

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 метода current_state класса Tsygane.....	9
3.2 Алгоритм конструктора класса Tsygane.....	9
3.3 Алгоритм метода change класса Tsygane.....	10
3.4 Алгоритм метода h_change класса Tsygane.....	10
3.5 Алгоритм метода call_hidden класса Tsygane.....	10
3.6 Алгоритм функции main.....	11
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	15
5.1 Файл main.cpp.....	15
5.2 Файл Tsygane.cpp.....	16
5.3 Файл Tsygane.h.....	16
6 ТЕСТИРОВАНИЕ.....	18
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 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 cin класса istream предназначен для ввода данных;
- объект cout класса ostream предназначен для вывода данных;
- объект класса Tsygane предназначен для ;
- new - оператор резервирования памяти;
- delete - оператор освобождения памяти;
- if..else - условный оператор.

Класс Tsygane:

- свойства/поля:
 - поле доступное свойство:
 - наименование — data;
 - тип — int;
 - модификатор доступа — public;
 - поле закрытое свойство:
 - наименование — h_data;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод current_state — вывод текущего состояния объекта;
 - метод Tsygane — конструктор, присваивание свойствам класса переданного значения;
 - метод change — увеличение значения доступного свойства на 4, закрытого на 1;
 - метод h_change — увеличение значения доступного свойства на 7, закрытого на 5;

- o метод `call_hidden` — вызов закрытого метода `h_change()`.

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 метода `current_state` класса `Tsygane`

Функционал: вывод текущего состояния объекта.

Параметры: нет.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `current_state` класса `Tsygane`

№	Предикат	Действия	№ перехода
1		Вывод значений свойств доступного/закрытого	Ø

3.2 Алгоритм конструктора класса `Tsygane`

Функционал: конструктор, присваивание свойствам класса переданного значения параметра.

Параметры: целочисленный - `i_data`.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса `Tsygane`

№	Предикат	Действия	№ перехода
1		Присваивание доступному свойству утроенного значения параметра	2
2		Присваивание закрытому свойству значения параметра	Ø

3.3 Алгоритм метода **change** класса **Tsygane**

Функционал: увеличение доступного свойства на 4, закрытого на 1.

Параметры: нет.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *change* класса *Tsygane*

№	Предикат	Действия	№ перехода
1		data += 4	2
2		h_data++	∅

3.4 Алгоритм метода **h_change** класса **Tsygane**

Функционал: увеличение доступного свойства на 7, закрытого на 5.

Параметры: нет.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *h_change* класса *Tsygane*

№	Предикат	Действия	№ перехода
1		data += 7	2
2		h_data += 5	∅

3.5 Алгоритм метода **call_hidden** класса **Tsygane**

Функционал: вызов закрытого метода *h_change()*.

Параметры: нет.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *call_hidden* класса *Tsygane*

№	Предикат	Действия	№ перехода
1		Вызов метода <i>h_change()</i>	Ø

3.6 Алгоритм функции *main*

Функционал: точка входа в программу.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности выполнения программы.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>i_data</i>	2
2		Ввод значения переменной <i>i_data</i> с клавиатуры	3
3		Инициализация указателя <i>entity</i> на объект класса <i>Tsygane</i>	4
4		Вызов метода <i>current_state()</i>	5
5		Вызов метода <i>change()</i>	6
6		Вызов метода <i>current_state()</i>	7
7		Ввод значения переменной <i>i_data</i> с клавиатуры	8
8	<i>i_data > entity->data</i>	Присваивание доступному свойству объекта значения <i>i_data * 8</i>	9
		Переход на следующий шаг	9
9		Вызов метода <i>current_state()</i>	10
10		Вызов метода <i>call_hidden()</i>	11
11		Вызов метода <i>current_state()</i>	12

№	Предикат	Действия	№ перехода
12		Удаление объекта по указателю	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

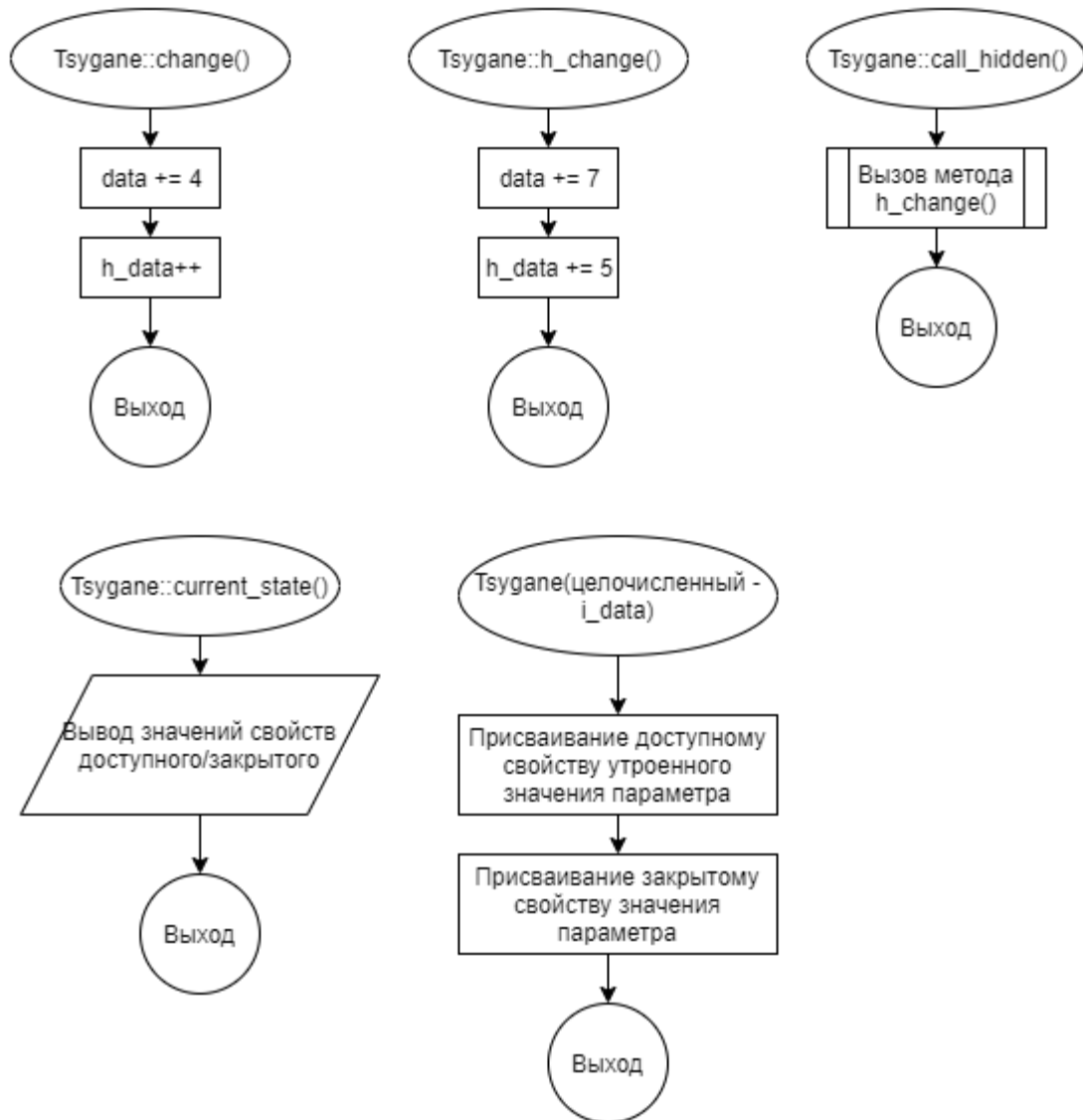


Рисунок 1 – Блок-схема алгоритма

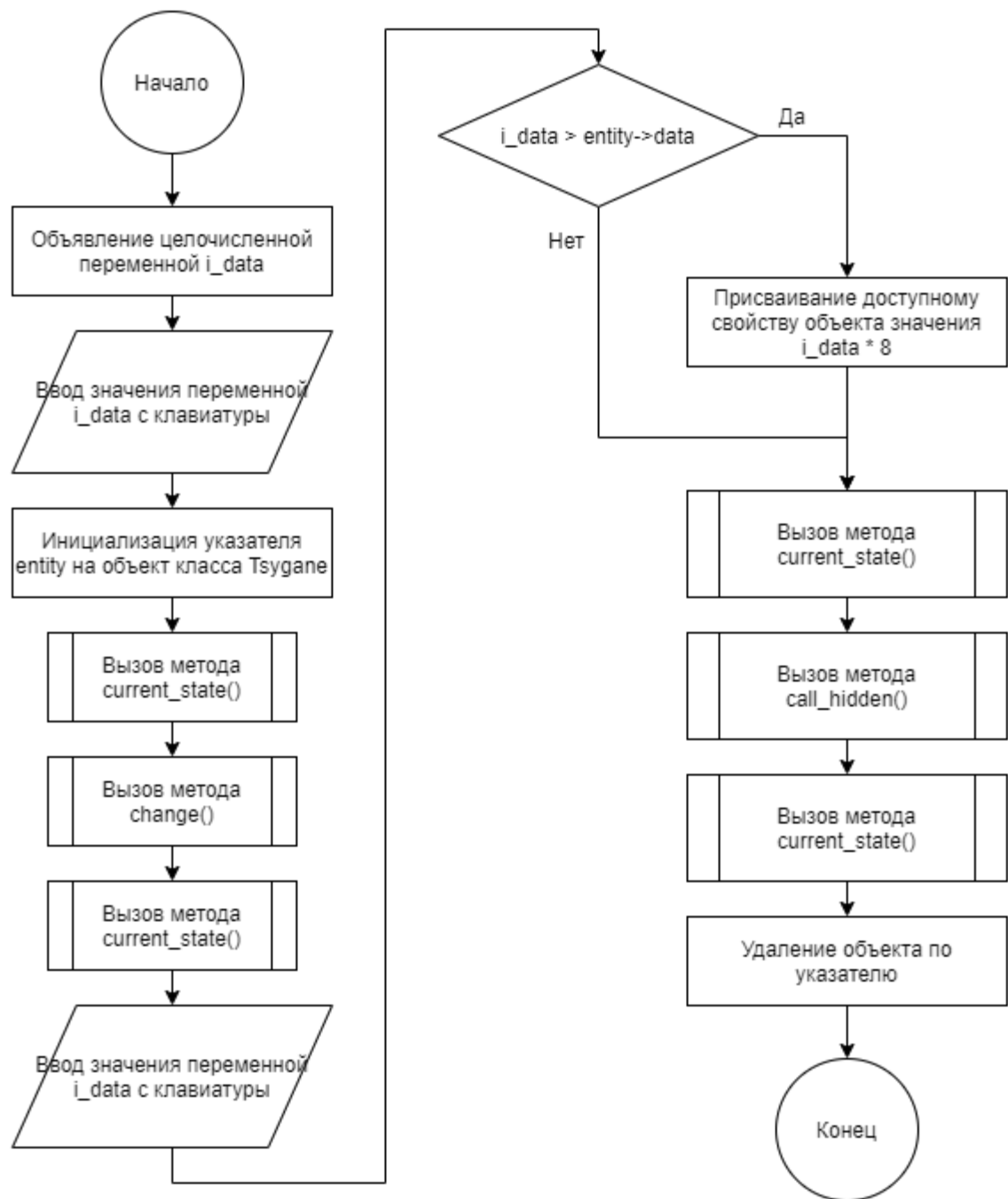


Рисунок 2 – Блок-схема алгоритма

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 "Tsygane.h"

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

    Tsygane *entity = new Tsygane(i_data);

    entity->current_state();
    std::cout << std::endl;

    entity->change();

    entity->current_state();
    std::cout << std::endl;

    std::cin >> i_data;

    if (i_data > entity->data) {
        entity->data = i_data * 8;
    }

    entity->current_state();
    std::cout << std::endl;

    entity->call_hidden();

    entity->current_state();

    delete entity;

    return 0;
}
```

5.2 Файл Tsygane.cpp

Листинг 2 – Tsygane.cpp

```
#include <iostream>
#include "Tsygane.h"

Tsygane::Tsygane(int tabor) {
    data = tabor;
    h_data = tabor * 3;
}

void Tsygane::change() {
    data += 4;
    h_data++;
}

void Tsygane::call_hidden() {
    h_change();
}

void Tsygane::h_change() {
    data += 7;
    h_data += 5;
}

void Tsygane::current_state() {
    std::cout << "Value of the available property " << data << "; Value of a
hidden property " << h_data;
}
```

5.3 Файл Tsygane.h

Листинг 3 – Tsygane.h

```
#ifndef __TSYGANE_V_AVDEEVKE__H
#define __TSYGANE_V_AVDEEVKE__H

class Tsygane {

private:
    int h_data;
    void h_change();

public:
    int data;

    Tsygane(int tabor);
```

```
void change();  
void call_hidden();  
void current_state();  
};  
#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

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).