

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 функции main.....	9
3.2 Алгоритм метода ChangePu класса Obekt.....	10
3.3 Алгоритм конструктора класса Obekt.....	10
3.4 Алгоритм метода Output класса Obekt.....	10
3.5 Алгоритм метода ChangePr класса Obekt.....	11
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 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:

- функция `main` для главная функция программы;
- библиотека `iostream`;
- пространство имён `std`;
- объект `cout` стандартного потока вывода данных;
- объект `cin` стандартного потока ввода данных.

Класс `Obekt`:

- свойства/поля:
 - поле `dataPu`, ответственное за хранение значения:
 - наименование — `dataPu`;
 - тип — `int`;
 - модификатор доступа — `public`;
 - поле `dataPr`, ответственное за хранение значения:
 - наименование — `dataPr`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `Obekt` — параметизированный конструктор, который создаёт объект класса и изменяет значения полей с помощью параметра;
 - метод `ChangePu` — доступный метод, изменение значений доступного и скрытого поля;
 - метод `ChangePr` — скрытый метод, изменение значений доступного и скрытого поля;
 - метод `Output` — вывод состояния - значения скрытого и

доступного поля;

- о метод `Private` — вызов скрытого метода `ChangePr`.

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 <code>i_data</code>	2
2		Ввод значения переменной <code>i_data</code>	3
3		Создание объекта <code>obj</code> класса <code>Obekt</code>	4
4		Вызов метода <code>Output</code> объекта <code>obj</code>	5
5		Переход на новую строку	6
6		Вызов метода <code>ChangePu</code> объекта <code>obj</code>	7
7		Вызов метода <code>Output</code> объекта <code>obj</code>	8
8		Переход на новую строку	9
9		Ввод значения переменной <code>i_data</code>	10
10		Полю <code>dataPu</code> объекта <code>obj</code> присваивается значения поля <code>dataPu</code> умноженное на значение <code>i_data</code>	11
11		Вызов метода <code>Output</code> объекта <code>obj</code>	12
12		Переход на новую строку	13

№	Предикат	Действия	№ перехода
13		Вызов метода Private объекта obj	∅

3.2 Алгоритм метода ChangePu класса Obekt

Функционал: изменение значений скрытого и доступного поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода ChangePu класса Obekt

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

3.3 Алгоритм конструктора класса Obekt

Функционал: Создание объекта класса с именем Obekt.

Параметры: int data - для изменения полей класса.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Obekt

№	Предикат	Действия	№ перехода
1		Полю dataPu присвоить значение параметра data	2
2		Полю dataPr присвоить удвоенное значение параметра data	∅

3.4 Алгоритм метода Output класса Obekt

Функционал: Вывод значений поля.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *Output* класса *Obekt*

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

3.5 Алгоритм метода *ChangePr* класса *Obekt*

Функционал: Изменение значений 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 увеличить на 5	2
2		Значение поля dataPr увеличить на 5	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

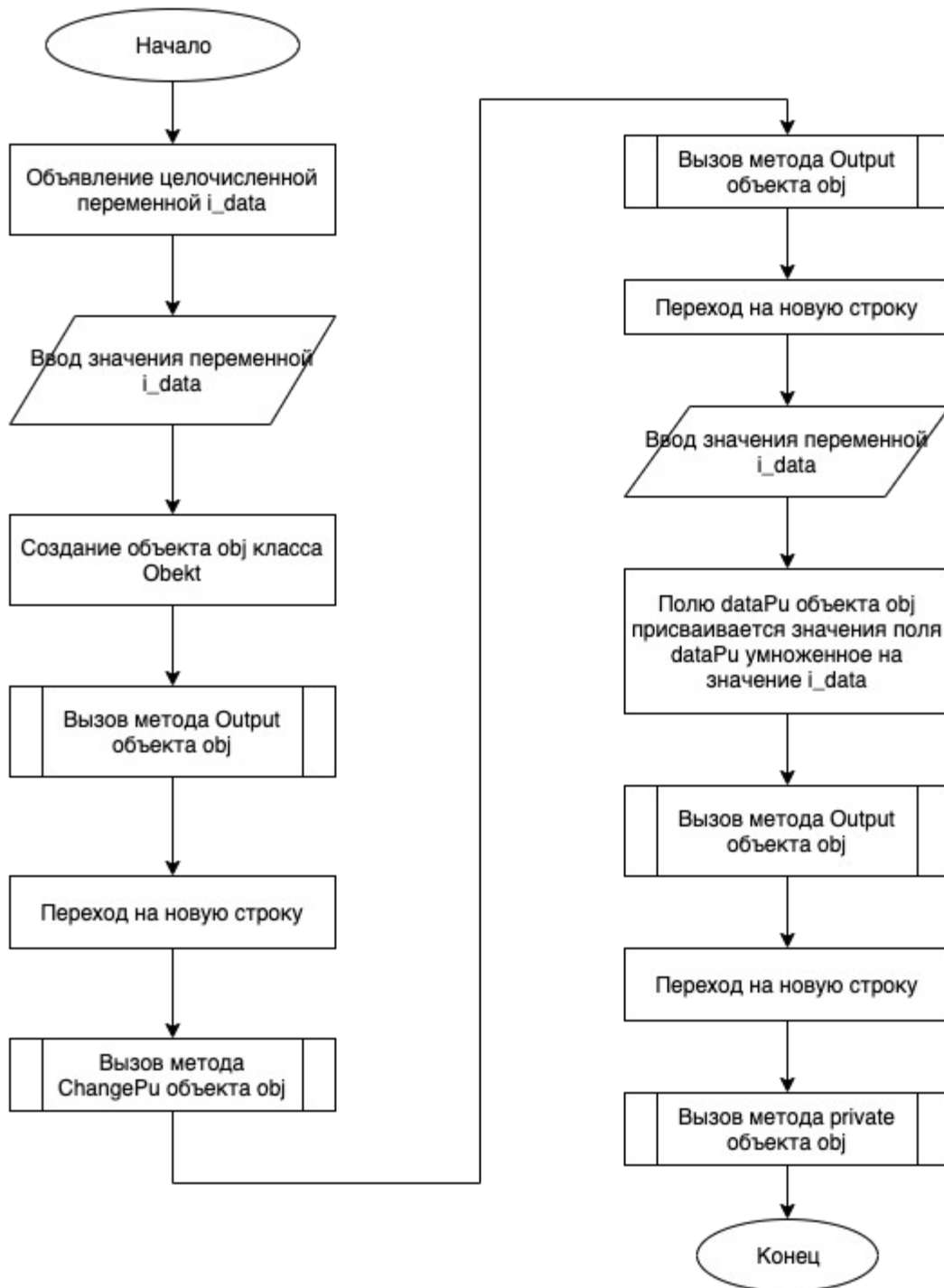


Рисунок 1 – Блок-схема алгоритма

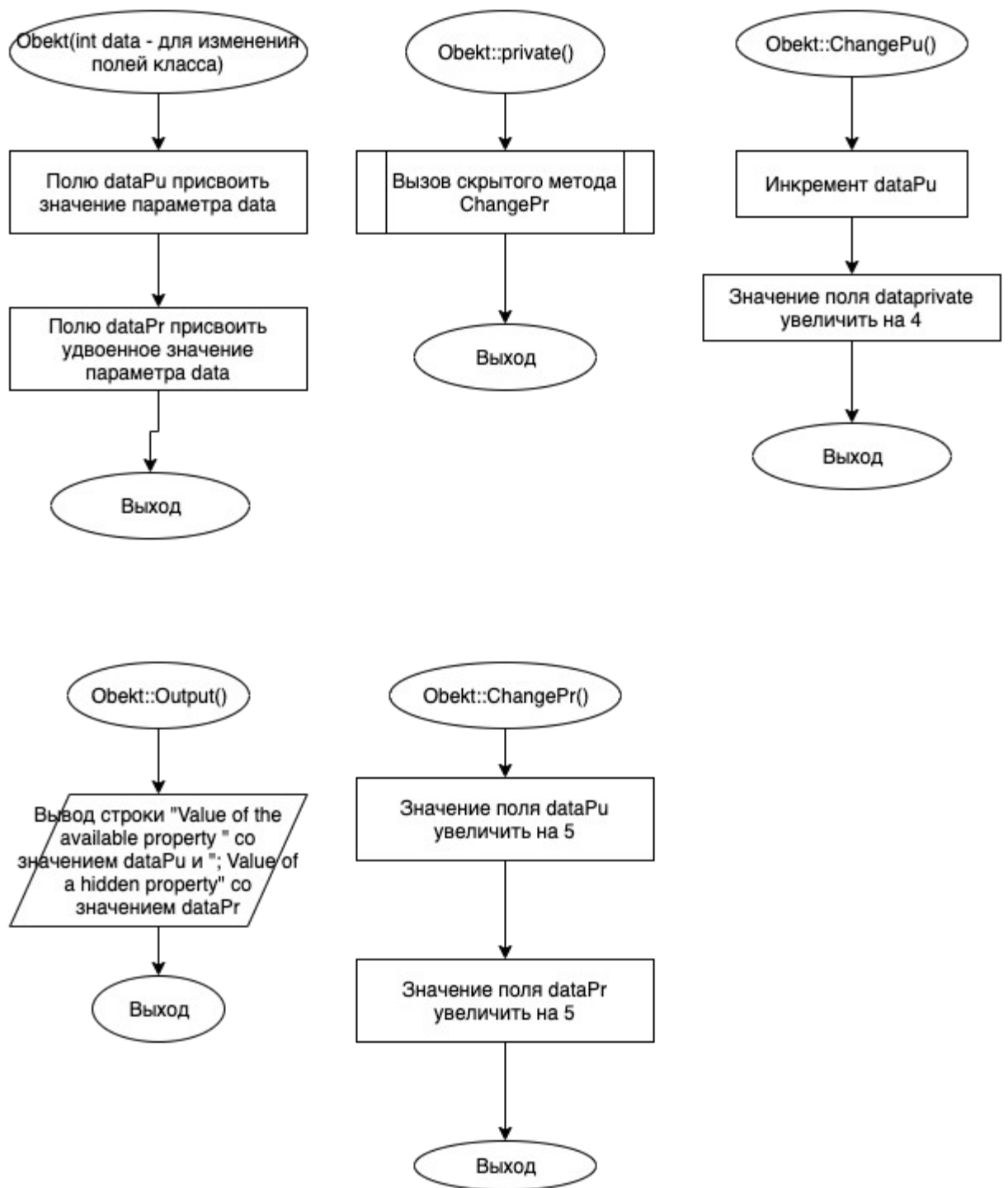


Рисунок 2 – Блок-схема алгоритма

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(i_data);
    obj.Output();
    cout<<endl;
    obj.ChangePu();
    obj.Output();
    cout<<endl;
    cin>>i_data;
    obj.dataPu*=i_data;
    obj.Output();
    cout<<endl;
    obj.Private();
    obj.Output();
    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=2*data;
}
```

```

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

```

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 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 8	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 48; Value of a hidden property 14 Value of the available property 53; 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 48; Value of a hidden property 14 Value of the available property 53; Value of a hidden property 21
3 9	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 36; Value of a hidden property 10 Value of the available property 41; 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 36; Value of a hidden property 10 Value of the available property 41; Value of a hidden property 17
1 5	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 10; Value of a	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 10; Value of a

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	hidden property 6 Value of the available property 15; Value of a hidden property 13	hidden property 6 Value of the available property 15; Value of a hidden property 13
4 2	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 10; Value of a hidden property 12 Value of the available property 15; 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 10; Value of a hidden property 12 Value of the available property 15; Value of a hidden property 19

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).