Здесь будет титульник, листай ниже

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 5](#__RefHeading___Toc2522_1702449289)

[1.1 Описание входных данных 6](#__RefHeading___Toc2524_1702449289)

[1.2 Описание выходных данных 6](#__RefHeading___Toc2526_1702449289)

[2 МЕТОД РЕШЕНИЯ 7](#__RefHeading___Toc2528_1702449289)

[3 ОПИСАНИЕ АЛГОРИТМОВ 8](#__RefHeading___Toc2530_1702449289)

[3.1 Алгоритм функции main 8](#__RefHeading___Toc2532_1702449289)

[3.2 Алгоритм конструктора класса Object 9](#__RefHeading___Toc2534_1702449289)

[3.3 Алгоритм метода ChangePropsPublic класса Object 9](#__RefHeading___Toc2536_1702449289)

[3.4 Алгоритм метода PrivateCall класса Object 10](#__RefHeading___Toc2538_1702449289)

[3.5 Алгоритм метода StatusOutput класса Object 10](#__RefHeading___Toc2540_1702449289)

[3.6 Алгоритм метода ChangePropsPrivate класса Object 11](#__RefHeading___Toc2542_1702449289)

[4 БЛОК-СХЕМЫ АЛГОРИТМОВ 12](#__RefHeading___Toc2544_1702449289)

[5 КОД ПРОГРАММЫ 14](#__RefHeading___Toc2546_1702449289)

[5.1 Файл main.cpp 14](#__RefHeading___Toc2548_1702449289)

[5.2 Файл Object.cpp 14](#__RefHeading___Toc2550_1702449289)

[5.3 Файл Object.h 15](#__RefHeading___Toc2552_1702449289)

[6 ТЕСТИРОВАНИЕ 16](#__RefHeading___Toc2554_1702449289)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#__RefHeading___Toc2556_1702449289)

# 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 параметризированный 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:

* Ввод целочисленного значения переменной i\_data.
* Создание объекта, параметризированному конструктору в качестве аргумента передается переменная i\_data.
* Вывод исходного состояния объекта.
* Вызов метода изменения значений свойств объекта.
* Вывод текущего состояния объекта.
* Ввод целочисленного значения переменной i\_data.
* Непосредственное изменение доступного свойства объекта посредством умножения его значения на i\_data.
* Вывод текущего состояния объекта.
* Вызов метода объекта, который вызывает скрытый метод объекта.
* 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

В программе используется библиотека iostream, пространство имен std, поток ввода - cin и поток вывода на экран - cout.

Класс Object:

Свойства/поля:

* Поле, отвечающее за хранение некоторого значения
  + Наименование - dataPublic
  + Тип - int
  + Модификатор доступа - public
* Поле, отвечающее за хранение некоторого значения
  + Наименование - dataPrivate
  + Тип - int
  + Модификатор доступа - private

Функционал:

* Метод Object - параметризированный конструктор для создания объекта класса и изменения значений полей с помощью параметра
* Метод ChangePropsPublic - используется для изменения значений доступного и скрытого поля
* Метод PrivateCall - используется для вызова скрытого метода ChangePropsPrivate
* Метод StatusOutput - используется для вывода состояния - значения доступного и скрытого поля
* Метод ChangePropsPrivate - скрытый метод, использующийся для изменения значений доступного и скрытого поля

# 3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

## Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: Отсутствуют.

Возвращаемое значение: int - индикатор корректности завершения 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 |  | Создание объекта obj класса Object путём вызова конструктора с параметром data | 4 |
| 4 |  | Вызов метода StatusOutput объекта obj | 5 |
| 5 |  | Переход на новую строку | 6 |
| 6 |  | Вызов метода ChangePropsPublic объекта obj | 7 |
| 7 |  | Вызов метода StatusOutput объекта obj | 8 |
| 8 |  | Переход на новую строку | 9 |
| 9 |  | Ввод значения переменной i\_data с клавиатуры | 10 |
| 10 |  | Присваивание полю dataPublic объекта obj значение поля dataPublic, умноженного на значение переменной i\_data | 11 |
| 11 |  | Вызов метода StatusOutput объекта obj | 12 |
| 12 |  | Переход на новую строку | 13 |
| 13 |  | Вызов метода PrivateCall объекта obj | 14 |
| 14 |  | Вызов метода StatusOutput объекта obj | 15 |
| 15 |  | Возвращение значения работы функции 0 | ∅ |

## Алгоритм конструктора класса Object

Функционал: Создание объекта класса с именем Object.

Параметры: int data - используется для изменения полей класса.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Присваивание значения параметра data полю dataPublic | 2 |
| 2 |  | Присваивание удвоенного значения параметра data полю dataPrivate | ∅ |

## Алгоритм метода ChangePropsPublic класса Object

Функционал: Изменение значений доступного и скрытого поля.

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода ChangePropsPublic класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Инкремент значения поля dataPublic | 2 |
| 2 |  | Увеличение значения поля dataPrivate на 4 | ∅ |

## Алгоритм метода PrivateCall класса Object

Функционал: Вызывает скрытый метод ChangePropsPrivate.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода PrivateCall класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вызов скрытого метода ChangePropsPrivate класса Object | ∅ |

## Алгоритм метода StatusOutput класса Object

Функционал: Вывод состояния - значения доступного и скрытого поля.

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода StatusOutput класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод строк "Value of the available property " со значением поля dataPublic и "; Value of a hidden property " со значением поля dataPrivate на экран | ∅ |

## Алгоритм метода ChangePropsPrivate класса Object

Функционал: Изменение значений доступного и скрытого поля.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

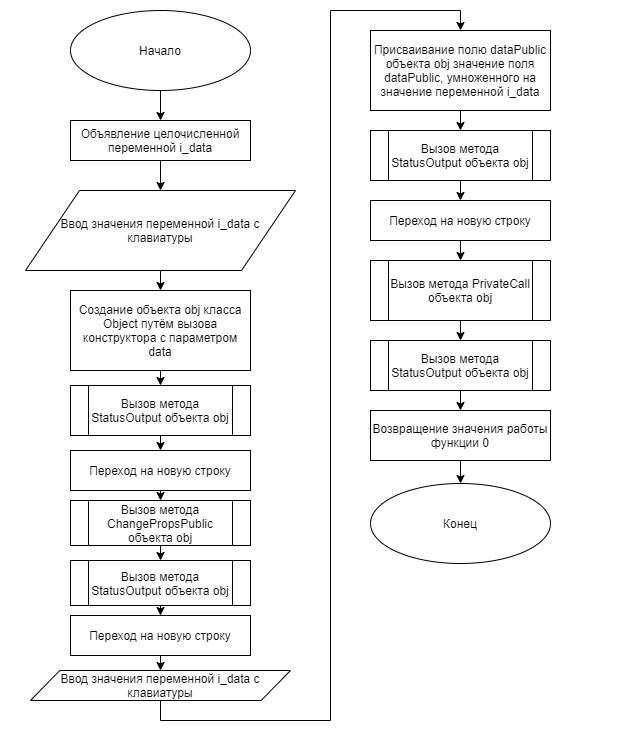
Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода ChangePropsPrivate класса Object

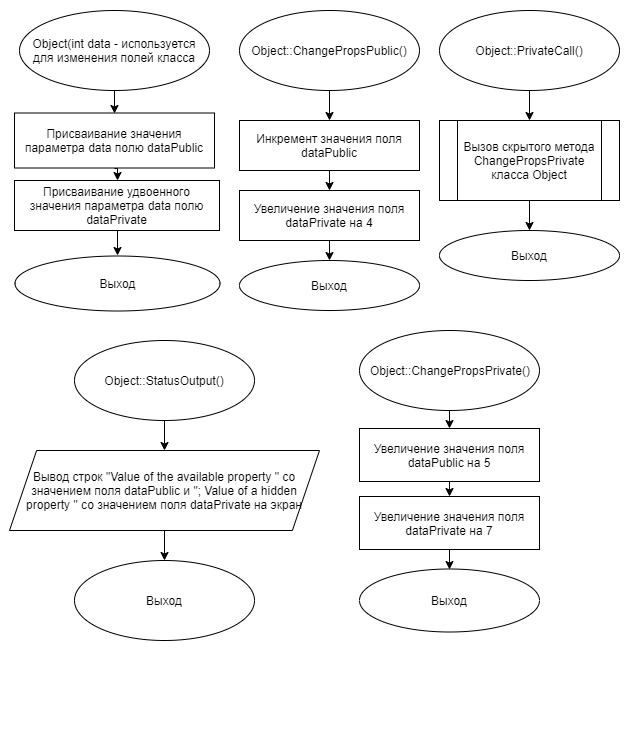
| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Увеличение значения поля dataPublic на 5 | 2 |
| 2 |  | Увеличение значения поля dataPrivate на 7 | ∅ |

# 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.

## **Файл** **main.cpp**

Листинг 1 – main.cpp

|  |
| --- |
| #include <iostream>  #include "Object.h"  using namespace std;  int main()  {  int i\_data;  cin >> i\_data;  Object obj(i\_data);  obj.StatusOutput();  cout << endl;  obj.ChangePropsPublic();  obj.StatusOutput();  cout << endl;  cin >> i\_data;  obj.dataPublic \*= i\_data;  obj.StatusOutput();  cout << endl;  obj.PrivateCall();  obj.StatusOutput();  return(0);  } |

## **Файл** **Object.cpp**

Листинг 2 – Object.cpp

|  |
| --- |
| #include <iostream>  #include "Object.h"  Object::Object(int data)  {  dataPublic = data;  dataPrivate = 2 \* data;  }  void Object::ChangePropsPublic()  {  dataPublic++;  dataPrivate += 4;  }  void Object::PrivateCall()  {  ChangePropsPrivate();  }  void Object::StatusOutput()  {  std::cout << "Value of the available property " << dataPublic  << "; Value of a hidden property " << dataPrivate;  }  void Object::ChangePropsPrivate()  {  dataPublic += 5;  dataPrivate += 7;  } |

## **Файл** **Object.h**

Листинг 3 – Object.h

|  |
| --- |
| #ifndef OBJECT\_H  #define OBJECT\_H  class Object  {  public:  int dataPublic;  Object(int data);  void ChangePropsPublic();  void PrivateCall();  void StatusOutput();  private:  int dataPrivate;  void ChangePropsPrivate();  };  #endif |

# 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
| --- | --- | --- |
| 2  4 | Value of the available property 2; Value of a hidden property 4  Value of the available property 3; Value of a hidden property 8  Value of the available property 12; Value of a hidden property 8  Value of the available property 17; Value of a hidden property 15 | Value of the available property 2; Value of a hidden property 4  Value of the available property 3; Value of a hidden property 8  Value of the available property 12; Value of a hidden property 8  Value of the available property 17; Value of a hidden property 15 |
| 3  5 | Value of the available property 3; Value of a hidden property 6  Value of the available property 4; Value of a hidden property 10  Value of the available property 20; Value of a hidden property 10  Value of the available property 25; Value of a hidden property 17 | Value of the available property 3; Value of a hidden property 6  Value of the available property 4; Value of a hidden property 10  Value of the available property 20; Value of a hidden property 10  Value of the available property 25; Value of a hidden property 17 |
| 4  6 | Value of the available property 4; Value of a hidden property 8  Value of the available property 5; Value of a hidden property 12  Value of the available property 30; Value of a hidden property 12  Value of the available property 35; Value of a hidden property 19 | Value of the available property 4; Value of a hidden property 8  Value of the available property 5; Value of a hidden property 12  Value of the available property 30; Value of a hidden property 12  Value of the available property 35; Value of a hidden property 19 |
| 5  7 | Value of the available property 5; Value of a hidden property 10  Value of the available property 6; Value of a hidden property 14  Value of the available property 42; Value of a hidden property 14  Value of the available property 47; Value of a hidden property 21 | Value of the available property 5; Value of a hidden property 10  Value of the available property 6; Value of a hidden property 14  Value of the available property 42; Value of a hidden property 14  Value of the available property 47; Value of a hidden property 21 |
| 6  8 | Value of the available property 6; Value of a hidden property 12  Value of the available property 7; Value of a hidden property 16  Value of the available property 56; Value of a hidden property 16  Value of the available property 61; Value of a hidden property 23 | Value of the available property 6; Value of a hidden property 12  Value of the available property 7; Value of a hidden property 16  Value of the available property 56; Value of a hidden property 16  Value of the available property 61; Value of a hidden property 23 |

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.

3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe\_posobie\_dlya\_laboratorny h\_rabot\_3.pdf (дата обращения 05.05.2021).

4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye\_k\_methodichke.pdf (дата обращения 05.05.2021).

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).