

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 конструктора класса cl_obj1..... | 9  |
| 3.2 Алгоритм метода set класса cl_obj2.....   | 9  |
| 3.3 Алгоритм функции main.....                | 9  |
| 3.4 Алгоритм функции mem.....                 | 10 |
| 4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....                  | 11 |
| 5 КОД ПРОГРАММЫ.....                          | 13 |
| 5.1 Файл cl_obj1.cpp.....                     | 13 |
| 5.2 Файл cl_obj1.h.....                       | 13 |
| 5.3 Файл cl_obj2.cpp.....                     | 13 |
| 5.4 Файл cl_obj2.h.....                       | 14 |
| 5.5 Файл main.cpp.....                        | 14 |
| 6 ТЕСТИРОВАНИЕ.....                           | 16 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....         | 17 |

# 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 используется:

- объект `cl1` класса `cl_obj1` предназначен для Объект первого типа;
- объект `cl2` класса `cl_obj2` предназначен для Объект второго типа;
- функция `mem` для Дружественная функция нахождения максимума из полей двух объектов;
- `cin/cout` - объекты стандартного потока ввода/вывода;
- `max` - функция поиска максимума из двух чисел.

Класс `cl_obj1`:

- свойства/поля:
  - поле Целое число:
    - наименование — `a`;
    - тип — `int`;
    - модификатор доступа — `private`;
- функционал:
  - метод `cl_obj1` — Параметризированный конструктор.

Класс `cl_obj2`:

- свойства/поля:
  - поле Целое число 1:
    - наименование — `a`;
    - тип — `int`;
    - модификатор доступа — `private`;
  - поле Целое число 2:
    - наименование — `b`;
    - тип — `int`;
    - модификатор доступа — `private`;

- функционал:
  - о метод set — Установка значений полей a и b.

Таблица 1 – Иерархия наследования классов

| № | Имя класса | Классы-наследники | Модификатор доступа при наследовании | Описание                                | Номер |
|---|------------|-------------------|--------------------------------------|---|-------|
| 1 | cl_obj1    |                   |                                      | Класс для создания объекта первого типа |       |
| 2 | cl_obj2    |                   |                                      | Класс для создания 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 конструктора класса `cl_obj1`

Функционал: Параметризированный конструктор.

Параметры: `int a` - целое число для установки.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса `cl_obj1`

| № | Предикат | Действия   | №<br>перехода |
|---|----------|--|---------------|
| 1 |          | Установка значения поля <code>a</code> равному значению параметра <code>a</code> | Ø             |

### 3.2 Алгоритм метода `set` класса `cl_obj2`

Функционал: Установка значений полей `a` и `b`.

Параметры: `int a, b` - целые числа.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода `set` класса `cl_obj2`

| № | Предикат | Действия   | №<br>перехода |
|---|----------|--|---------------|
| 1 |          | Установка значений полей <code>a</code> и <code>b</code> значениями параметров <code>a</code> и <code>b</code> | Ø             |

### 3.3 Алгоритм функции `main`

Функционал: Главная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *main*

| № | Предикат | Действия  | №<br>перехода |
|---|----------|---|---------------|
| 1 |          | Объявление переменной a1 типа int   | 2             |
| 2 |          | Ввод значения a1  | 3             |
| 3 |          | Инициализация объекта cl1 класса cl_obj1 с передачей a1 параметризованному конструктору | 4             |
| 4 |          | Объявление переменных a1 и b1 типа int  | 5             |
| 5 |          | Ввод значений a1 и b1   | 6             |
| 6 |          | Объявление объекта cl2 класса cl_obj2   | 7             |
| 7 |          | Вызов метода met с передачей cl1 и cl2 как параметров и вывод возвращаемого значения    | 8             |
| 8 |          | Возврат значения 0  | ∅             |

### 3.4 Алгоритм функции *met*

Функционал: Вычисление максимального значения из полей двух классов.

Параметры: cl\_obj1 cl1, cl\_obj2 cl2 - объекты классов для нахождения максимума из их полей.

Возвращаемое значение: int - максимальное число.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции *met*

| № | Предикат | Действия  | №<br>перехода |
|---|----------|---|---------------|
| 1 |          | Возврат значения вызова функции max с параметрами cl1.a и значения вызова функции max с параметрами cl2.a и cl2.b | ∅             |



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

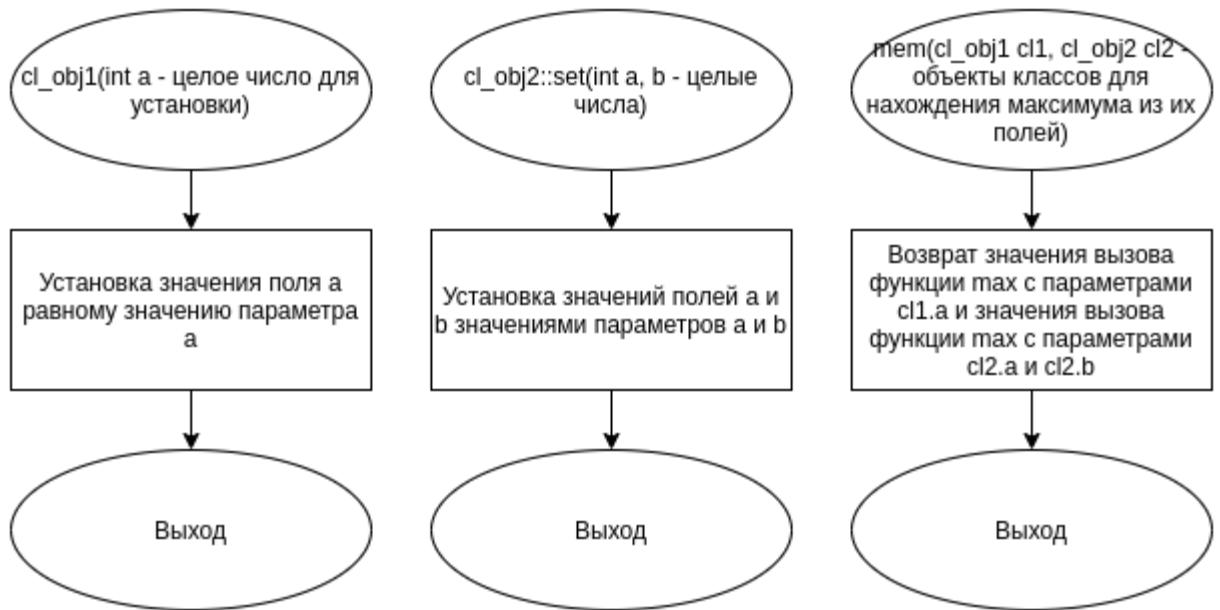


Рисунок 1 – Блок-схема алгоритма

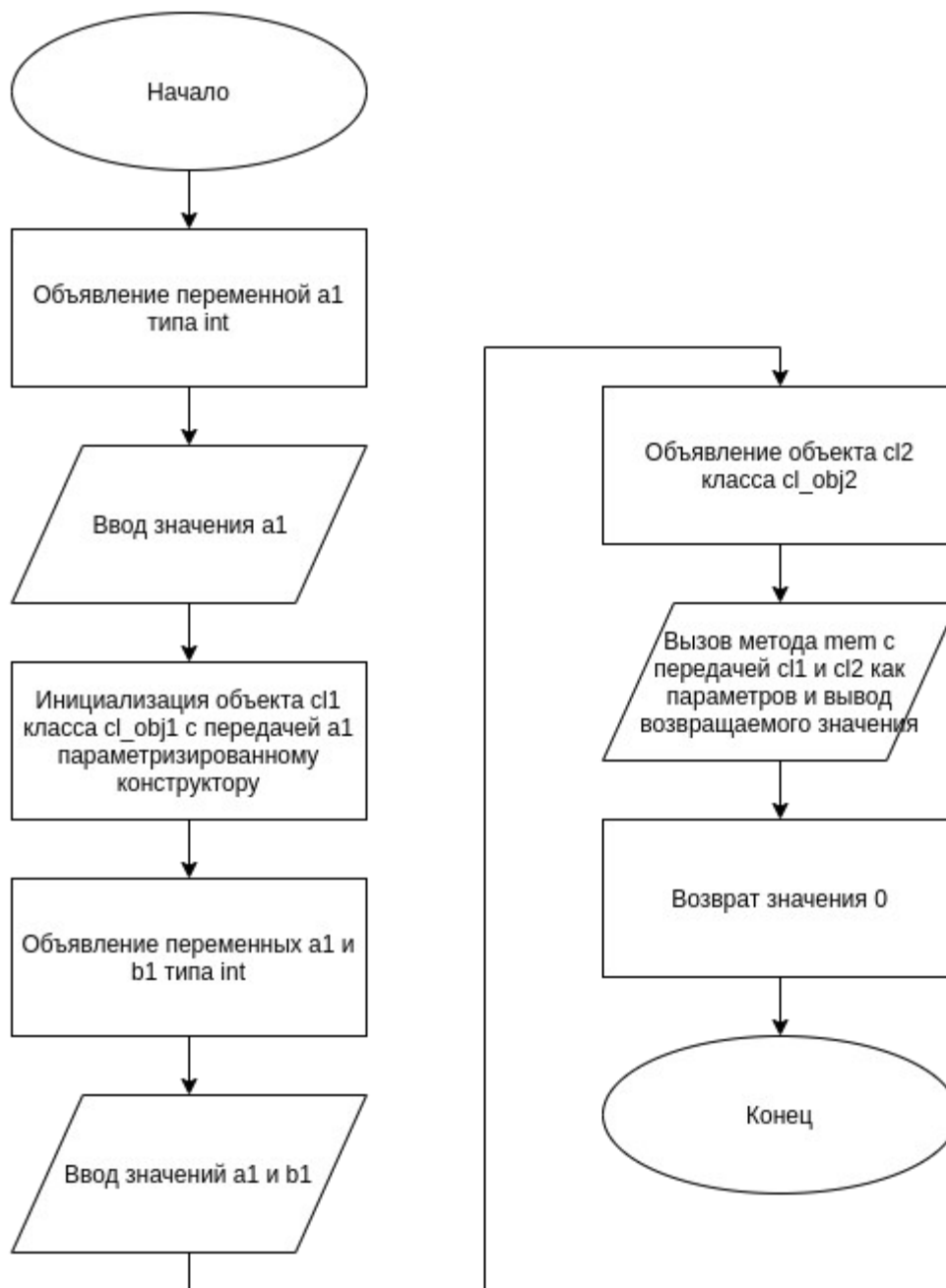


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл cl\_obj1.cpp

*Листинг 1 – cl\_obj1.cpp*

```
#include "cl_obj1.h"

cl_obj1::cl_obj1(int a){
    this->a = a;
}
```

### 5.2 Файл cl\_obj1.h

*Листинг 2 – cl\_obj1.h*

```
#ifndef __CL_OBJ1__H
#define __CL_OBJ1__H

class cl_obj1
{
private:
    int a;
public:
    cl_obj1(int);
    friend int mem(cl_obj1&, class cl_obj2&);
};

#endif
```

### 5.3 Файл cl\_obj2.cpp

*Листинг 3 – cl\_obj2.cpp*

```
#include "cl_obj2.h"
```

```
void cl_obj2::set(int a, int b) {  
    this->a = a;  
    this->b = b;  
}
```

## 5.4 Файл cl\_obj2.h

*Листинг 4 – cl\_obj2.h*

```
#ifndef __CL_OBJ2__H  
#define __CL_OBJ2__H  
  
class cl_obj2  
{  
private:  
    int a, b;  
public:  
    void set(int, int);  
    friend int mem(class cl_obj1&, cl_obj2&);  
};  
#endif
```

## 5.5 Файл main.cpp

*Листинг 5 – main.cpp*

```
#include <stdlib.h>  
#include <stdio.h>  
#include <iostream>  
#include <algorithm>  
#include "cl_obj1.h"  
#include "cl_obj2.h"  
  
using namespace std;  
  
int mem(cl_obj1& cl1, cl_obj2& cl2) {  
    return max(cl1.a, max(cl2.a, cl2.b));  
}
```

```
int main()
{
    int a1;
    cin >> a1;
    cl_obj1 c1(a1);

    int a2, b2;
    cin >> a2 >> b2;
    cl_obj2 c2;
    c2.set(a2, b2);

    cout << "max = " << mem(c1, c2);
    return(0);
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

*Таблица 6 – Результат тестирования программы*

| <b>Входные данные</b> | <b>Ожидаемые выходные данные</b> | <b>Фактические выходные данные</b> |
|-----------------------|----------------------------------|------------------------------------|
| 4<br>5 9              | max = 9                          | max = 9                            |

## 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](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](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).