr_public *= i_data;
    obj.res();
    cout << "\n";
    obj.second();
    obj.res();
    return(0);
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

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

MyClass::MyClass(int i_data)
{
```

```

    parametr_public = i_data;
    parametr_private = i_data*2;
}

void MyClass::first()
{
    parametr_public += 1;
    parametr_private += 4;
}
void MyClass::second()
{
    hidden();
}
void MyClass::res()
{
    cout << "Value of the available property " << parametr_public << "; " <<
    "Value of a hidden property " << parametr_private;
}
void MyClass::hidden()
{
    parametr_public += 5;
    parametr_private += 7;
}

```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```

#ifndef __MYCLASS__H
#define __MYCLASS__H
class MyClass
{
private:
    int parametr_private;
    void hidden();
public:
    MyClass(int i_data);
    void first();
    void second();
    void res();
    int parametr_public;
};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 3	Value of the available property 2; Value of a hidden property 4 Value of the available property 3; Value of a hidden property 8 Value of the available property 9; Value of a hidden property 8 Value of the available property 14; Value of a hidden property 15	Value of the available property 2; Value of a hidden property 4 Value of the available property 3; Value of a hidden property 8 Value of the available property 9; Value of a hidden property 8 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 (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).