

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 конструктора класса Test..... | 8 |
| 3.2 Алгоритм метода change_open_close класса Test..... | 8 |
| 3.3 Алгоритм метода output класса Test..... | 9 |
| 3.4 Алгоритм метода call_private класса Test..... | 9 |
| 3.5 Алгоритм метода hide класса Test..... | 9 |
| 3.6 Алгоритм функции main..... | 10 |
| 4 БЛОК-СХЕМЫ АЛГОРИТМОВ..... | 11 |
| 5 КОД ПРОГРАММЫ..... | 13 |
| 5.1 Файл main.cpp..... | 13 |
| 5.2 Файл Test.cpp..... | 13 |
| 5.3 Файл Test.h..... | 14 |
| 6 ТЕСТИРОВАНИЕ..... | 16 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 18 |

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 Test;
- Объект стандартного потока ввода с клавиатуры cin;
- Объект стандартного потока вывода на экран cout;
- Условный оператор if..else.

Класс Test:

- свойства/поля:
 - поле Доступное свойство:
 - наименование — open;
 - тип — Целое;
 - модификатор доступа — public;
 - поле Скрытое свойство:
 - наименование — close;
 - тип — Целое;
 - модификатор доступа — private;
- функционал:
 - метод Test — Конструктор;
 - метод change_open_close — Метод изменения значения доступного и скрытого свойства;
 - метод output — Метод вывода состояния, выводит значение доступного и скрытого свойства;
 - метод call_private — Метод вызова закрытого метода;
 - метод hide — Метод изменения значения доступного и скрытого свойства.

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 Test

Функционал: Конструктор.

Параметры: Переменная целого типа `i_data` для инициализации переменных.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Test

| № | Предикат | Действия | № перехода |
|---|----------|---|---------------|
| 1 | | Открыта переменная <code>open</code> инициализируется параметром <code>i_data</code> | 2 |
| 2 | | Скрытая переменная <code>close</code> инициализируется утроенным параметром <code>i_data</code> | ∅ |

3.2 Алгоритм метода `change_open_close` класса Test

Функционал: Метод изменения значения доступного и скрытого свойства.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `change_open_close` класса Test

| № | Предикат | Действия | № перехода |
|---|----------|---|---------------|
| 1 | | Увеличение открытой переменной <code>open</code> на 4 | 2 |
| 2 | | Увеличение скрытой переменной <code>close</code> на 1 | ∅ |

3.3 Алгоритм метода output класса Test

Функционал: Метод вывода состояния, выводит значение доступного и скрытого свойства.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода output класса Test

| № | Предикат | Действия | № перехода |
|---|----------|---|---------------|
| 1 | | Вывод на экран "Value of the available property (значение доступного свойства); Value of a hidden property (значение закрытого свойства)/n" | Ø |

3.4 Алгоритм метода call_private класса Test

Функционал: Метод вызова закрытого метода.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода call_private класса Test

| № | Предикат | Действия | № перехода |
|---|----------|----------------------------|---------------|
| 1 | | Вызов скрытого метода hide | Ø |

3.5 Алгоритм метода hide класса Test

Функционал: Метод изменения значения доступного и скрытого свойства.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *hide* класса *Test*

| № | Предикат | Действия | № перехода |
|---|----------|---|---------------|
| 1 | | Увеличение открытой переменной <i>open</i> на 7 | 2 |
| 2 | | Увеличение скрытой переменной <i>close</i> на 5 | ∅ |

3.6 Алгоритм функции *main*

Функционал: Выполнение действий, поставленных в задаче.

Параметры: нет.

Возвращаемое значение: Целочисленное значение.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

| № | Предикат | Действия | № перехода |
|----|-------------------------------|---|---------------|
| 1 | | Объявление переменной целого типа <i>i_data</i> | 2 |
| 2 | | Ввод с клавиатуры значения переменной <i>i_data</i> | 3 |
| 3 | | Создание указателя <i>a</i> на объект класса <i>Test</i> с параметром <i>i_data</i> | 4 |
| 4 | | Вызов метода <i>output</i> объекта с адресом <i>a</i> | 5 |
| 5 | | Вызов метода <i>change_open_close</i> объекта с адресом <i>a</i> | 6 |
| 6 | | Вызов метода <i>output</i> объекта с адресом <i>a</i> | 7 |
| 7 | | Ввод с клавиатуры значения переменной <i>i_data</i> | 8 |
| 8 | <i>i_data > a->open</i> | Открытой переменной <i>open</i> объекта с адресом <i>a</i> присваивается значение <i>i_data * 8</i> | 9 |
| | | | 9 |
| 9 | | Вызов метода <i>output</i> объекта с адресом <i>a</i> | 10 |
| 10 | | Вызов метода <i>call_private</i> объекта с адресом <i>a</i> | 11 |
| 11 | | Вызов метода <i>output</i> объекта с адресом <i>a</i> | 12 |
| 12 | | Очистка памяти по адресу <i>a</i> | ∅ |

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

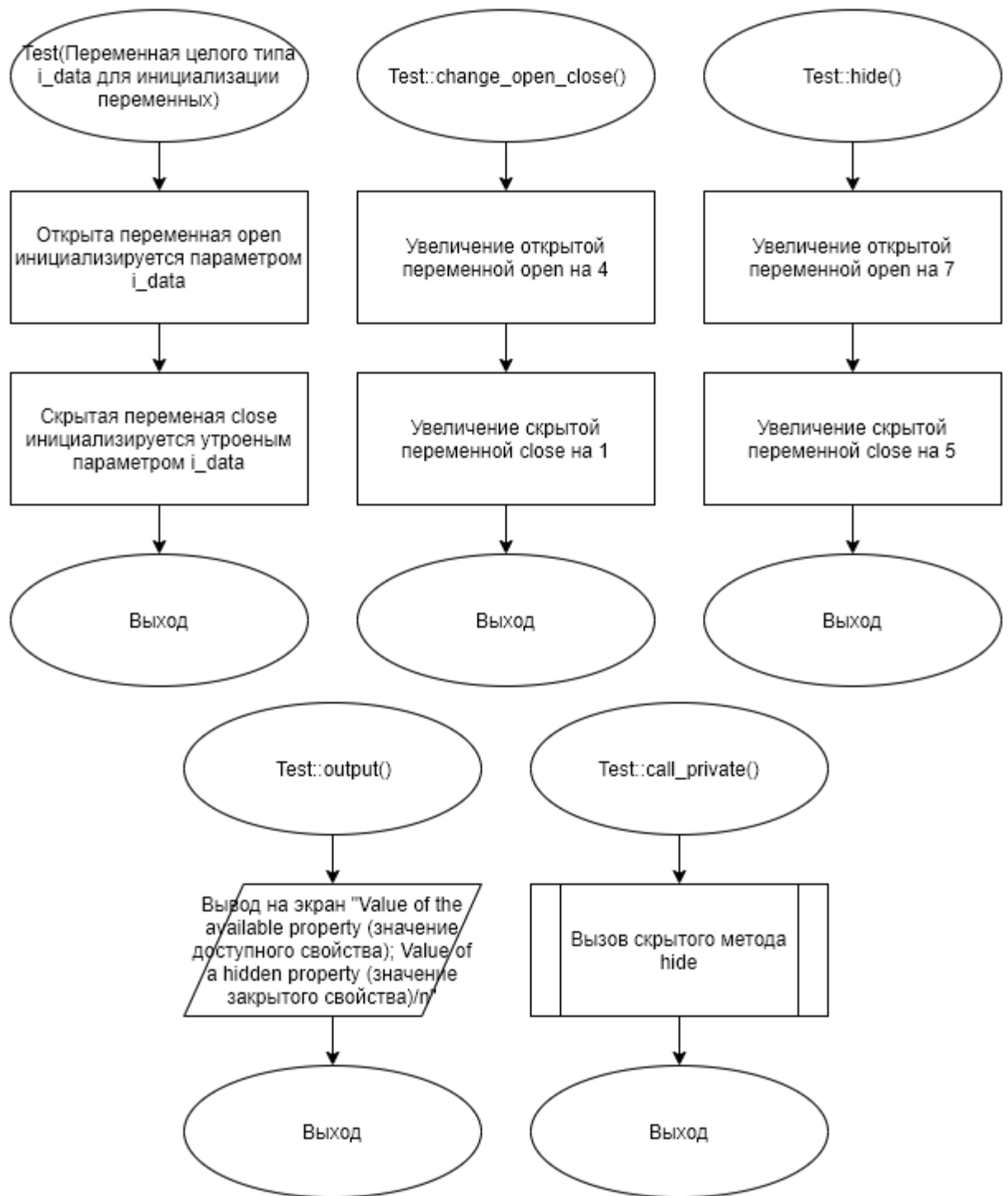


Рисунок 1 – Блок-схема алгоритма

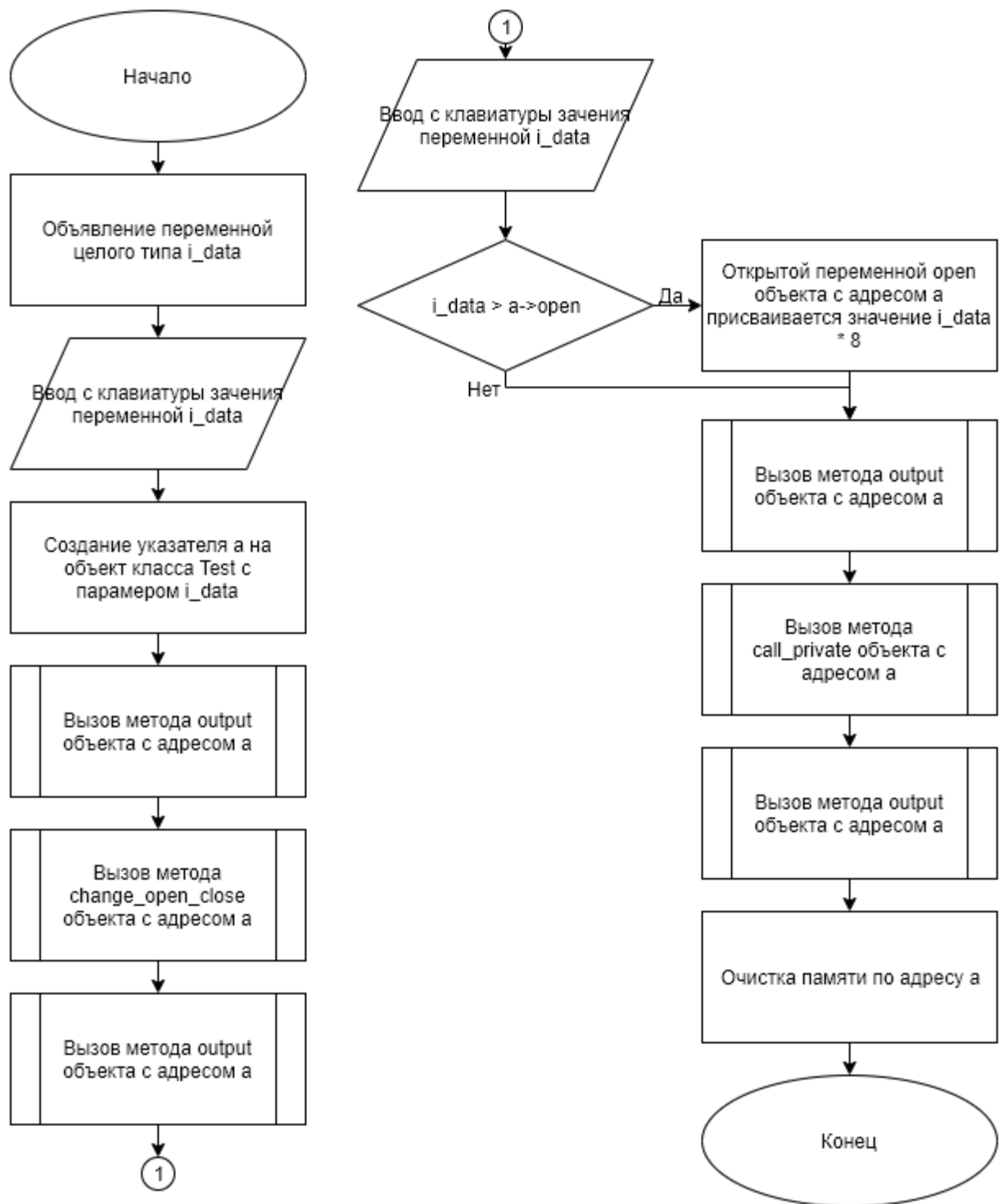


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "Test.h"

using namespace std;

int main()
{
    int i_data;
    cin >> i_data;
    Test *a = new Test(i_data);
    a->output();
    a->change_open_close();
    a->output();
    cin >> i_data;
    if (i_data > a->open) {
        a->open = i_data * 8;
    }
    a->output();
    a->call_private();
    a->output();
    delete[] a;
    return(0);
}
```

5.2 Файл Test.cpp

Листинг 2 – Test.cpp

```
#include "Test.h"

Test::Test(int i_data)
{
```

```

        open = i_data;
        close = i_data * 3;
    }

    void Test::change_open_close()
    {
        open += 4;
        close++;
    }

    void Test::call_private()
    {
        hide();
    }

    void Test::output()
    {
        cout << "Value of the available property " << open << "; Value of a hidden
property " << close << endl;
    }

    void Test::hide()
    {
        open += 7;
        close += 5;
    }

```

5.3 Файл Test.h

Листинг 3 – Test.h

```

#ifndef __TEST__H
#define __TEST__H
#include <iostream>

using namespace std;

class Test
{
private:
    int close;
    void hide();

public:
    int open;
    Test(int i_data);
    void change_open_close();
    void call_private();
    void output();

```

```
};
```

```
#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
|----------------|--|--|
| 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 |
| 1 7 | 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 56; Value of a hidden property 4 Value of the available property 63; 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 56; Value of a hidden property 4 Value of the available property 63; Value of a hidden property 9 |
| 3 7 | 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 | 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 |

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
|----------------|---|---|
| | property 10 Value of the available property 14; Value of a hidden property 15 | 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 (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).