

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.....	8
3.1 Алгоритм конструктора класса cl1.....	8
3.2 Алгоритм метода set_num класса cl2.....	8
3.3 Алгоритм функции main.....	9
3.4 Алгоритм функции f.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	13
5.1 Файл cl1.....	13
5.2 Файл cl2.....	13
5.3 Файл main.cpp.....	14
6 ТЕСТИРОВАНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16

# 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 поставленной задачи используется:

условный оператор if

объект класса cl1

объект класса cl2

функция f(cl1& A, cl2& B) - является дружественной функцией для классов cl1 и cl2. Находит максимальное значение полей объекта класса cl1 и полей объекта класса cl2

объект потокового ввода cin

объект потокового вывода cout

оператор цикла for

**класс cl1:**

поля:

Методы:

Доступные:

конструктор cl1(int n) - параметризированный конструктор, определяющий значение скрытого свойства

**класс cl2:**

Методы:

Доступные:

set\_num(int num1, int num2) - определяет значение скрытых полей m1 и m2.

Поля:

## 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 *cl1*

Функционал: Определение значения свойства *n*.

Параметры: *int num*.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *cl1*

№	Предикат	Действия	№ перехода
1		Присвоение свойству <i>n</i> значения аргумента <i>num</i>	Ø

### 3.2 Алгоритм метода *set\_num* класса *cl2*

Функционал: Определение значения свойств *m1* и *m2*.

Параметры: *int num1*, *int num2*.

Возвращаемое значение: *void*.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *set\_num* класса *cl2*

№	Предикат	Действия	№ перехода
1		Присвоение свойствам <i>m1</i> , <i>m2</i> значений аргументов <i>num1</i> , <i>num2</i> соответственно	Ø

### 3.3 Алгоритм функции main

Функционал: Главный метод программы.

Параметры: .

Возвращаемое значение: void.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных n, m1, m2	2
2		Ввод целочисленных переменных n, m1, m2	3
3		Создание объекта класса cl1 с передаваемым аргументом n	4
4		Создание объекта класса cl2	5
5		Вызов метода set_num с передаваемыми ему аргументами m1, m2	6
6		Вывод результата работы функции f с передаваемыми ей в качестве аргументов элементами классов cl1 и cl2	∅

### 3.4 Алгоритм функции f

Функционал: Вычисление максимального значения полей элементов классов.

Параметры: cl1& A, cl2& B.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции f

№	Предикат	Действия	№ перехода
1		Присвоение целочисленной переменной max значения свойства n объекта A	2
2	Значение max <= значения	присвоение переменной max значения свойства m1	3

№	Предикат	Действия	№ перехода
	свойства m1 объекта B	объекта B	
			3
3	Значение max <= значения свойства m2 объекта B	присвоение переменной max значения свойства m2 объекта B	4
			4
4		Возврат переменной max	∅



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

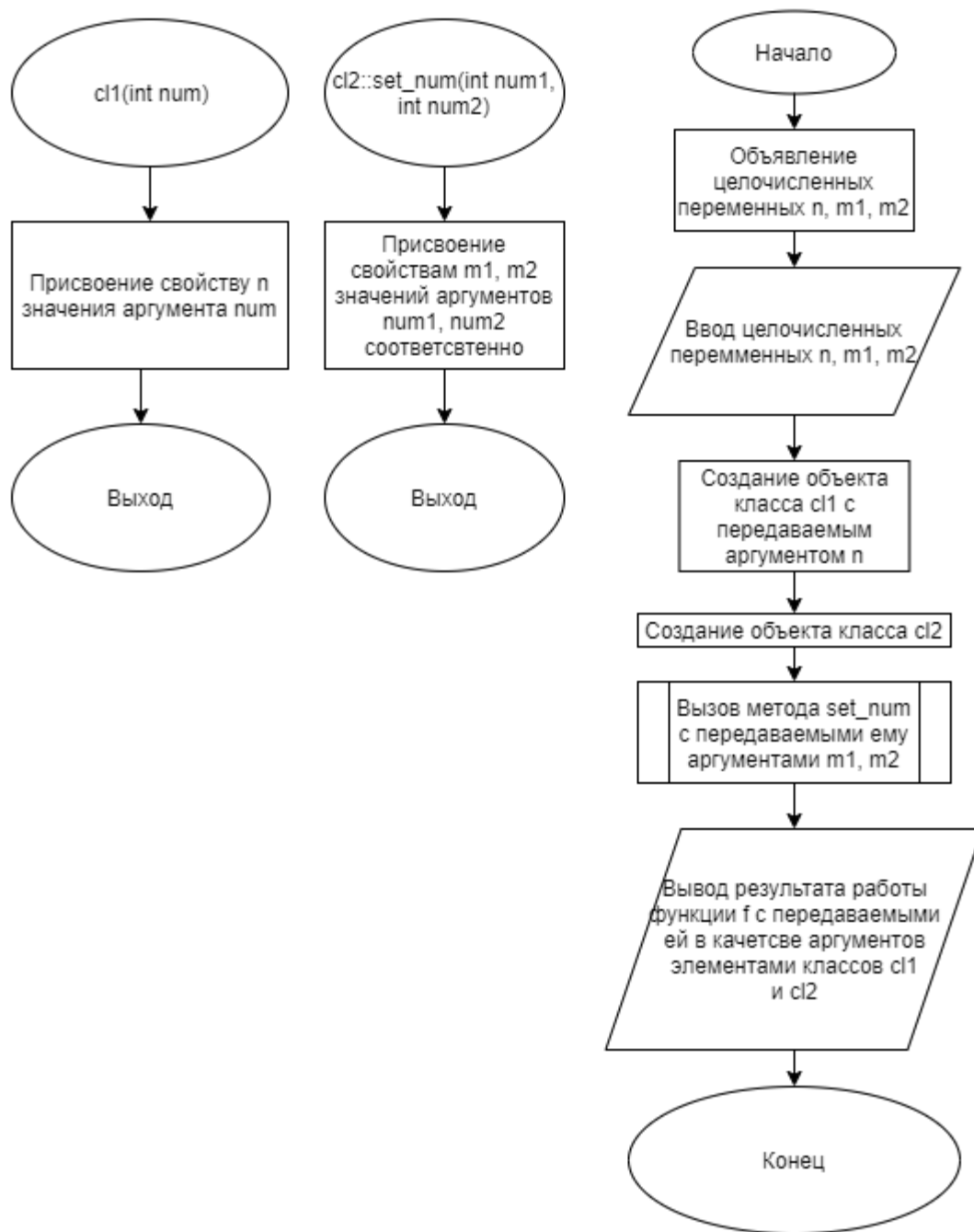


Рисунок 1 – Блок-схема алгоритма

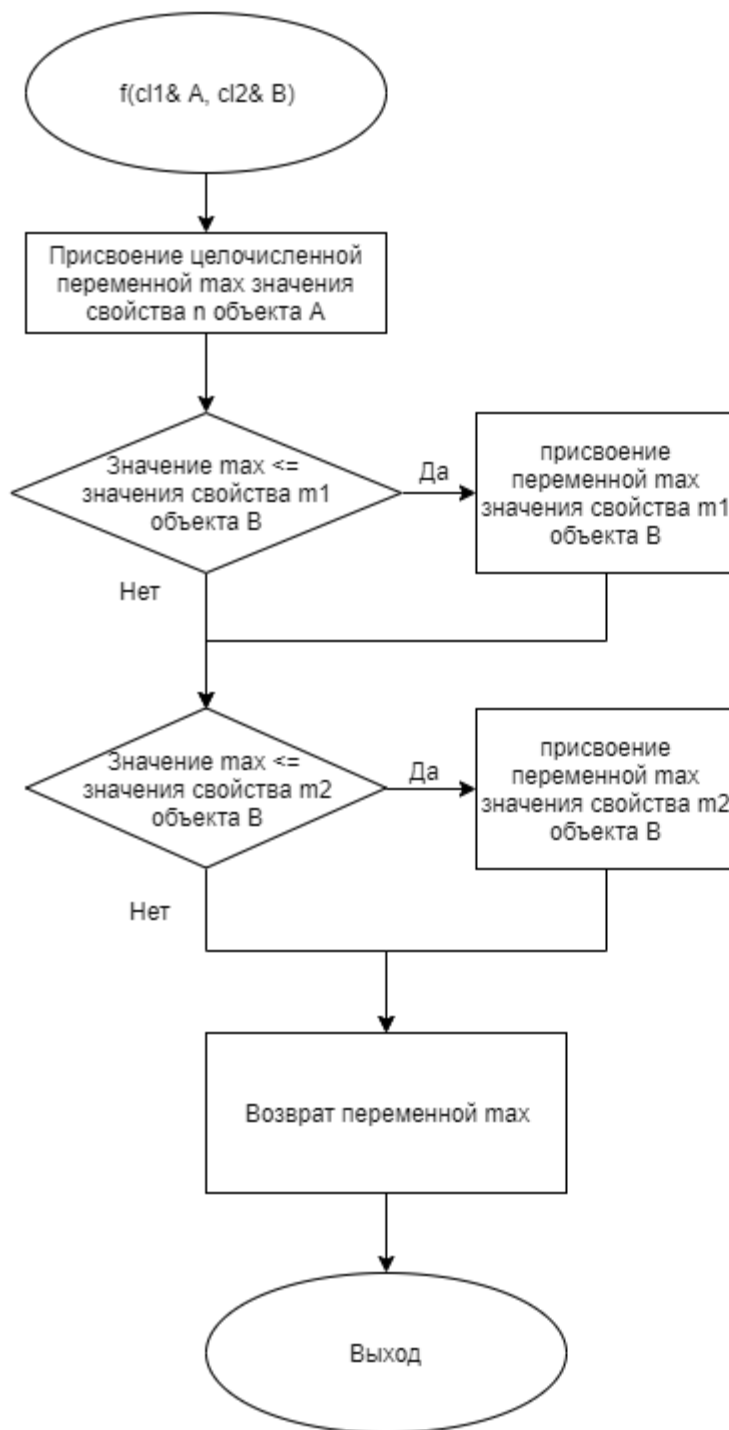


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл cl1

*Листинг 1 – cl1*

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
using namespace std;
class cl2;
class cl1{
private:
    int n;
public:
    cl1(int num){
        n = num;
    }
    friend int f(cl1&, cl2&);
};
```

### 5.2 Файл cl2

*Листинг 2 – cl2*

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
using namespace std;
class cl2{
private:
    int m1, m2;
public:
    void set_num(int num1, int num2){
        m1 = num1;
        m2 = num2;
    }
    friend int f(cl1&, cl2&);
};
```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "cl1"
#include "cl2"
using namespace std;
int f(cl1& A, cl2& B){
    int max = A.n;
    if (max <= B.m1){
        max = B.m1;
    }
    if (max <= B.m2){
        max = B.m2;
    }
    return max;
}
int main()
{
    int n, m1, m2;
    cin >> n >> m1 >> m2;
    cl1 obj1(n);
    cl2 obj2;
    obj2.set_num(m1, m2);
    cout << "max = " << f(obj1, obj2);
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

*Таблица 5 – Результат тестирования программы*

<b>Входные данные</b>	<b>Ожидаемые выходные данные</b>	<b>Фактические выходные данные</b>
1 10 100	max = 100	max = 100

## 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-avvora.ru/student/files/methodicheskoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avvora.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-avvora.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avvora.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).