

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 Class1.....	9
3.2 Алгоритм метода setter класса Class2.....	9
3.3 Алгоритм функции func.....	9
3.4 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	15
5.1 Файл Class1.cpp.....	15
5.2 Файл Class1.h.....	15
5.3 Файл Class2.cpp.....	16
5.4 Файл Class2.h.....	16
5.5 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

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 obj1 класса Class1 предназначен для присваивания полю переменной с класса Class1;
- объект obj2 класса Class2 предназначен для присваивание значения закрытым переменным-членам a и b класса Class2;
- функция func для дружественная функция;
- if - условный оператор;
- cout - объект стандартного потока вывода;
- cin - объект стандартного потока ввода.

Класс Class1:

- свойства/поля:
 - поле сравниваемое поле:
 - наименование — c;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод конструктор — создание объекта, заполнение поля c.

Класс Class2:

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

- тип — int;
- модификатор доступа — private;
- функционал:
 - o метод setter — установка значение полям a и b.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	Class1			создание объекта, установка полей	
2	Class2			установка значение полям a и b	

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 Class1

Функционал: создание объекта, установка полей.

Параметры: int c.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Class1

№	Предикат	Действия	№ перехода
1		присваивание c полю c	Ø

3.2 Алгоритм метода setter класса Class2

Функционал: установка значение полям a и b.

Параметры: int a, int b.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода setter класса Class2

№	Предикат	Действия	№ перехода
1		присваивание a и b полям a и b	Ø

3.3 Алгоритм функции func

Функционал: нахождение максимума среди полей объектов obj1, obj2.

Параметры: const obj1, const obj2.

Возвращаемое значение: int max.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции func

№	Предикат	Действия	№ перехода
1		инициализация переменной max типа int = полюс объекта obj1	2
2	max<a	присваивание значения a переменной max	3
			3
3	max<b	присваивание значения b переменной max	4
			4
4		возврат max	∅

3.4 Алгоритм функции main

Функционал: запуск программы.

Параметры: none.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		инициализация переменной c типа int	2
2		ввод значения c	3
3		создание объекта obj1 класса Class1 с параметром c	4
4		объявление переменных a и b типа int	5
5		ввод значений a и b	6
6		создание объекта obj2 класса Class2	7
7		вызов метода setter у obj2 с параметрами a и b	8

№	Предикат	Действия	№ перехода
8		вывод "max = " + результат вызова func(obj1, obj2)	9
9		возврат значения 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

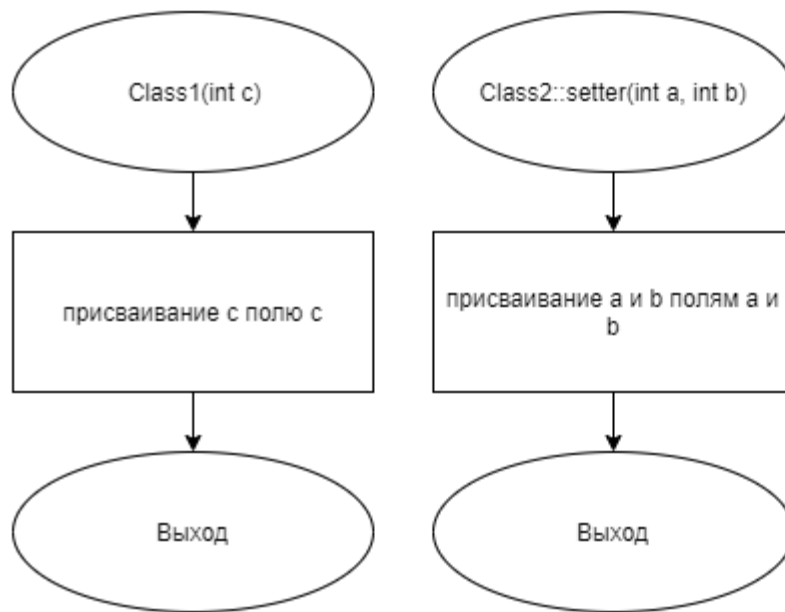


Рисунок 1 – Блок-схема алгоритма

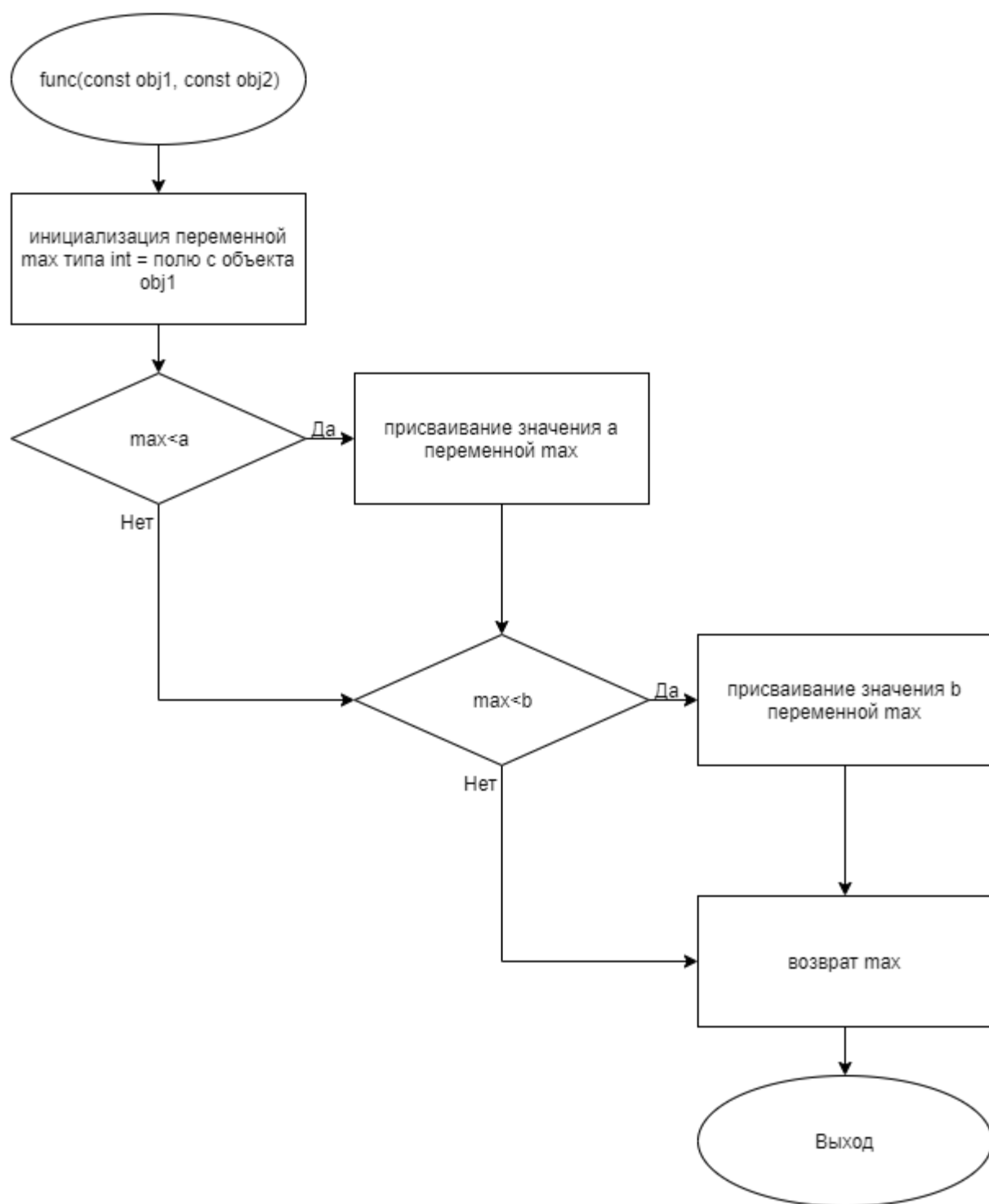


Рисунок 2 – Блок-схема алгоритма

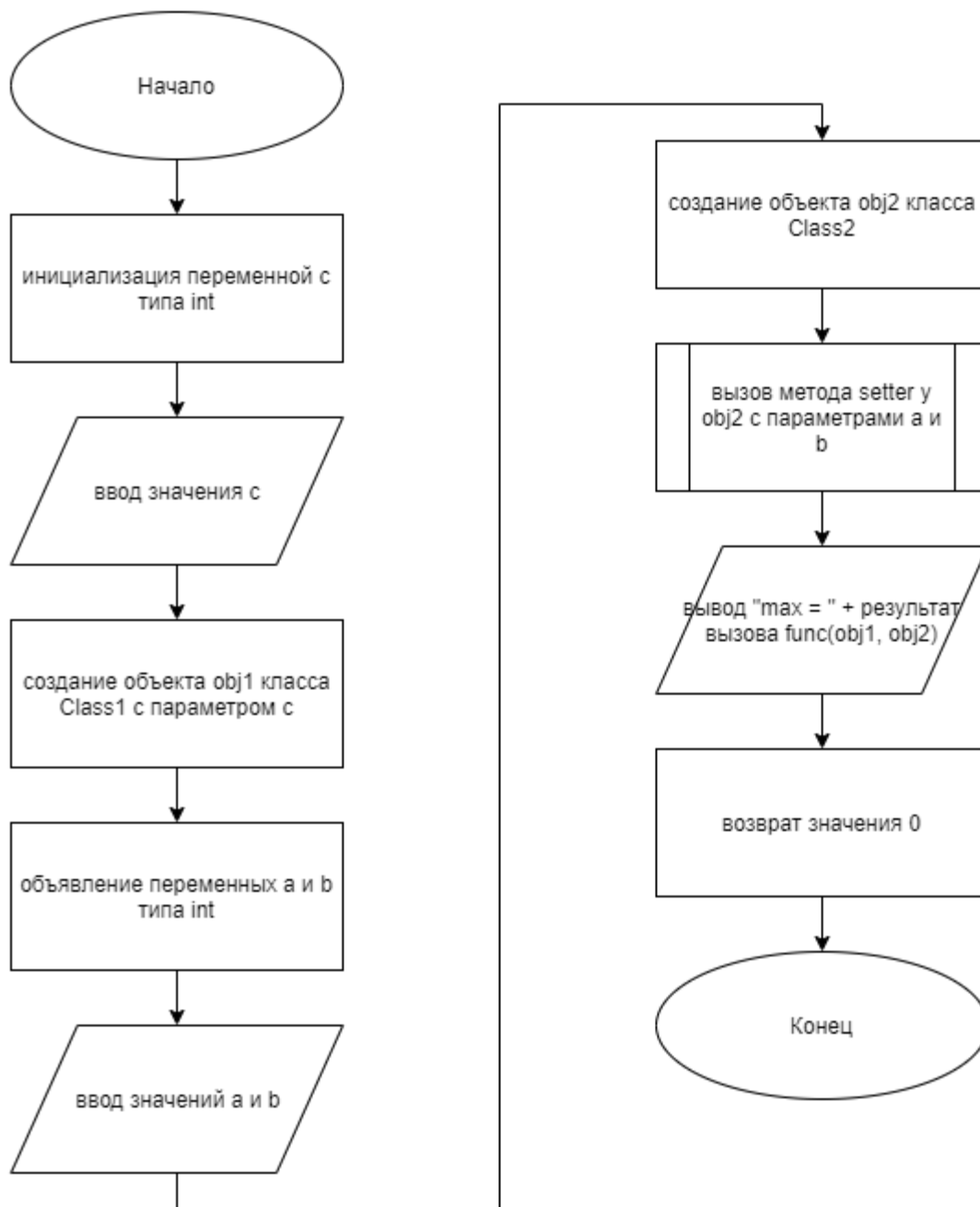


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Class1.cpp

Листинг 1 – Class1.cpp

```
#include "Class1.h"

Class1::Class1(int c)
{
    this->c=c;
}
```

5.2 Файл Class1.h

Листинг 2 – Class1.h

```
#ifndef __CLASS1__H
#define __CLASS1__H
#include <iostream>
#include "Class2.h"

using namespace std;

class Class1;
class Class2;

class Class1
{
public:
    Class1(int);
    friend int func(const Class1, const Class2);
private:
    int c;
};

#endif
```

5.3 Файл Class2.cpp

Листинг 3 – Class2.cpp

```
#include "Class2.h"

void Class2::setter(int a, int b)
{
    this->a=a;
    this->b=b;
}
```

5.4 Файл Class2.h

Листинг 4 – Class2.h

```
#ifndef __CLASS2__H
#define __CLASS2__H
#include "Class1.h"

class Class1;
class Class2;

class Class2
{
public:
    void setter(int a, int b);
    friend int func(const Class1, const Class2);
private:
    int a, b;
};

#endif
```

5.5 Файл main.cpp

Листинг 5 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "Class1.h"
#include "Class2.h"
#include <iostream>
```

```

using namespace std;

int main()
{
    int c;
    cin>>c;
    int a, b;
    Class1 obj1(c);
    cin>>a>>b;
    Class2 obj2;
    obj2.setter(a, b);
    cout<<"max = "<<func(obj1, obj2);
    return(0);
}

int func(const Class1 obj1, const Class2 obj2)
{
    int max=obj1.c;
    if (max<obj2.a)
    {
        max=obj2.a;
    }
    if (max<obj2.b)
    {
        max=obj2.b;
    }
    return max;
}

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
9 9 1	max = 9	max = 9
1 4 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 (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).