

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 prisvoenie класса horrible.....	9
3.3 Алгоритм метода izmenenie класса horrible.....	9
3.4 Алгоритм метода skritiy класса horrible.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	13
5.1 Файл horrible.cpp.....	13
5.2 Файл horrible.h.....	14
5.3 Файл main.cpp.....	14
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая демонстрирует возможность определения режима доступа к элементам объекта.

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 доступные методы со следующим функционалом:

- метод с параметром целого типа, для определения исходного значения элементов объекта. В методе доступному свойству присваивается значение параметра, а скрытому свойству утроенное значение параметра;
- 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, где доступному свойству добавляется 9, скрытому свойству добавляется 11.

Написать программу, которая состоит из описания класса представленного выше объекта и основной функции, в которой реализован следующий алгоритм:

1. Ввод целочисленного значения переменной `i_data`.
2. Создание объекта с помощью оператора `new`.
3. Вызов метода для определения исходного значения элементов объекта, в качестве аргумента передается переменная `i_data`.
4. Вывод исходного состояния объекта.
5. Вызов метода изменения значений свойств объекта.

6. Вывод текущего состояния объекта.
7. Ввод целочисленного значения переменной `i_data`.
8. Если значение `i_data` больше, чем значение доступного свойства объекта, то
 - 8.1. Присвоение доступному свойству объекта значение выражения `i_data * 8` и переход к пункту 10.
9. Иначе
 - 9.1. Переход к пункту 10.
10. Вывод текущего состояния объекта.
11. Вызов метода объекта, который вызывает скрытый метод объекта.
12. Вывод текущего состояния объекта.
13. Удаление 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 pussy класса horrible предназначен для ;
- оператор if;
- операторы-функция new для ;
- операторы-функция delete для очищения выделенной памяти.

Класс horrible:

- свойства/поля:
 - поле доступное свойство объекта:
 - наименование — n;
 - тип — int;
 - модификатор доступа — private;
 - поле скрытое свойство объекта:
 - наименование — r;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод prisoenie — доступному свойству присваивается значение параметра, а скрытому утроенное значение параметра;
 - метод izmenenie — доступному свойству добавляется 4, а скрытому 1;
 - метод skritiy — к доступному свойству прибавляется 9, к скрытому 11.

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: Главная функция программы.

Параметры: нет.

Возвращаемое значение: Целое число, индикатор работоспособности 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	3
3		Создание объекта класса horrible, инициализация указателя first_obj на класс horrible адресом этого объекта	4
4		Вызов метода prisoenie объекта по указателю first_obj со значением i_data в качестве параметра	5
5		Вызов метода sosto объекта по указателю first_obj	6
6		Вызов метода izmenenie объекта по указателю first_obj	7
7		Вызов метода sosto объекта по указателю first_obj	8
8		Ввод значения переменной i_data	9
9	i_data больше доступного свойства объекта по	Присвоение доступному свойству объекта по указателю test_obj значение выражения i_data*8	10

№	Предикат	Действия	№ перехода
	указателю test_obj		
			10
10		Вызов метода sosto объекта по указателю first_obj	11
11		Вызов метода vyzov_skryt объекта по указателю first_obj	12
12		Вызов метода sosto объекта по указателю first_obj	13
13		Удаление объекта по указателю test_obj	14
14		Вернуть значение 0	∅

3.2 Алгоритм метода prisvoenie класса horrible

Функционал: доступному свойству присваивается значение параметра, а скрытому утроенное значение параметра.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода prisvoenie класса horrible

№	Предикат	Действия	№ перехода
1		Присвоить доступному свойству значение параметра n	2
2		Присвоить скрытому свойству значение утроенное параметра n	∅

3.3 Алгоритм метода izmenenie класса horrible

Функционал: доступному свойству добавляется 4, а скрытому 1.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *izmenenie* класса *horrible*

№	Предикат	Действия	№ перехода
1		Увеличение значения доступного свойства на 4	2
2		Увеличение значения скрытого свойства на 1	∅

3.4 Алгоритм метода *skritiy* класса *horrible*

Функционал: к доступному свойству прибавляется 9, к скрытому 11.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *skritiy* класса *horrible*

№	Предикат	Действия	№ перехода
1		Увеличение значения доступного свойства на 9	2
2		Увеличение значения скрытого свойства на 11	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

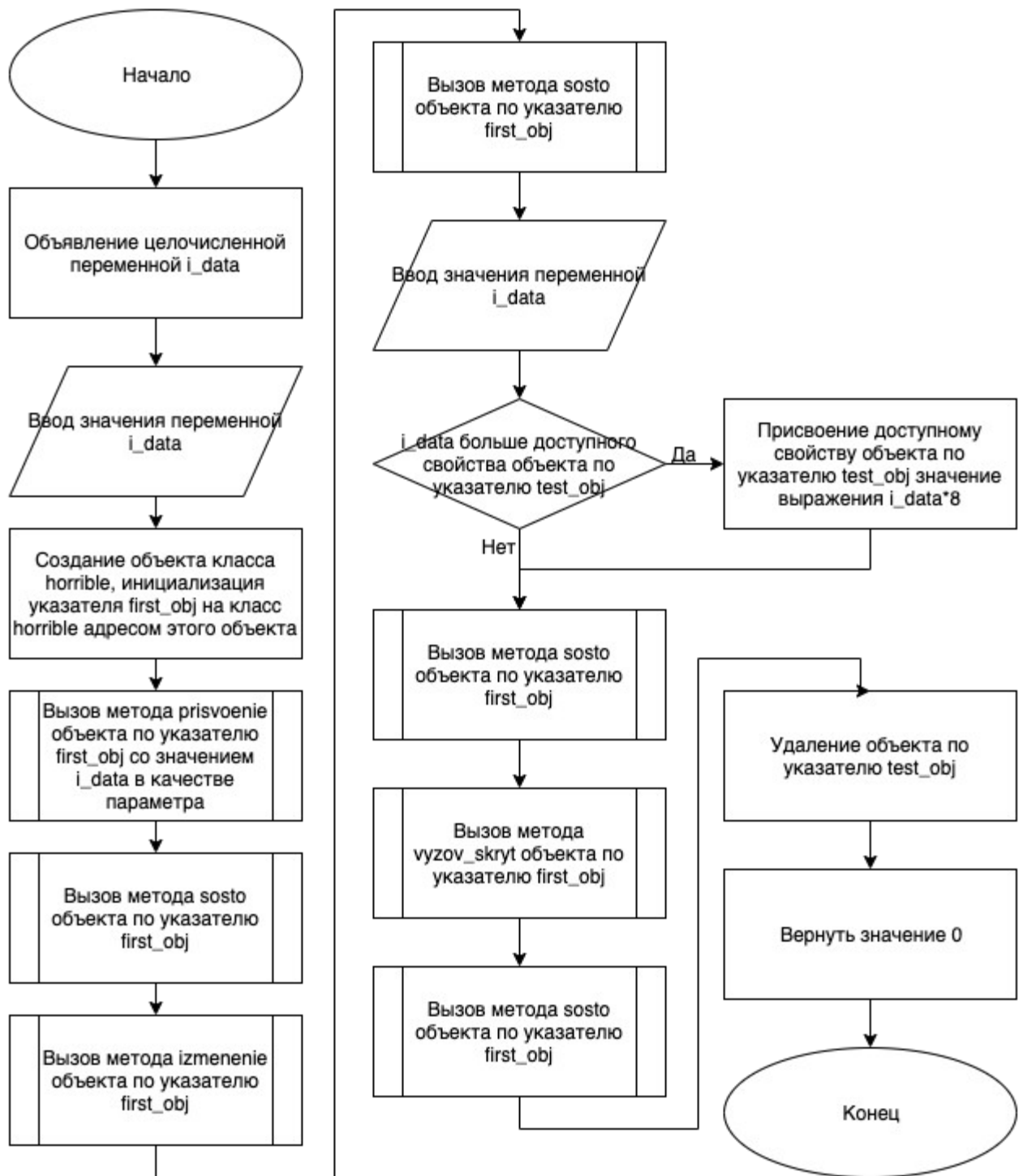


Рисунок 1 – Блок-схема алгоритма

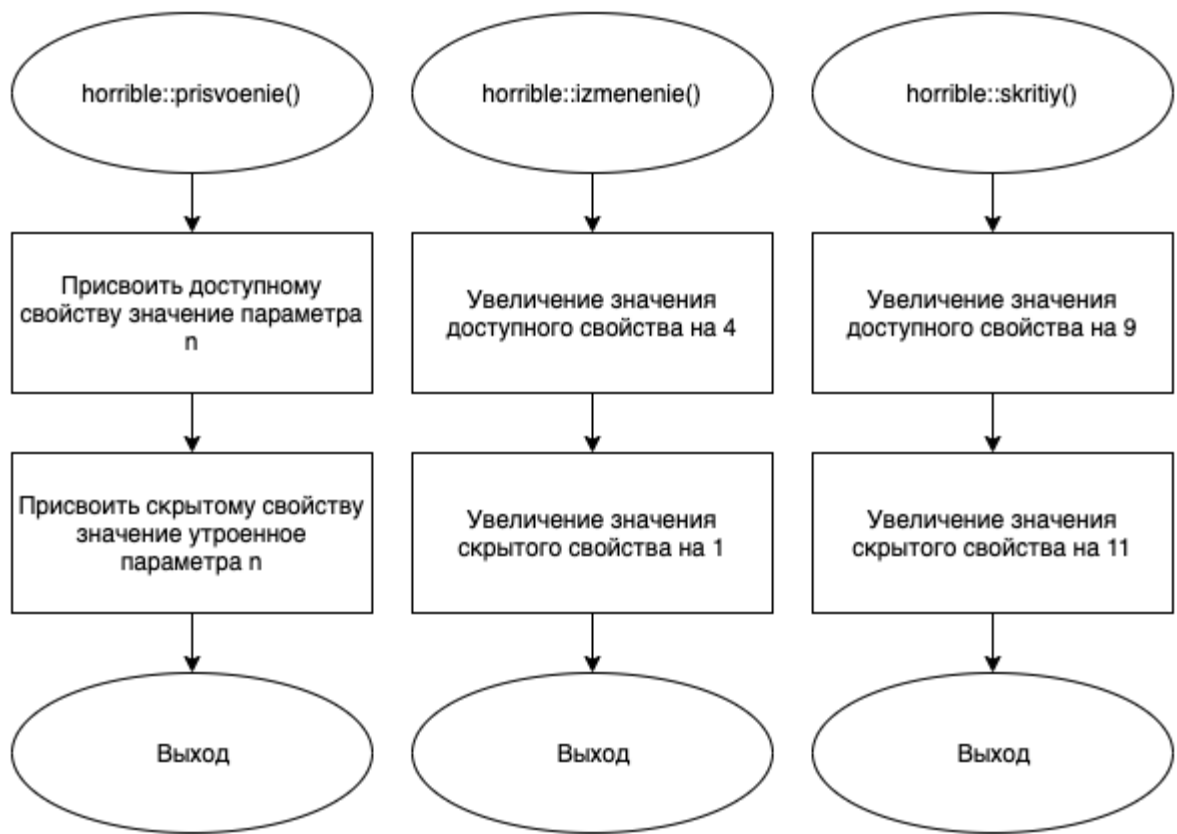


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл horrible.cpp

Листинг 1 – horrible.cpp

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

horrible::horrible() : n(0), r(0) {}
horrible::~horrible(){}

void horrible::prisvoit(int i)
{
    n = i;
    r = i * 3;
}

void horrible::izmenit()
{
    n += 4;
    r += 1;
}

void horrible::vyzov_skryt()
{
    skrytiy();
}

void horrible::sosto()
{
    cout << "Value of the available property " << n << "; Value of a hidden
property " << r << "\n";
}

void horrible::skrytiy()
{
    n += 9;
    r += 11;
}
```

5.2 Файл horrible.h

Листинг 2 – horrible.h

```
#ifndef __HORRIBLE__H
#define __HORRIBLE__H
using namespace std;

class horrible
{
public:
    int n;

    void prisvoit(int i);
    void izmenit();
    void vyzov_skryt();
    void sosto();

    horrible();
    ~horrible();

private:
    int r;

    void skrytiy();

};

#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

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

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

    horrible* pussy = new horrible();
    pussy->prisvoit(i_data);
    pussy->sosto();
}
```

```
    pussy->izmenit();  
    pussy->sosto();  
  
    cin >> i_data;  
    if (i_data > pussy->n) {  
        pussy->n = i_data * 8;  
    }  
    pussy->sosto();  
  
    pussy->vyzov_skryt();  
    pussy->sosto();  
  
    delete pussy;  
    return(0);  
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 3	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 14; Value of a hidden property 15	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 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).