

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 деструктора класса Class1.....	9
3.3 Алгоритм конструктора класса Class2.....	9
3.4 Алгоритм метода Method класса Class2.....	10
3.5 Алгоритм деструктора класса Class2.....	10
3.6 Алгоритм функции Func.....	10
3.7 Алгоритм функции main.....	11
3.8 Алгоритм метода get_r() класса Class1.....	12
3.9 Алгоритм метода get_r2() класса Class2.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	18
5.1 Файл Class1.cpp.....	18
5.2 Файл Class1.h.....	18
5.3 Файл Class2.cpp.....	19
5.4 Файл Class2.h.....	19
5.5 Файл main.cpp.....	20
6 ТЕСТИРОВАНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

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 `a` класса `Class1` предназначен для Создания поля;
- объект `b` класса `Class2` предназначен для Создания двух полей и дружественной функции;
- функция `Func` для ищет максимум среди 3х элементов;
- `cin` - объект стандартного потока ввода;
- `cout` - объект стандартного потока вывода;
- `max` - выбор максимального значения, среди заданных.

Класс `Class1`:

- свойства/поля:
 - поле хранение значения:
 - наименование — `x1`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `Class1` — Конструктор (присвоение `x1` значения `x`);
 - метод `get_r()` — Возвращение значение закрытого поля;
 - метод `~Class1` — Деструктор.

Класс `Class2`:

- свойства/поля:
 - поле Хранение значения:
 - наименование — `x2`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле Хранение значения:

- наименование — y2;
- тип — int;
- модификатор доступа — private;
- функционал:
 - o метод Class2 — Конструктор;
 - o метод Method — Присвоение x2 значения x, y2 значения y;
 - o метод get_r2() — Возвращение значение закрытых полей;
 - o метод ~Class2 — Деструктор.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	Class1			Первый класс с одним полем	
2	Class2			Второй класс с двумя полями и дружественной функцией	

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: Конструктор (присвоение x1 значения x).

Параметры: int x - значение для поля x1.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Class1

№	Предикат	Действия	№ перехода
1		присвоение x1 значения x	Ø

3.2 Алгоритм деструктора класса Class1

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса Class1

№	Предикат	Действия	№ перехода
1		Удаление объекта	Ø

3.3 Алгоритм конструктора класса Class2

Функционал: Конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса Class2

№	Предикат	Действия	№ перехода
1		Создание объекта	Ø

3.4 Алгоритм метода Method класса Class2

Функционал: Присвоение x2 значения x, y2 значения y.

Параметры: int x, y - значения для полей x2, y2.

Возвращаемое значение: void - не возвращает значения.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Method класса Class2

№	Предикат	Действия	№ перехода
1		Присвоение x2 значения x,	2
2		Присвоение y2 значения y	Ø

3.5 Алгоритм деструктора класса Class2

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 6.

Таблица 6 – Алгоритм деструктора класса Class2

№	Предикат	Действия	№ перехода
1		Удаление объекта	Ø

3.6 Алгоритм функции Func

Функционал: поиск максимума среди трех значений.

Параметры: int x,y,z - три значения для поиска среди них максимума.

Возвращаемое значение: int - максимальное значение.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции Func

№	Предикат	Действия	№ перехода
1		Возвращение значения $\max(\{x,y,z\})$	Ø

3.7 Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: целочисленный - индикатор корректности завершения алгоритма.

Алгоритм функции представлен в таблице 8.

Таблица 8 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленных x1,x2,y2	2
2		Ввод значения x1 с клавиатуры	3
3		создание объекта a класса Class1 с параметром x1	4
4		Ввод значений x2, y2 с клавиатуры	5
5		создание объекта b класса Class2	6
6		вызов для объекта b метода Method с параметрами x2, y2	7
7		присвоение x2, y2 = вызов метода get_r2() объекта b	8
8		Вывод на экран с первой строки, с помощью дружественной функции Func с параметрами a.get_r() (метода get_r для объекта a класса Class1), x2, y2, максимального значения среди трех полей двух классов	Ø

3.8 Алгоритм метода `get_r()` класса `Class1`

Функционал: Возвращение значение закрытого поля.

Параметры: нет.

Возвращаемое значение: `int` - значение `x1`.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода `get_r()` класса `Class1`

№	Предикат	Действия	№ перехода
1		Возвращение значение закрытого поля <code>x1</code>	Ø

3.9 Алгоритм метода `get_r2()` класса `Class2`

Функционал: Возвращение значение закрытых полей.

Параметры: нет.

Возвращаемое значение: `int` - значения `x2`, `y2`.

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода `get_r2()` класса `Class2`

№	Предикат	Действия	№ перехода
1		Возвращение значений закрытых полей <code>x2</code> , <code>y2</code>	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

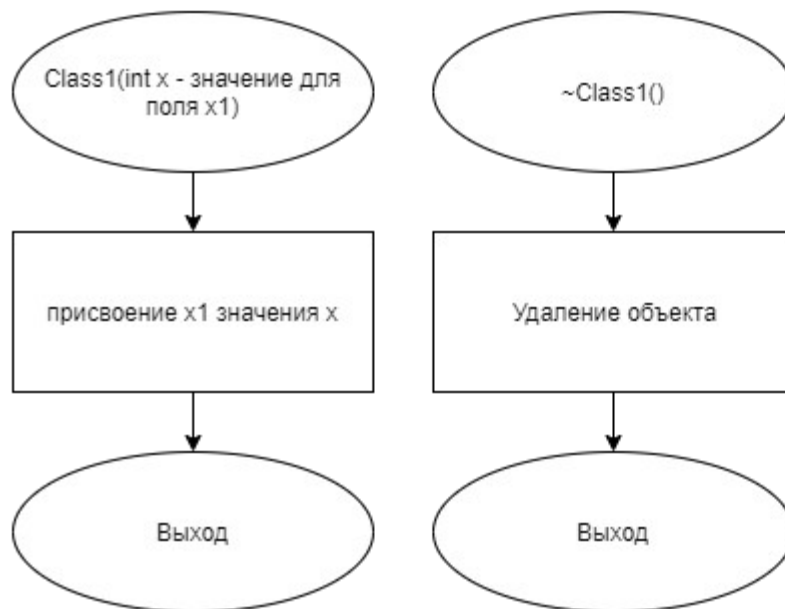


Рисунок 1 – Блок-схема алгоритма

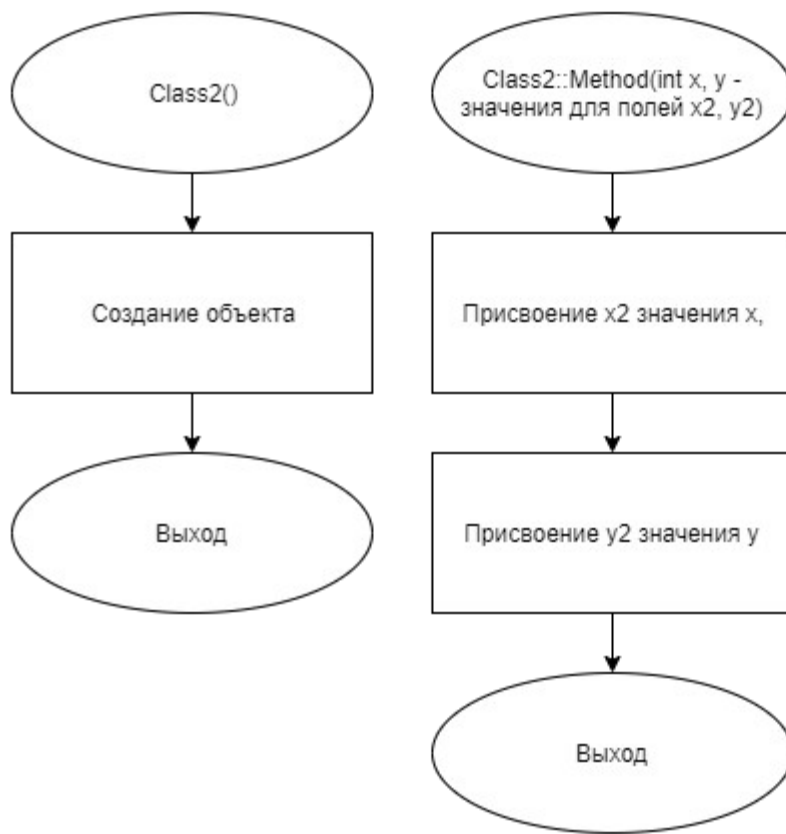


Рисунок 2 – Блок-схема алгоритма

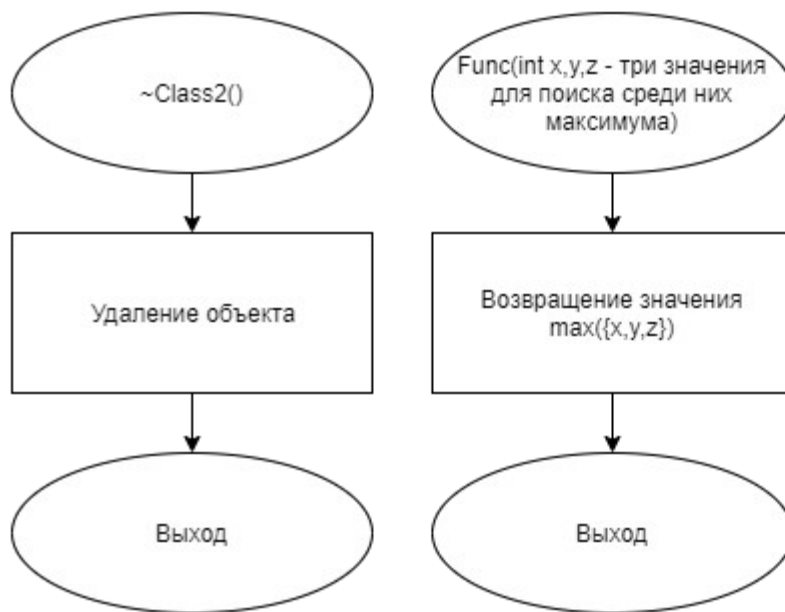


Рисунок 3 – Блок-схема алгоритма

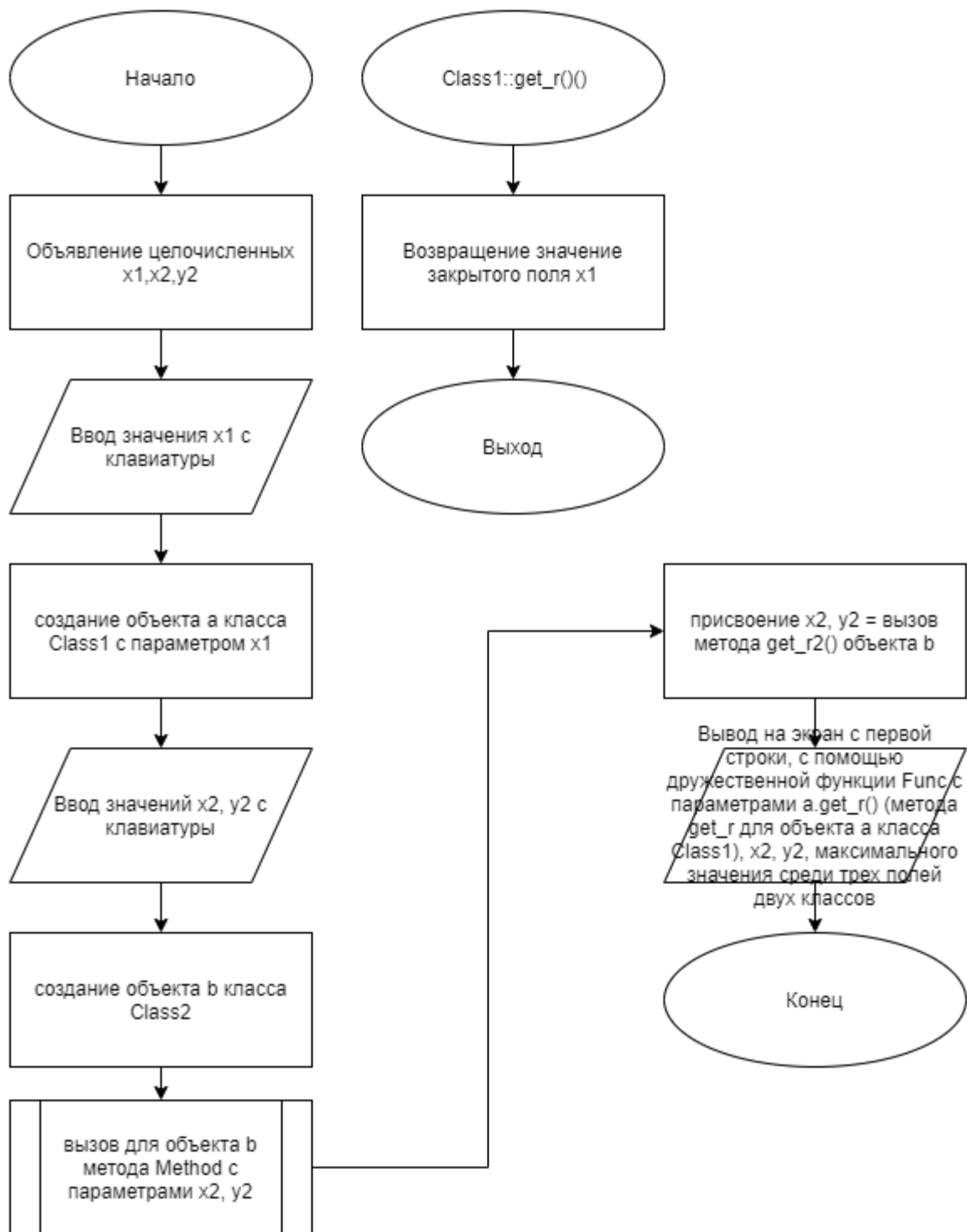


Рисунок 4 – Блок-схема алгоритма



Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Class1.cpp

Листинг 1 – Class1.cpp

```
#include "Class2.h"
#include "Class1.h"
#include <iostream>
#include <algorithm>
using namespace std;
Class1::Class1(int x){
    x1=x;
}
int Class1::get_r(){
    return x1;
}
Class1::~~Class1(){}

```

5.2 Файл Class1.h

Листинг 2 – Class1.h

```
#ifndef __CLASS1__H
#define __CLASS1__H
class Class1{
    friend class Class2;
private:
    int x1;
public:

    Class1(int y);
    int get_r();
    ~Class1();

};

#endif

```


5.3 Файл Class2.cpp

Листинг 3 – Class2.cpp

```
#include "Class1.h"
#include "Class2.h"
#include <iostream>
#include <algorithm>
using namespace std;

Class2::Class2(){
void Class2::Method(int x, int y){
    x2=x;
    y2=y;
}
int Class2::get_r2(){
    return x2,y2;
}
Class2::~~Class2(){}

```

5.4 Файл Class2.h

Листинг 4 – Class2.h

```
#ifndef __CLASS2__H
#define __CLASS2__H

class Class2{
    private:
        int x2;
        int y2;
    public:
        Class2();
        void Method(int x, int y);
        int get_r2();
        friend int Func(int x, int y,int z);
        ~Class2();
};
#endif

```

5.5 Файл main.cpp

Листинг 5 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <algorithm>
#include "Class1.h"
#include "Class2.h"
using namespace std;
int Func(int x, int y, int z){
    return(max({x,y,z}));
}
int main()
{
    int x1,x2,y2;
    cin>>x1;
    Class1 a(x1);
    cin>>x2>>y2;
    Class2 b;
    b.Method(x2,y2);
    x2, y2 = b.get_r2();
    cout<<"max = "<<Func(a.get_r(), x2, y2);;
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 5 100	max = 100	max = 100

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).