

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 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:

- `if..else` - условный оператор;
- `cin/cout` - объекты потокового ввода/вывода;
- `new/delete` - операторы выделения/освобождения динамической памяти.

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 функция программы.

Параметры: нет.

Возвращаемое значение: `int` - Код ошибки.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `main`

| № | Предикат | Действия | № перехода |
|----|--|---|---------------|
| 1 | | Объявление переменной <code>i_data</code> типа <code>int</code> | 2 |
| 2 | | Ввод значения переменной <code>i_data</code> | 3 |
| 3 | | Создание объекта <code>Example</code> класса <code>Test</code> посредством оператора функции <code>new</code> и использованием указателя на объект, передавая параметризованному конструктору <code>i_data</code> . | 4 |
| 4 | | Вызов метода <code>show</code> у объекта <code>Example</code> | 5 |
| 5 | | Вызов метода <code>pub_add</code> у объекта <code>Example</code> | 6 |
| 6 | | Вызов метода <code>show</code> у объекта <code>Example</code> | 7 |
| 7 | | Ввод значения переменной <code>i_data</code> | 8 |
| 8 | <code>i_data > Example->pub</code> | <code>Example->pub = i_data * 8;</code> | 9 |
| | | Вызов метода <code>show</code> у объекта <code>Example</code> | 9 |
| 9 | | Вызов метода <code>call_hid</code> у объекта <code>Example</code> | 10 |
| 10 | | Вызов метода <code>show</code> у объекта <code>Example</code> | 11 |
| 11 | | Удаление памяти по указателю <code>Example</code> | 12 |

| № | Предикат | Действия | № перехода |
|----|----------|--------------------|---------------|
| 12 | | Возврат значения 0 | ∅ |

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

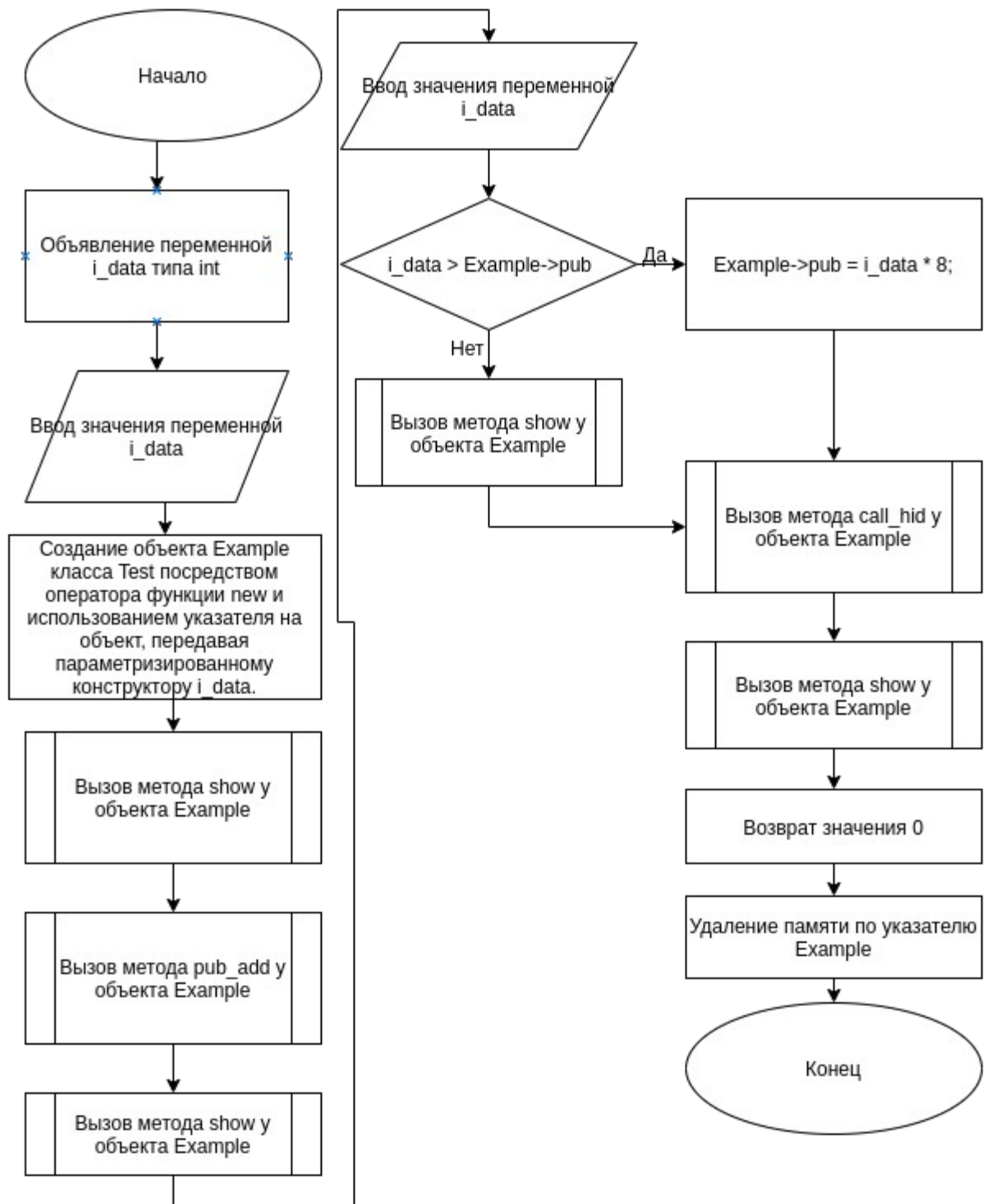


Рисунок 1 – Блок-схема алгоритма

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 "Test.h"
#include <iostream>

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

    Test *Example = new Test(i_data);

    Example->show();
    Example->pub_add();
    Example->show();

    std::cin >> i_data;
    if(i_data > Example->pub)
        Example->pub = i_data * 8;

    Example->show();
    Example->call_hid();
    Example->show();

    delete Example;

    return(0);
}
```

5.2 Файл Test.cpp

Листинг 2 – Test.cpp

```
#include "Test.h"
```

```

#include <iostream>

Test::Test(int i_data)
{
    pub = i_data;
    hid = i_data * 3;
}

void Test::pub_add () {
    pub += 4;
    hid += 1;
}

void Test::call_hid() {
    hid_add();
}

void Test::show() {
    std::cout << "Value of the available property " << pub << "; Value of a
hidden property " << hid << std::endl;
}

void Test::hid_add() {
    pub += 7;
    hid += 5;
}

```

5.3 Файл Test.h

Листинг 3 – Test.h

```

#ifndef __TEST__H
#define __TEST__H

class Test
{
public:
    int pub;
    Test(int i_data);
    void pub_add ();
    void call_hid();
    void show();
private:
    int hid;
    void hid_add();
};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 2.

Таблица 2 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
|----------------|--|--|
| 3 1 | 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 (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).