

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 current_state класса Tsygane.....	9
3.3 Алгоритм метода change класса Tsygane.....	9
3.4 Алгоритм метода h_change класса Tsygane.....	9
3.5 Алгоритм метода call_hidden класса Tsygane.....	10
3.6 Алгоритм конструктора класса Tsygane.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	13
5.1 Файл main.cpp.....	13
5.2 Файл Tsygane.cpp.....	14
5.3 Файл Tsygane.h.....	14
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 `cout` класса `ostream`;
- объект `cin` класса `istream`;
- объект класса `Tsygane`.

Класс `Tsygane`:

- свойства/поля:
 - поле хранение закрытого свойства:
 - наименование — `h_data`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле хранение доступного свойства:
 - наименование — `data`;
 - тип — `int`;
 - модификатор доступа — `public`;
- функционал:
 - метод `current_state` — вывод текущего состояния объекта;
 - метод `change` — увеличение значения доступного свойства на 1, закрытого на 4;
 - метод `h_change` — увеличение значения доступного свойства на 5, закрытого на 7;
 - метод `call_hidden` — вызов закрытого метода `h_change()`;
 - метод `Tsygane` — присваивание свойствам класса переданного значения.

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: точка входа в программу.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности выполнения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной i_data	2
2		Ввод значения переменной i_data с клавиатуры	3
3		Объявление объекта tsygan класса Tsygane	4
4		Вызов метода current_state()	5
5		Вызов метода change()	6
6		Вызов метода current_state()	7
7		Ввод значения переменной i_data с клавиатуры	8
8		Присваивание доступному свойству data класса Tsygan значения переменной умноженного на data	9
9		Вызов метода current_state()	10
10		Вызов метода call_hidden()	11
11		Вызов метода current_state()	Ø

3.2 Алгоритм метода `current_state` класса `Tsygane`

Функционал: вывод текущего состояния объекта.

Параметры: нет.

Возвращаемое значение: текст, текущее состояние объекта.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `current_state` класса `Tsygane`

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

3.3 Алгоритм метода `change` класса `Tsygane`

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

Параметры: нет.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 3.

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

№	Предикат	Действия	№ перехода
1		<code>data++</code>	2
2		<code>h_data += 4</code>	Ø

3.4 Алгоритм метода `h_change` класса `Tsygane`

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

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 += 5	2
2		h_data += 7	∅

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>	∅

3.6 Алгоритм конструктора класса *Tsygane*

Функционал: присваивание свойствам класса переданного значения.

Параметры: целочисленный *tabor*.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса *Tsygane*

№	Предикат	Действия	№ перехода
1		Присваивание <i>data</i> переданного параметра	2
2		Присваивание <i>h_data</i> удвоенного переданного параметра	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

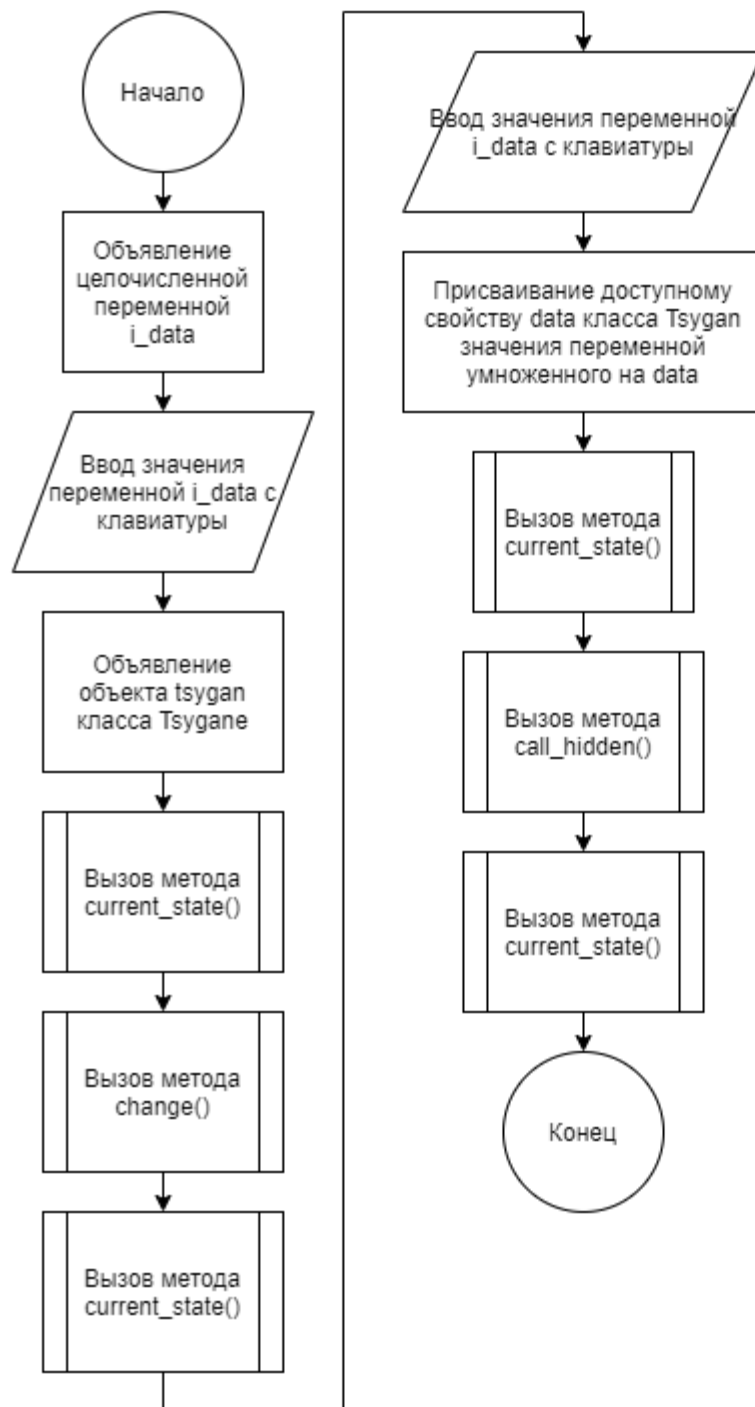


Рисунок 1 – Блок-схема алгоритма

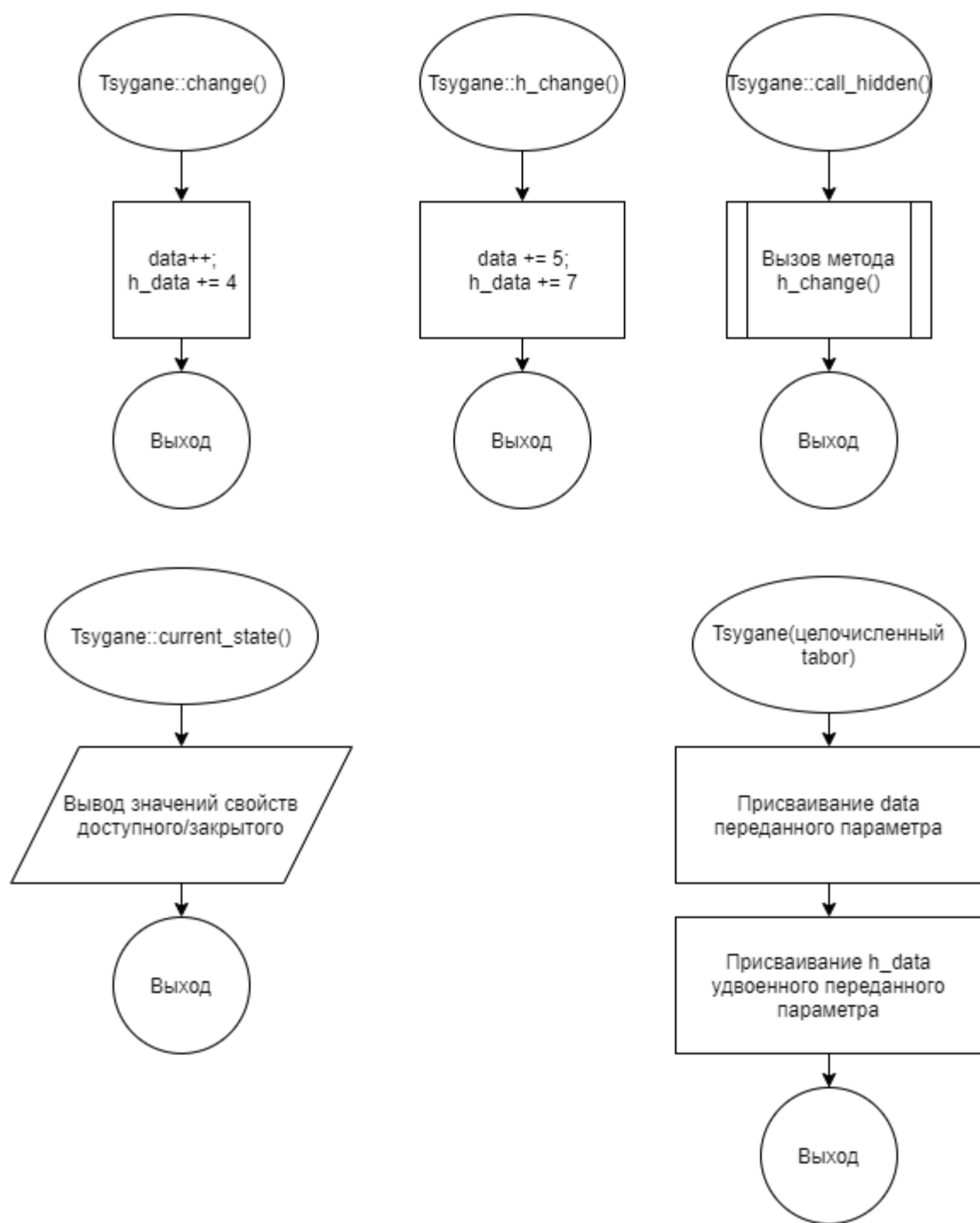


Рисунок 2 – Блок-схема алгоритма

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 tsygan(i_data);

    tsygan.current_state();
    std::cout << std::endl;

    tsygan.change();

    tsygan.current_state();
    std::cout << std::endl;

    std::cin >> i_data;
    tsygan.data *= i_data;

    tsygan.current_state();
    std::cout << std::endl;
    tsygan.call_hidden();

    tsygan.current_state();

    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 * 2;
}

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

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

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

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 2 Value of the available property 2; Value of a hidden property 6 Value of the available property 4; Value of a hidden property 6 Value of the available property 9; Value of a hidden property 13	Value of the available property 1; Value of a hidden property 2 Value of the available property 2; Value of a hidden property 6 Value of the available property 4; Value of a hidden property 6 Value of the available property 9; Value of a hidden property 13

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).