

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 конструктора класса Obekt.....	9
3.3 Алгоритм метода ChangePu класса Obekt.....	9
3.4 Алгоритм метода Private класса Obekt.....	10
3.5 Алгоритм метода ChangePr класса Obekt.....	10
3.6 Алгоритм метода Output класса Obekt.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл Obekt.cpp.....	14
5.3 Файл Obekt.h.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18

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

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, 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	3
3		Создание объекта obj класса Obejt	4
4		Вызов метода Output класса obj	5
5		Переход на новую строку	6
6		Вызов метода ChangePu объекта obj	7
7		Вызов метода Output объекта obj	8
8		Переход на новую строку	9
9		Ввод значения переменной i_data	10
10	Значение переменной i_data больше значения поля dataPu объекта obj	Полю dataPu объекта obj присваивается значения переменной i_data, умноженной на 8	11
			11
11		Вызов метода Output объекта obj	12

№	Предикат	Действия	№ перехода
12		Переход на новую строку	13
13		Вызов метода Private объекта obj	14
14		Вызов метода Output объекта obj	∅

3.2 Алгоритм конструктора класса *Obekt*

Функционал: Создание класс с именем *Obekt*.

Параметры: целое, изменение полей класса.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *Obekt*

№	Предикат	Действия	№ перехода
1		Полю dataPu присвоить значение параметра data	2
2		Полю dataPr присвоить значение параметра data умноженное на 3	∅

3.3 Алгоритм метода *ChangePu* класса *Obekt*

Функционал: Изменение значений доступного и скрвтого поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *ChangePu* класса *Obekt*

№	Предикат	Действия	№ перехода
1		Значение поля dataPu увеличивается на 4	2
2		Инкремент dataPr	∅

3.4 Алгоритм метода Private класса Obekt

Функционал: Вызывает скрытый метод ChangePr.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода Private класса Obekt

№	Предикат	Действия	№ перехода
1		Вызов скрытого метода ChangePr класса Obekt	Ø

3.5 Алгоритм метода ChangePr класса Obekt

Функционал: Изменение значений доступного и скрытого поля.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода ChangePr класса Obekt

№	Предикат	Действия	№ перехода
1		Значение поля dataPu увеличивается на 7	2
2		Значение поля dataPr увеличивается на 5	Ø

3.6 Алгоритм метода Output класса Obekt

Функционал: Вывод значения поля.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода Output класса Objekt

№	Предикат	Действия	№ перехода
1		Вывод строки "Value of the available property " со значением dataPu и "; Value of a hidden property" со значением dataPr	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

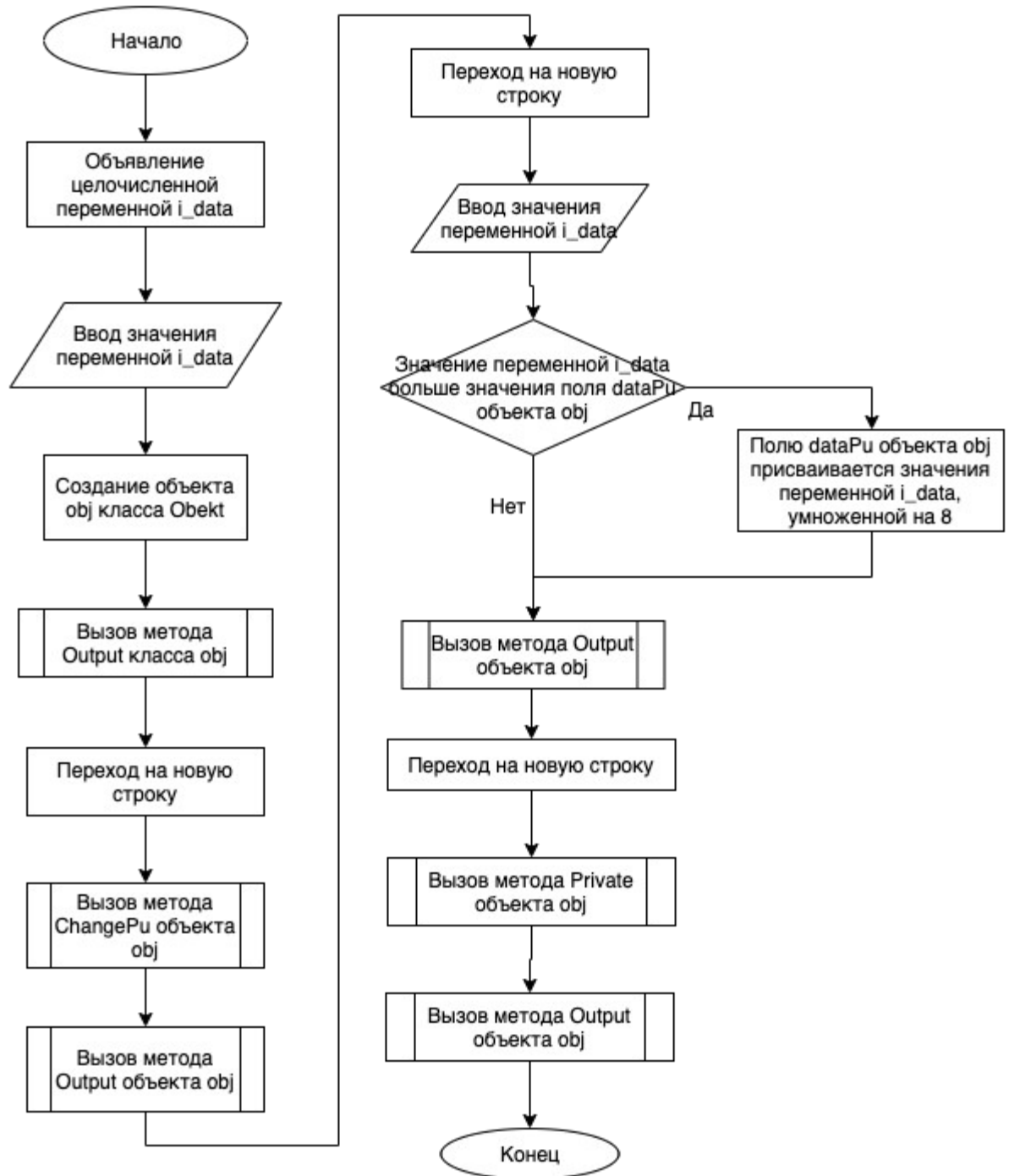


Рисунок 1 – Блок-схема алгоритма

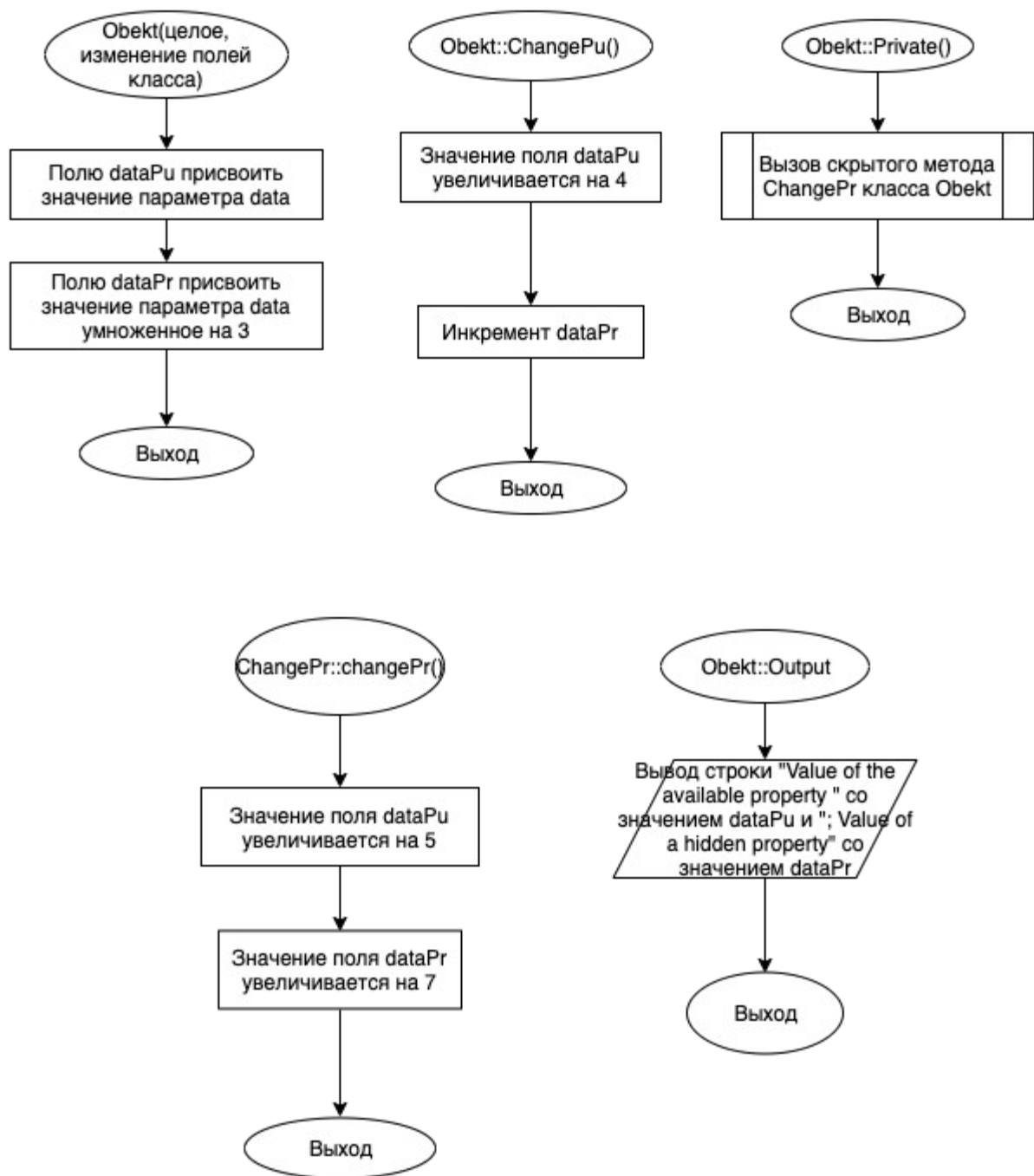


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "Obekt.h"
#include <iostream>
using namespace std;
int main()
{
    int i_data;
    cin>>i_data;
    Obekt *obj=new Obekt(i_data);
    obj->Output();
    cout<<endl;
    obj->ChangePu();
    obj->Output();
    cout<<endl;
    cin>>i_data;
    if (i_data>obj->dataPu)
    {
        obj->dataPu=i_data*8;
    }
    obj->Output();
    cout<<endl;
    obj->Private();
    obj->Output();
    delete obj;
    return(0);
}
```

5.2 Файл Obekt.cpp

Листинг 2 – Obekt.cpp

```
#include "Obekt.h"
#include <iostream>
using namespace std;
Obekt::Obekt(int data)
```

```

{
    dataPu=data;
    dataPr=3*data;
}
void Obekt::ChangePu()
{
    dataPu+=4;
    dataPr++;
}
void Obekt::Private()
{
    ChangePr();
}
void Obekt::Output()
{
    cout<<"Value of the available property "<<dataPu<< "; Value of a hidden
property "<<dataPr;
}
void Obekt::ChangePr()
{
    dataPu+=7;
    dataPr+=5;
}

```

5.3 Файл Obekt.h

Листинг 3 – Obekt.h

```

#ifndef __OBEKT__H
#define __OBEKT__H
class Obekt{
public:
    int dataPu;
    Obekt (int data);
    void ChangePu();
    void Private();
    void Output();
private:
    int dataPr;
    void ChangePr();
};
#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 9	Value of the available property 4; Value of a hidden property 12 Value of the available property 8; Value of a hidden property 13 Value of the available property 72; Value of a hidden property 13 Value of the available property 79; Value of a hidden property 18	Value of the available property 4; Value of a hidden property 12 Value of the available property 8; Value of a hidden property 13 Value of the available property 72; Value of a hidden property 13 Value of the available property 79; Value of a hidden property 18
3 1	Value of the available property 3; Value of a hidden property 9 Value of the available property 7; Value of a hidden property 10 Value of the available property 7; Value of a hidden property 10 Value of the available property 14; Value of a hidden property 15	Value of the available property 3; Value of a hidden property 9 Value of the available property 7; Value of a hidden property 10 Value of the available property 7; Value of a hidden property 10 Value of the available property 14; Value of a hidden property 15
4 6	Value of the available property 4; Value of a hidden property 12 Value of the available property 8; Value of a hidden property 13 Value of the available property 8; Value of a hidden	Value of the available property 4; Value of a hidden property 12 Value of the available property 8; Value of a hidden property 13 Value of the available property 8; Value of a hidden

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	property 13 Value of the available property 15; Value of a hidden property 18	property 13 Value of the available property 15; Value of a hidden property 18
9 2	Value of the available property 9; Value of a hidden property 27 Value of the available property 13; Value of a hidden property 28 Value of the available property 13; Value of a hidden property 28 Value of the available property 20; Value of a hidden property 33	Value of the available property 9; Value of a hidden property 27 Value of the available property 13; Value of a hidden property 28 Value of the available property 13; Value of a hidden property 28 Value of the available property 20; Value of a hidden property 33

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).