

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 конструктора класса horrible.....	9
3.2 Алгоритм деструктора класса horrible.....	9
3.3 Алгоритм метода prisvoit класса horrible.....	9
3.4 Алгоритм метода izmenit класса horrible.....	10
3.5 Алгоритм метода vyzov_skryt класса horrible.....	10
3.6 Алгоритм метода sosto класса horrible.....	11
3.7 Алгоритм метода skrytiy класса horrible.....	11
3.8 Алгоритм функции main.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	17
5.1 Файл horrible.cpp.....	17
5.2 Файл horrible.h.....	18
5.3 Файл main.cpp.....	18
6 ТЕСТИРОВАНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

# 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 удвоенное значение параметра;
- 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 4, скрытому свойству добавляет 8.

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. Объявление объекта,
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. Непосредственное изменение доступного свойства объекта посредством умножения его значения на i\_data.
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 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 `pusy` класса `horrible` предназначен для демонстрации возможности определения режима доступа к элементам объекта;
- библиотека `iostream`;
- объект стандартного ввода данных `cin`;
- объект стандартного вывода данных `cout`;
- оператор присваивания `=`;
- целочисленная переменная `i_data`.

Класс `horrible`:

- свойства/поля:
  - поле доступное свойство объекта:
    - наименование — `n`;
    - тип — `int`;
    - модификатор доступа — `public`;
  - поле скрытое свойство объекта:
    - наименование — `r`;
    - тип — `int`;
    - модификатор доступа — `protected`;
- функционал:
  - метод `horrible` — конструктор;
  - метод `~horrible` — деструктор;
  - метод `prisvoit` — метод с параметром целого типа, для определения исходного значения элементов объекта. В методе доступному свойству присваивается значение параметра, а скрытому свойству удвоенное значение параметра;

- о метод `izmenit` — метод изменения значения доступного и скрытого свойства. Доступному свойству добавляется 1, скрытому свойству добавляется 4;
- о метод `vyzov_skryt` — метод вызова скрытого метода;
- о метод `sosto` — метод вывода состояния, выводит значение доступного и скрытого свойства;
- о метод `skrytiy` — метод доступному свойству добавляет 4, скрытому свойству добавляет 8.

## 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 *horrible*

Функционал: конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *horrible*

№	Предикат	Действия	№ перехода
1			Ø

### 3.2 Алгоритм деструктора класса *horrible*

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса *horrible*

№	Предикат	Действия	№ перехода
1			Ø

### 3.3 Алгоритм метода *prisvoit* класса *horrible*

Функционал: метод с параметром целого типа, для определения исходного значения элементов объекта. В методе доступному свойству присваивается

значение параметра, а скрытому свойству удвоенное значение параметра.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

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

№	Предикат	Действия	№ перехода
1		Доступному свойству присваивается значение параметра ( $n = i$ )	2
2		Доступному свойству присваивается удвоенное значение параметра ( $r = i*2$ )	Ø

### 3.4 Алгоритм метода *izmenit* класса *horrible*

Функционал: метод изменения значения доступного и скрытого свойства.  
Доступному свойству добавляется 1, скрытому свойству добавляется 4.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

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

№	Предикат	Действия	№ перехода
1		Доступному свойству добавляется 1 ( $n += 1$ )	2
2		Скрытому свойству добавляется 4 ( $r += 4$ )	Ø

### 3.5 Алгоритм метода *vyzov\_skryt* класса *horrible*

Функционал: метод вызова скрытого метода.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.



Таблица 5 – Алгоритм метода *vzov\_skryt* класса *horrible*

№	Предикат	Действия	№ перехода
1		вызов скрытого метода <i>skrytiy()</i>	Ø

### 3.6 Алгоритм метода *sosto* класса *horrible*

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

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода *sosto* класса *horrible*

№	Предикат	Действия	№ перехода
1		Вывод "Value of the available property ", значение n, "; Value of a hidden property:", значение r, "\n"	Ø

### 3.7 Алгоритм метода *skrytiy* класса *horrible*

Функционал: метод доступному свойству добавляет 4, скрытому свойству добавляет 8.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *skrytiy* класса *horrible*

№	Предикат	Действия	№ перехода
1		Доступному свойству добавляется 4 ( $n += 4$ )	2
2		Скрытому свойству добавляется 8 ( $r += 8$ )	Ø

### 3.8 Алгоритм функции main

Функционал: Главная функция программы.

Параметры: нет.

Возвращаемое значение: Целое число, индикатор работоспособности программы.

Алгоритм функции представлен в таблице 8.

Таблица 8 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной i_data	2
2		Ввод целочисленного значения переменной i_data	3
3		Объявление объекта pussy класса horrible	4
4		Вызов метода prisvoit() для определения исходного значения элементов объекта, в качестве аргумента передается переменная i_data	5
5		Вывод исходного состояния объекта методов sosto()	6
6		Вызов метода izmenit() для изменения значений свойств объекта	7
7		Вывод текущего состояния объекта методов sosto()	8
8		Ввод целочисленного значения переменной i_data	9
9		Непосредственное изменение доступного свойства объекта посредством умножения его значения на i_data	10
10		Вывод текущего состояния объекта методом sosto()	11
11		Вызов метода vyzov_skryt(), который вызывает скрытый метод skrytiy()	12
12		Вывод текущего состояния объекта методом sosto()	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

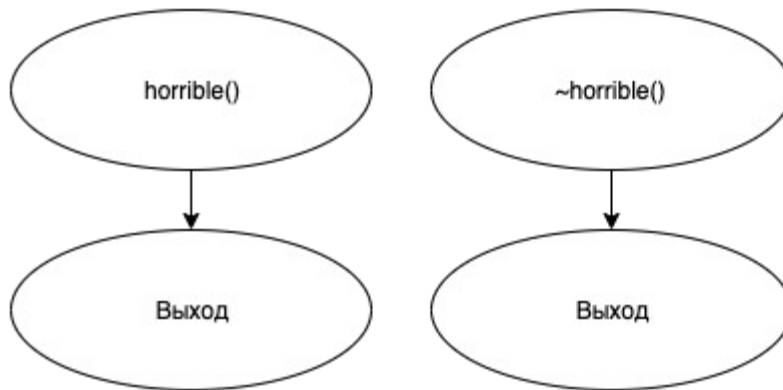


Рисунок 1 – Блок-схема алгоритма

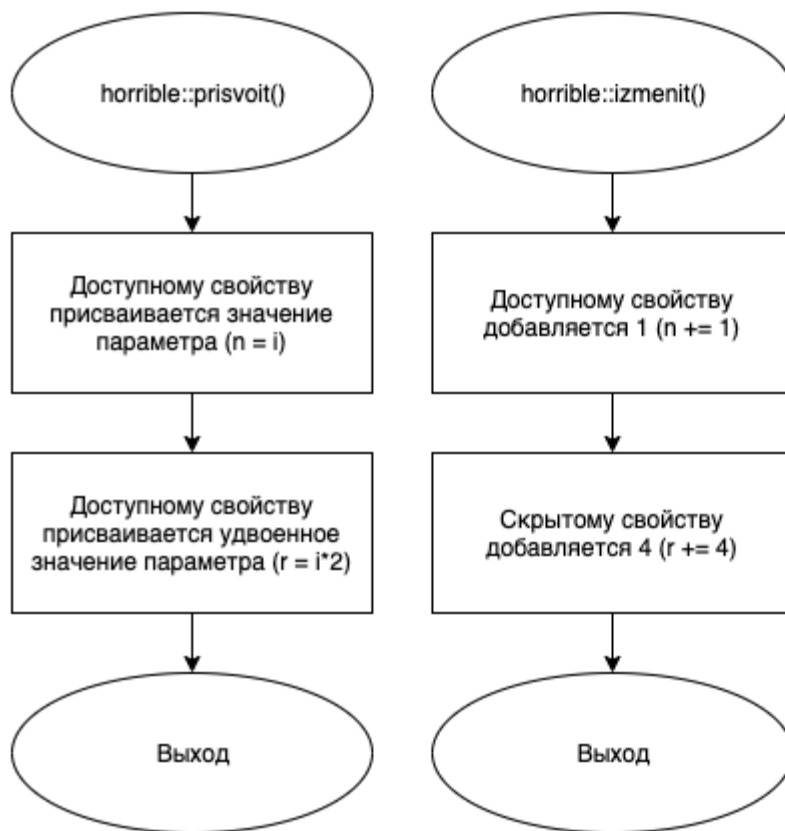


Рисунок 2 – Блок-схема алгоритма

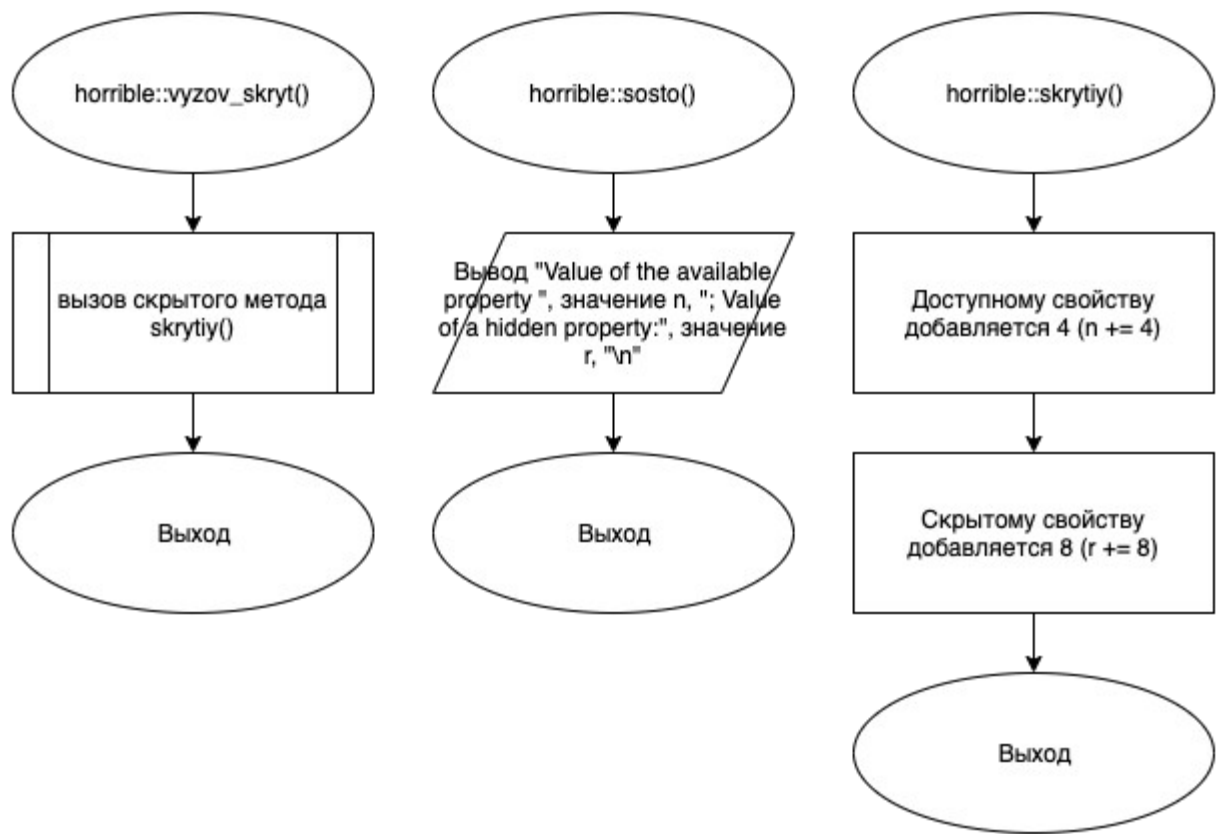


Рисунок 3 – Блок-схема алгоритма

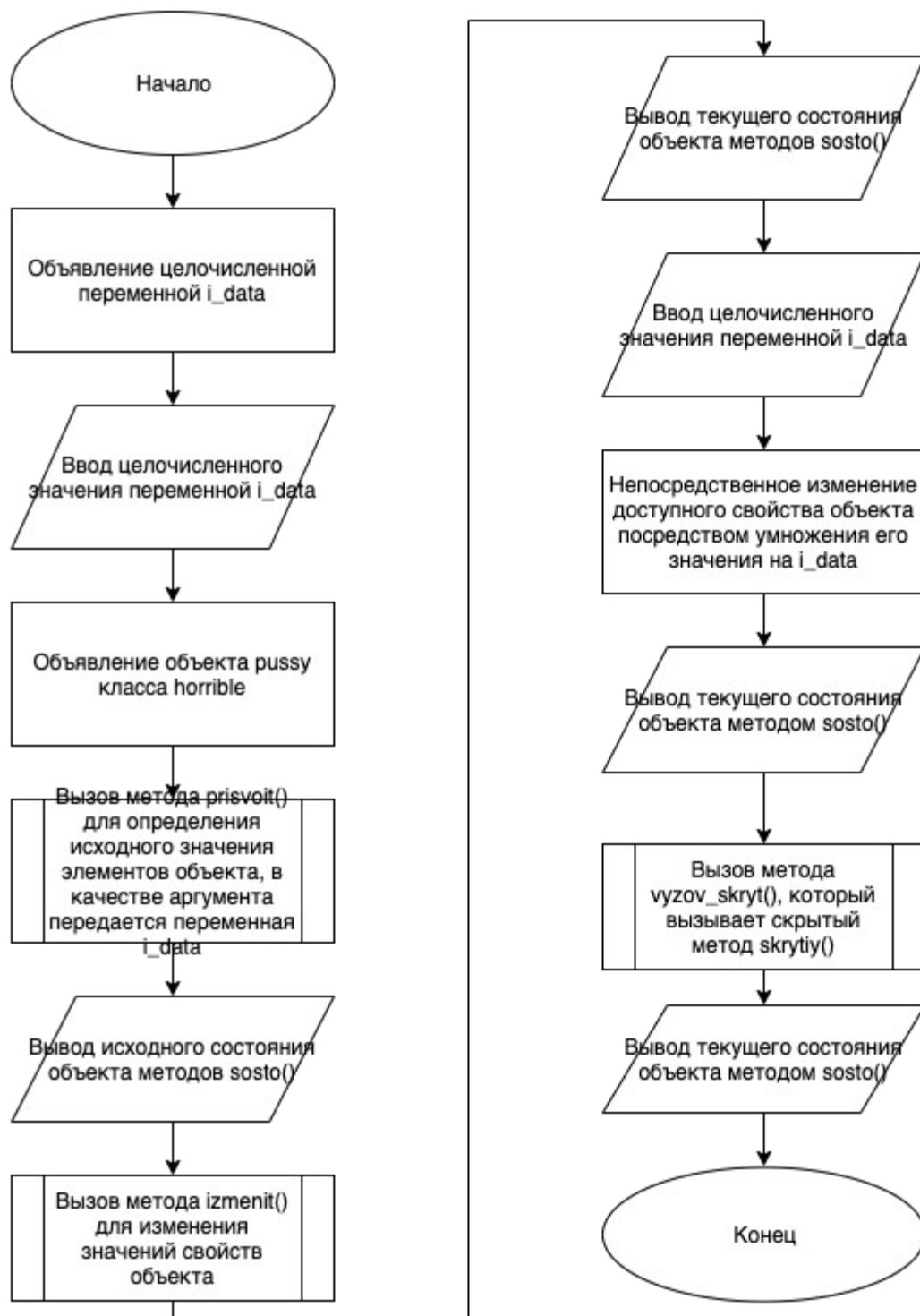


Рисунок 4 – Блок-схема алгоритма

## 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 * 2;
}

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

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 += 4;
    r += 8;
}
```

## 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;
    pussy.prisvoit(i_data);
    pussy.sosto();
}
```



```
    pussy.izmenit();  
    pussy.sosto();  
  
    cin >> i_data;  
    pussy.n *= i_data;  
    pussy.sosto();  
  
    pussy.vyzov_skryt();  
    pussy.sosto();  
  
    return(0);  
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 9.

Таблица 9 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 3	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 6; Value of a hidden property 6 Value of the available property 10; Value of a hidden property 14	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 6; Value of a hidden property 6 Value of the available property 10; Value of a hidden property 14
10 50	Value of the available property 10; Value of a hidden property 20 Value of the available property 11; Value of a hidden property 24 Value of the available property 550; Value of a hidden property 24 Value of the available property 554; Value of a hidden property 32	Value of the available property 10; Value of a hidden property 20 Value of the available property 11; Value of a hidden property 24 Value of the available property 550; Value of a hidden property 24 Value of the available property 554; Value of a hidden property 32

## 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/methodichescoc\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avvora.ru/student/files/methodichescoc_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](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).