

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	9
3.1 Алгоритм функции main.....	9
3.2 Алгоритм конструктора класса cl1.....	9
3.3 Алгоритм метода input класса cl2.....	10
3.4 Алгоритм функции maxs.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	14
5.1 Файл cl1.cpp.....	14
5.2 Файл cl1.h.....	14
5.3 Файл cl2.cpp.....	15
5.4 Файл cl2.h.....	15
5.5 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18

# 1 ПОСТАНОВКА ЗАДАЧИ

Создать объект первого типа, у которого одно целочисленное свойство. Значение данного свойства определяется посредством параметризованного конструктора.

Создать объект второго типа, у которого две целочисленных свойства. Значение данных свойств определяется посредством метода объекта.

Реализовать дружественную функцию, которая находит максимальное значение полей объекта первого типа и полей объекта второго типа.

Написать программу:

1. Вводит значение для поля объекта первого типа.
2. Создает объект первого типа.
3. Вводит значения полей для полей объекта второго типа.
4. Создает объект второго типа.
5. Определяет значения полей объекта второго типа.
6. Определяет максимальное значение полей, созданных двух объектов разного типа посредством дружественной функции.
7. Выводит полученный результат.

## 1.1 Описание входных данных

**Первая строка:**

«целое число в десятичном формате»

**Вторая строка:**

«целое число в десятичном формате» „целое число в десятичном формате»

## **1.2 Описание выходных данных**

**Первая строка**, с первой позиции:

max = «целочисленное значение в десятичном формате»

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи понадобится:

дружественная функция

условный оператор if

2 объекта класса

указатель this

### Класс cl1

Поля:

скрытый элемент int N

Методы:

открытые:

конструктор cl1(int n) - значение данного свойства определяется посредством параметризованного конструктора.

дружественная функция friend int maxs(cl1 &object1, cl2 &object2) - находит максимальное значение полей объекта первого типа и полей объекта второго типа.

### Класс cl2

Поля:

скрытый элемент int num1, num2

Методы:

открытые:

input(int num1, int num2) - приравнивает значения в закрытые поля данных

дружественная функция friend int maxs(cl1 &object1, cl2 &object2) - находит

максимальное значение полей объекта первого типа и полей объекта второго типа.

## 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: главный метод программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		ввод значение для поля объекта первого типа	2
2		создание объекта первого типа	3
3		ввод значения полей для полей объекта второго типа	4
4		создание объекта второго типа	5
5		определение значения полей объекта второго типа	6
6		определение максимального значение полей, созданных двух объектов разного типа посредством дружественной функции	7
7		вывод полученной результат	Ø

### 3.2 Алгоритм конструктора класса cl1

Функционал: присвоение данного.

Параметры: n.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *cl1*

№	Предикат	Действия	№ перехода
1		присвоение значение параметризованного конструктора	Ø

### 3.3 Алгоритм метода *input* класса *cl2*

Функционал: присвоение данных.

Параметры: *num1*, *num2*.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *input* класса *cl2*

№	Предикат	Действия	№ перехода
1		присвоение значения	Ø

### 3.4 Алгоритм функции *maxs*

Функционал: нахождение максимального значения и вывод.

Параметры: *object1*, *object2*.

Возвращаемое значение: *int*.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *maxs*

№	Предикат	Действия	№ перехода
1	второе число больше третьего и второе число больше первого	вывод второго числа	Ø
	третье число больше второго и третье число больше первого	вывод третьего числа	Ø



№	Предикат	Действия	№ перехода
	первое число больше второго и первое число больше третьего числа	вывод первого числа	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

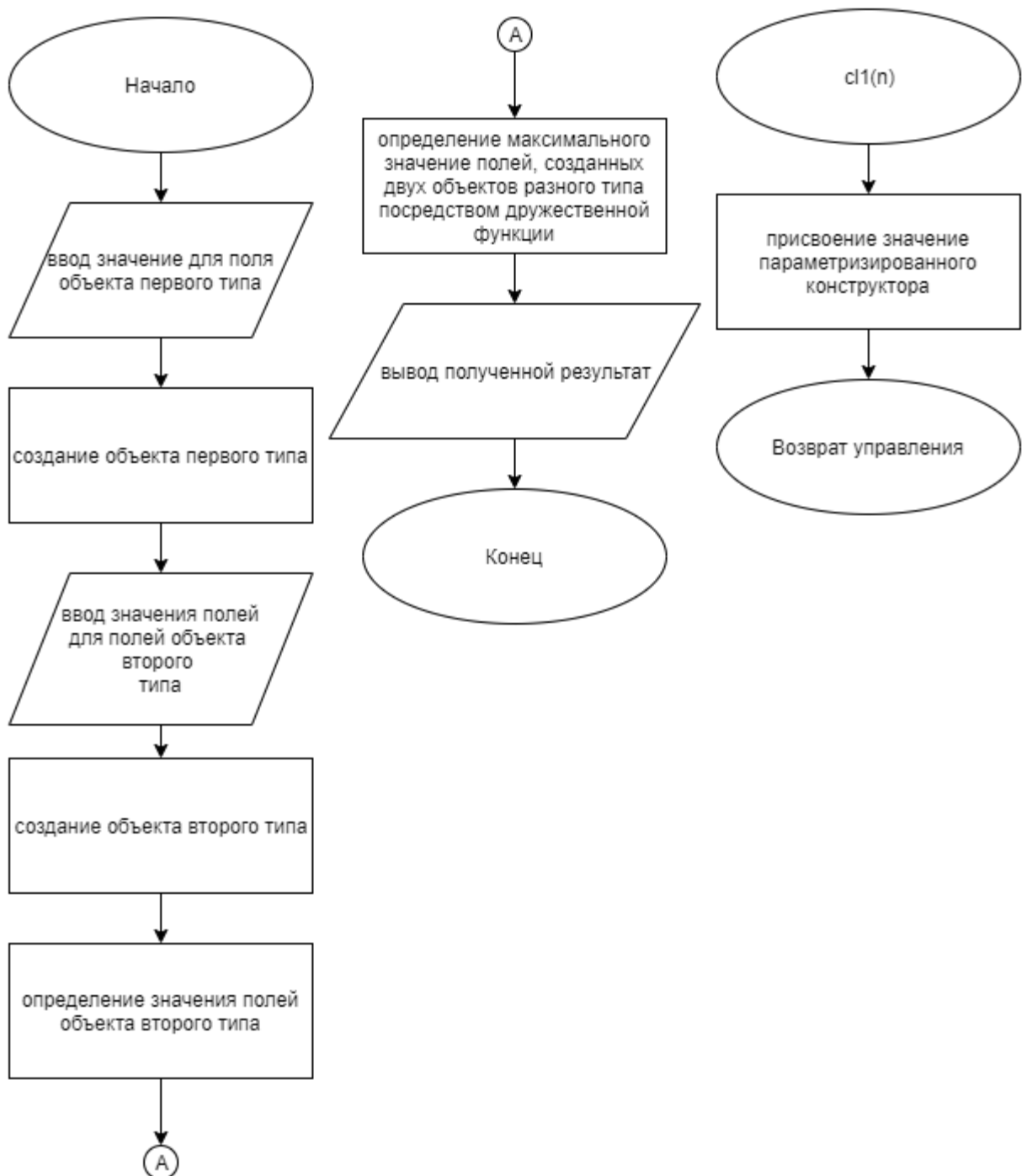
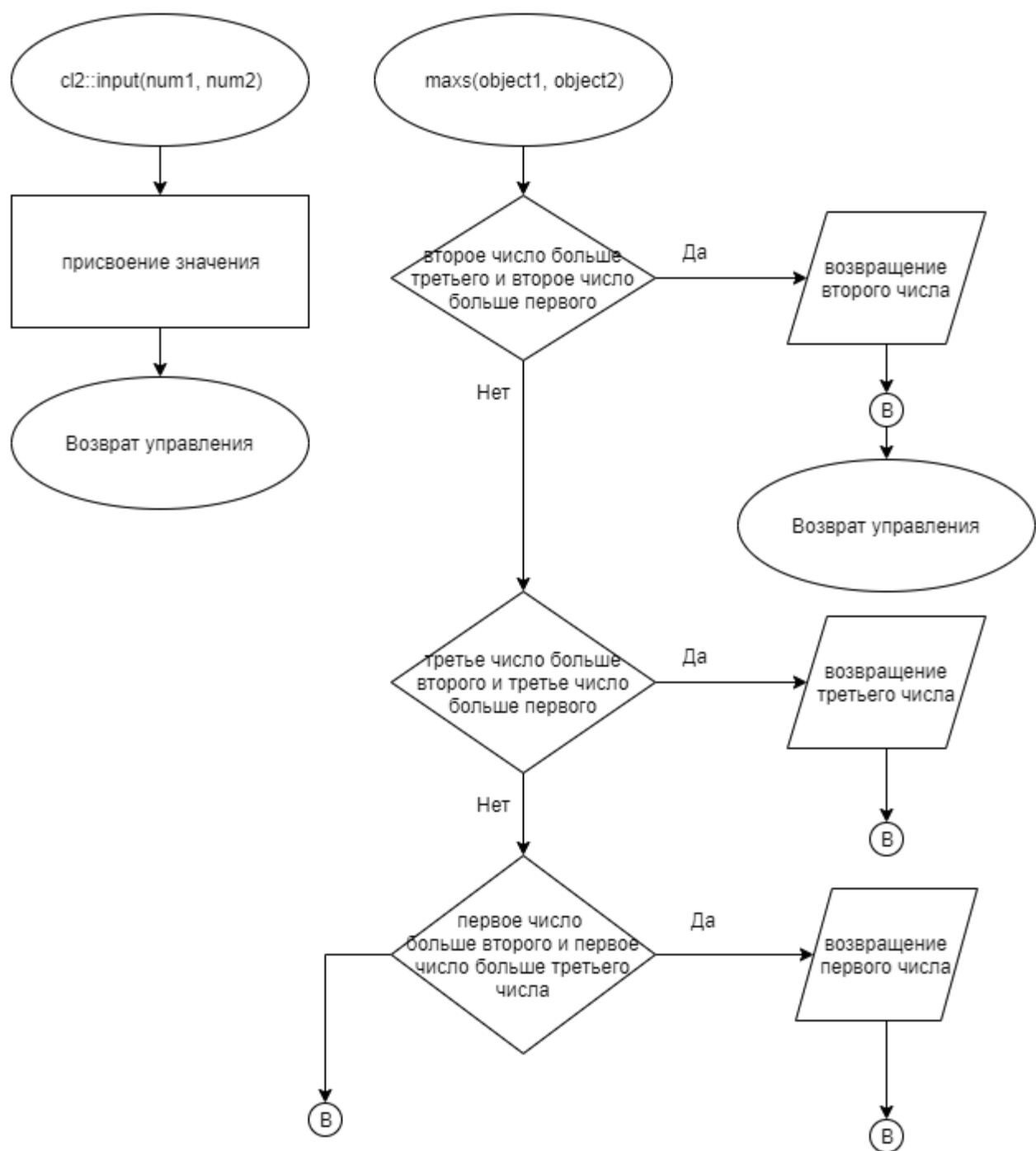


Рисунок 1 – Блок-схема алгоритма



**Рисунок 2 – Блок-схема алгоритма**

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл cl1.cpp

*Листинг 1 – cl1.cpp*

```
#include <iostream>
#include "cl1.h"
#include "cl2.h"

using namespace std;

cl1::cl1(int n)
{
    N = n;
}
```

### 5.2 Файл cl1.h

*Листинг 2 – cl1.h*

```
#ifndef __CL1_H
#define __CL1_H

class cl2;
class cl1
{
private:
    int N;
public:
    cl1(int n);
    friend int maxs(cl1 &object1, cl2 &object2);
};

#endif
```

## 5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

```
#include <iostream>
#include "cl2.h"
#include "cl1.h"

using namespace std;

void cl2::input(int num1, int num2)
{
    this->num1 = num1;
    this->num2 = num2;
}

int maxs(cl1 &object1, cl2 &object2)
{
    if (object2.num1 > object2.num2 && object2.num1 > object1.N)
    {
        return object2.num1;
    }
    else if (object2.num2 > object2.num1 && object2.num2 > object1.N)
    {
        return object2.num2;
    }
    else
    {
        return object1.N;
    }
}
```

## 5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2_H
#define __CL2_H

class cl1;
class cl2
{
private:
    int num1, num2;
public:
    void input(int num1, int num2);
    friend int maxs(cl1 &object1, cl2 &object2);
};

#endif
```

## 5.5 Файл main.cpp

*Листинг 5 – main.cpp*

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "cl1.h"
#include "cl2.h"

using namespace std;

int main()
{
    int n, num1, num2;
    cin >> n;
    cl1 object1(n);
    cin >> num1 >> num2;
    cl2 object2;
    object2.input(num1, num2);
    cout << "max = " << maxs(object1, object2);

    return(0);
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 2 0	max = 2	max = 2

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: 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-avrova.ru/student/files/methodicheskoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avrova.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_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-avrova.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avrova.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).