

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	11
3.1 Алгоритм конструктора класса OuterClass.....	11
3.2 Алгоритм деструктора класса OuterClass.....	11
3.3 Алгоритм метода sumToOtherObj класса OuterClass.....	11
3.4 Алгоритм метода setOtherObj класса OuterClass.....	12
3.5 Алгоритм метода sumValues класса InnerClass.....	12
3.6 Алгоритм деструктора класса InnerClass.....	13
3.7 Алгоритм функции main.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	15
5 КОД ПРОГРАММЫ.....	20
5.1 Файл InnerClass.cpp.....	20
5.2 Файл InnerClass.h.....	20
5.3 Файл main.cpp.....	21
5.4 Файл OuterClass.cpp.....	21
5.5 Файл OuterClass.h.....	22
6 ТЕСТИРОВАНИЕ.....	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	24

1 ПОСТАНОВКА ЗАДАЧИ

Разработать систему, которая демонстрирует возможность размещения и использования объекта одного класса в составе объекта другого класса. При этом, жизненный цикл внутреннего объекта полностью реализуется в рамках внешнего объекта.

Спроектировать внешний объект, который содержит внутренний объект в закрытом доступе. Содержит свойство строкового типа для хранения наименования объекта в закрытом доступе.

Внутренний объект имеет одно целочисленное свойство, в закрытом доступе, для хранения суммы полученных целочисленных значений.

Внутренний объект имеет метод в открытом доступе, с одним целочисленным параметром, который суммирует полученные по параметру значения.

В деструкторе внутреннего объекта выводиться значение полученной в процессе функционирования объекта суммы.

Внешний объект имеет параметризованный конструктор, с параметром строкового типа, который передает наименование объекта.

Внешний объект имеет метод с целочисленным параметром в открытом доступе, который передает значение этого параметра другому внешнему объекту посредством вызова метода и последующего суммирования.

Внешний объект имеет деструктор, который выводит "Object_" и наименование объекта.

Организовать связи между объектами.

Для организации связи между объектами добавить не более одного метода в открытом доступе к внешнему объекту, не более одного свойства и одного метода в открытом доступе внутреннему объекту,

Алгоритм конструирования и отработки системы:

1. Объявляется строковая переменная.
2. Вводится наименование первого внешнего объекта.
3. Объявляется первый объект внешнего класса, с использованием параметризованного конструктора.
4. Вводится наименование второго внешнего объекта.
5. Объявляется второй объект внешнего класса, с использованием параметризованного конструктора.
6. Организуется связь между объектами.
7. Объявляется целочисленная переменная, для хранения номера внешнего объекта.
8. Объявляется целочисленная переменная, для хранения целочисленного значения.
9. Начало цикла.
 - 9.1. Вводится номер внешнего объекта и целочисленное значение.
 - 9.2. Если номер внешнего объекта равно 0, работа цикла завершается.
 - 9.3. Вызывается метод внешнего объекта согласно введенному номеру и передается в качестве аргумента переменная с целочисленным значением, которое посредством организованной связи передается другому внешнему объекту.
10. Конец цикла.
11. Завершается работа системы.

1.1 Описание входных данных

Первая строка:

«наименование первого объекта»

Вторая строка:

«наименование второго объекта»

Начиная с третьей строки, построчно:

«целое число, номер объекта» «целое число»

Последняя строка:

0

Пример ввода:

```
ffff
ssss
1 1
2 2
2 2
1 4
0
```

1.2 Описание выходных данных

Первая строка:

Object_«наименование второго объекта» SUMM = «целое число, значение суммы»

Вторая строка:

Object_«наименование первого объекта» SUMM = «целое число, значение суммы»

Курсор устанавливается в начале третьей строки.

Пример вывода:

```
Object_ssss      SUMM = 5
Object_ffff      SUMM = 4
```


2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj1 класса OuterClass предназначен для ;
- объект obj2 класса OuterClass предназначен для ;
- библиотека iostream;
- арифметические операторы;
- условная конструкция if...else;
- цикл с предусловием while.

Класс OuterClass:

- свойства/поля:
 - поле объект вложенного класса innerClass:
 - наименование — innerObj;
 - тип — InnerClass;
 - модификатор доступа — private;
 - поле название:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
 - поле указатель на другой объект:
 - наименование — otherObj;
 - тип — OuterClass*;
 - модификатор доступа — private;
- функционал:
 - метод OuterClass — конструктор с параметром;
 - метод ~OuterClass — деструктор;
 - метод sumToOtherObj — прибавление числа к счетчику другого

объекта;

- о метод `setOtherObj` — установление указателя на другой объект.

Класс `InnerClass`:

- свойства/поля:
 - о поле `сумма`:
 - наименование — `sum`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - о метод `sumValues` — прибавление параметра к `sum`;
 - о метод `~InnerClass` — деструктор.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	OuterClass			Внешний класс	
2	InnerClass			Внутренний класс	

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 OuterClass

Функционал: конструктор с параметром.

Параметры: name.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса OuterClass

№	Предикат	Действия	№ перехода
1		Присвоение полю name значения параметра name	Ø

3.2 Алгоритм деструктора класса OuterClass

Функционал: деструктор.

Параметры: name.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса OuterClass

№	Предикат	Действия	№ перехода
1		Вывод строки "Object_", name, " "	Ø

3.3 Алгоритм метода sumToOtherObj класса OuterClass

Функционал: прибавление числа к счетчику другого объекта.

Параметры: целочисленный параметр value.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *sumToOtherObj* класса *OuterClass*

№	Предикат	Действия	№ перехода
1		Вызов метода <i>sumValues</i> у объекта <i>innerObj</i> , который является свойством объекта <i>otherObj</i> с передачей в него параметра <i>value</i>	Ø

3.4 Алгоритм метода *setOtherObj* класса *OuterClass*

Функционал: установление указателя на другой объект.

Параметры: *otherObj* типа *OuterClass**.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *setOtherObj* класса *OuterClass*

№	Предикат	Действия	№ перехода
1		Присвоение полю <i>otherObj</i> значения параметра <i>otherObj</i>	Ø

3.5 Алгоритм метода *sumValues* класса *InnerClass*

Функционал: прибавление параметра к *sum*.

Параметры: целочисленный параметр *value*.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода *sumValues* класса *InnerClass*

№	Предикат	Действия	№ перехода
1		Прибавление значения <i>value</i> к полю <i>sum</i>	Ø

3.6 Алгоритм деструктора класса InnerClass

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 7.

Таблица 7 – Алгоритм деструктора класса InnerClass

№	Предикат	Действия	№ перехода
1		Вывод строки "SUMM = ", sum и переход на новую строку	Ø

3.7 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое число, идентификатор работоспособности программы.

Алгоритм функции представлен в таблице 8.

Таблица 8 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление переменной name типа string	2
2		Ввод значения name	3
3		Создание объекта класса OuterClass с передачей в конструктор значения name и инициализация переменной obj1 указателем на данный объект	4
4		Ввод значения name	5
5		Создание объекта класса OuterClass с передачей в конструктор значения name и инициализация переменной obj2 указателем на данный объект	6
6		Вызов метода setOtherObj у объекта obj1 с	7

№	Предикат	Действия	№ перехода
		параметром obj2	
7		Вызов метода setOtherObj у объекта obj2 с параметром obj1	8
8		Объявление целочисленных переменных num и value	9
9		Ввод значений num и value	10
10	num == 0?		13
			11
11	num == 1?	Вызов метода sumToOtherObj у объекта obj2	9
			12
12	num == 2?	Вызов метода sumToOtherObj у объекта obj2	9
			9
13		Удаление объекта по указателю obj1	14
14		Удаление объекта по указателю obj2	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

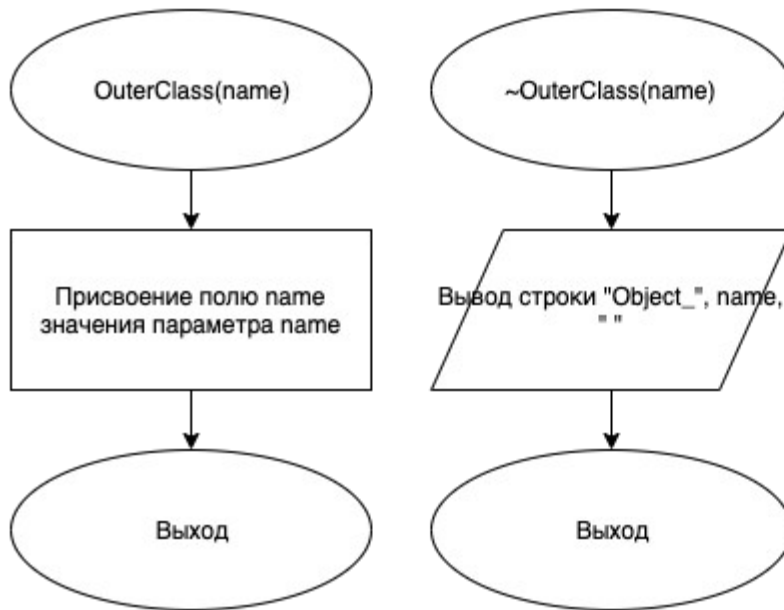


Рисунок 1 – Блок-схема алгоритма

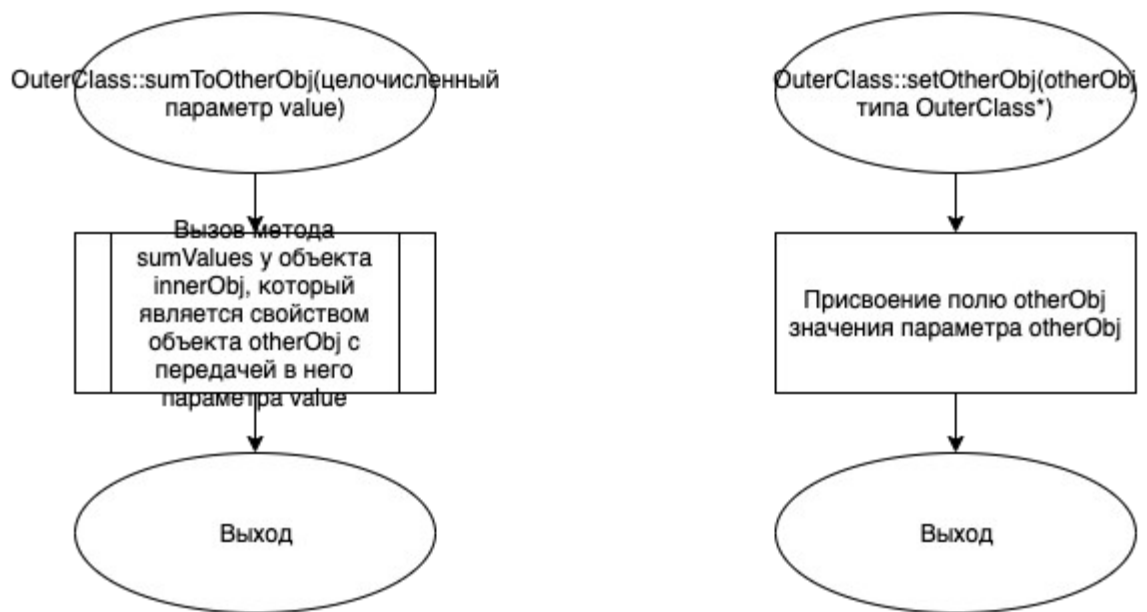


Рисунок 2 – Блок-схема алгоритма

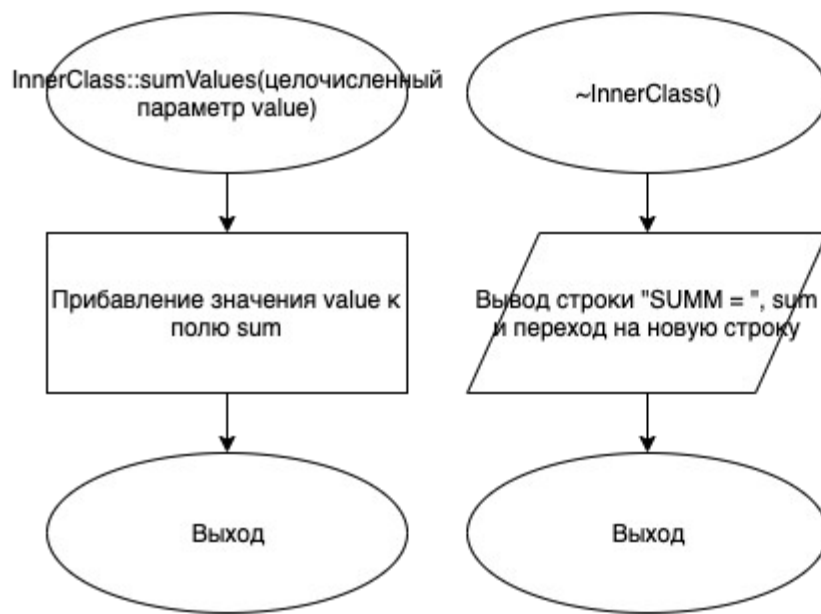


Рисунок 3 – Блок-схема алгоритма



Рисунок 4 – Блок-схема алгоритма

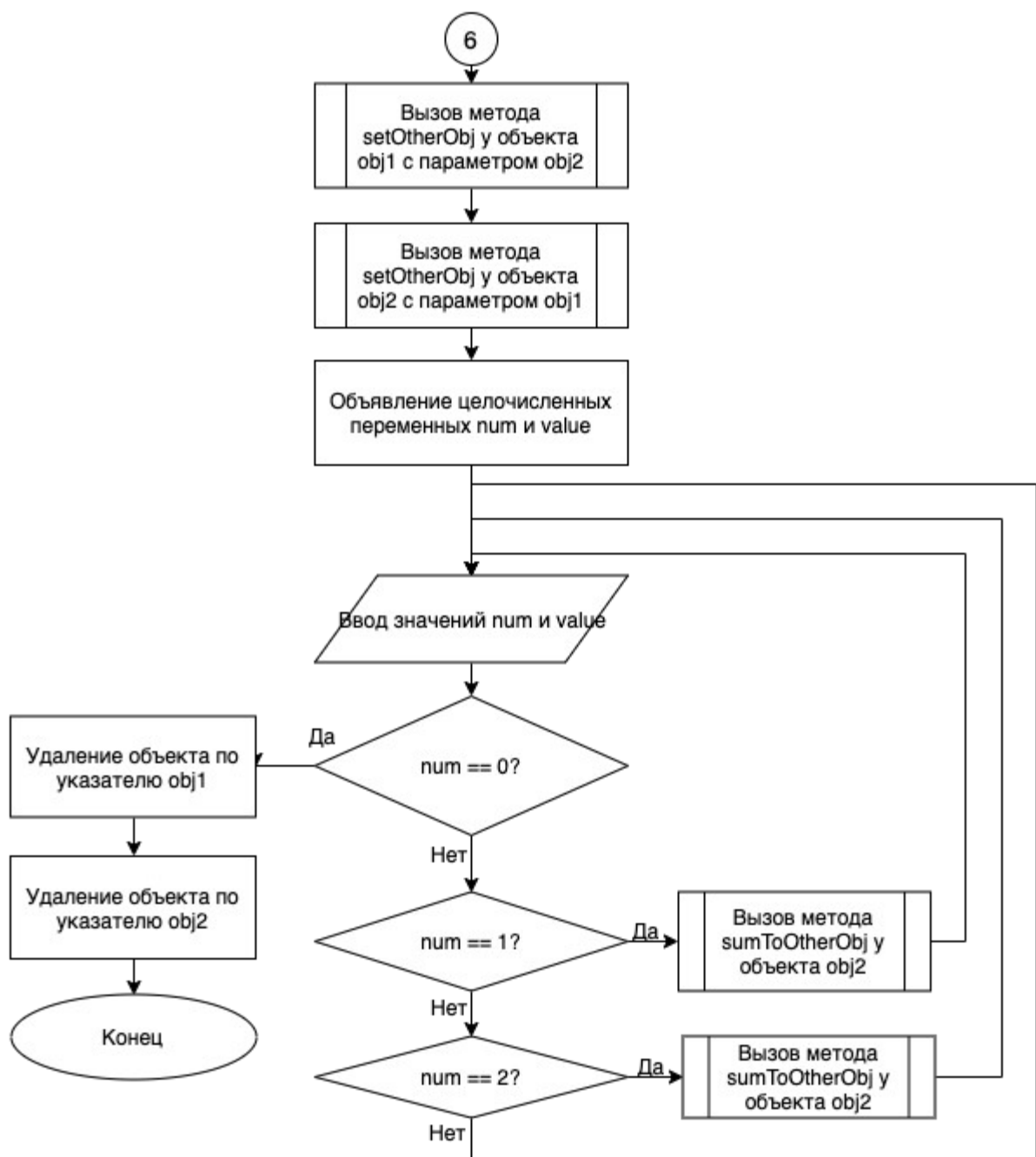


Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл InnerClass.cpp

Листинг 1 – InnerClass.cpp

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

InnerClass::~InnerClass(){
    cout << "SUMM = " << sum << endl;
}

void InnerClass::sumValues(int value){
    sum += value;
}
```

5.2 Файл InnerClass.h

Листинг 2 – InnerClass.h

```
#ifndef __INNERCLASS__H
#define __INNERCLASS__H
#include <iostream>
using namespace std;

class InnerClass{
    int sum;
public:
    void sumValues(int value);
    ~InnerClass();
};

#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

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

int main()
{
    string name;
    cin >> name;
    OuterClass* obj1 = new OuterClass(name);
    cin >> name;
    OuterClass* obj2 = new OuterClass(name);

    obj1->setOtherObj(obj2);
    obj2->setOtherObj(obj1);

    int num;
    int value;
    while (true) {
        cin >> num >> value;
        if (num == 0){
            break;
        } else if (num == 1){
            obj1->sumToOtherObj(value);
        } else if (num == 2) {
            obj2->sumToOtherObj(value);
        }
    }
    delete obj2;
    delete obj1;
    return(0);
}
```

5.4 Файл OuterClass.cpp

Листинг 4 – OuterClass.cpp

```
#include "OuterClass.h"
#include "InnerClass.h"
#include <string>
#include <iostream>
using namespace std;

OuterClass::OuterClass(string name){
```

```

        this->name = name;
    }

    OuterClass::~~OuterClass(){
        cout << "Object_" << name << "    ";
    }

    void OuterClass::sumToOtherObj(int value){
        otherObj->innerObj.sumValues(value);
    }

    void OuterClass::setOtherObj(OuterClass* otherObj){
        this->otherObj = otherObj;
    }

```

5.5 Файл OuterClass.h

Листинг 5 – OuterClass.h

```

#ifndef __OUTERCLASS__H
#define __OUTERCLASS__H
#include <iostream>
#include <string>
#include "InnerClass.h"
using namespace std;

class OuterClass{
public:
    OuterClass(string name);
    ~OuterClass();

    void sumToOtherObj(int value);
    void setOtherObj(OuterClass* otherObj);
private:
    string name;
    OuterClass* otherObj;
    InnerClass innerObj;
};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 9.

Таблица 9 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
ffff ssss 1 1 2 2 2 2 1 4 0	Object_ssss SUMM = 5 Object_ffff SUMM = 4	Object_ssss SUMM = 5 Object_ffff SUMM = 4

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).