

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 FirstType.....	9
3.2 Алгоритм метода setProperties класса SecondType.....	9
3.3 Алгоритм функции findMax.....	10
3.4 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	15
5.1 Файл FirstType.cpp.....	15
5.2 Файл FirstType.h.....	15
5.3 Файл main.cpp.....	16
5.4 Файл SecondType.cpp.....	17
5.5 Файл SecondType.h.....	17
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 FirstType предназначен для функционирования системы;
- объект obj2 класса SecondType предназначен для функционирования системы;
- объект cin класса потокового ввода предназначен для функционирования системы;
- объект cout класса потокового вывода предназначен для функционирования системы;
- функция findMax для нахождения максимального значения среди свойств объектов класса 'FirstType' и 'SecondType';
- оператор return - возврат значения из функции.

Класс FirstType:

- свойства/поля:
  - поле хранит объект типа 'FirstType':
    - наименование — property;
    - тип — вещественный;
    - модификатор доступа — private;
- функционал:
  - метод FirstType — конструктор.

Класс SecondType:

- свойства/поля:
  - поле первое свойство объекта типа 'SecondType':
    - наименование — property1;
    - тип — вещественный;

- модификатор доступа — public;
- поле второе свойство объекта типа 'SecondType':
  - наименование — property2;
  - тип — вещественный;
  - модификатор доступа — public;
- поле свойство объекта 'FirstType':
  - наименование — property;
  - тип — вещественный;
  - модификатор доступа — private;
- функционал:
  - метод setProperties — установка значения обоих свойств 'property1' и 'property2'.

*Таблица 1 – Иерархия наследования классов*

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	FirstType			создание объекта	
2	SecondType			создание объекта	

## 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 FirstType

Функционал: конструктор.

Параметры: int value.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса FirstType

№	Предикат	Действия	№ перехода
1		принимается один аргумент типа 'double' и инициализирует приватное поле 'property' значением этого аргумента при создании объекта класса 'FirstType'	Ø

### 3.2 Алгоритм метода setProperties класса SecondType

Функционал: установка значения обоих свойств 'property1' и 'property2'.

Параметры: value1, value2.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода setProperties класса SecondType

№	Предикат	Действия	№ перехода
1		установка значений для двух свойств объекта класса 'SecondType', аргумент 'value1' устанавливается в свойство 'property1', аргумент 'value2' устанавливается в свойство 'property2'	2

№	Предикат	Действия	№ перехода
2		присвоение свойству 'property1' значение переменной 'value1'	3
3		присвоение свойству 'property2' значение переменной 'value2'	∅

### 3.3 Алгоритм функции findMax

Функционал: нахождение максимального значения среди свойств объектов классов 'FirstType' и 'SecondType'.

Параметры: константные ссылки на объекты классов 'FirstType' и 'SecondType'.

Возвращаемое значение: целочисленное значение max.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции findMax

№	Предикат	Действия	№ перехода
1		объявление функции 'findMax', которая принимает два константных ссылочных аргумента	2
2		присвоение значений свойств объекта 'first' класса 'FirstType' к переменной max	3
3	second.getProperty1() > max	присвоение значений первого свойства объекта 'second' класса 'SecondType' к переменной max	4
			4
4	second.getProperty2() > max	присвоение значений второго свойства объекта 'second' класса 'SecondType' к переменной max	5
			5
5		возврат значения max	∅

### 3.4 Алгоритм функции main

Функционал: основной алгоритм работы программы.



Параметры: отсутствуют.

Возвращаемое значение: целочисленный индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		объявление дробных переменных value1, value2, value3	2
2		ввод значения переменной value1	3
3		создание объекта класса 'FirstType' с именем 'obj1', и его конструктор вызывается с передачей значения 'value1' - аргумент	4
4		ввод значения переменной value2, value3	5
5		создание объекта класса 'SecondType' с именем 'obj2'	6
6		вызов метода 'setProperties' объекта 'obj2' класса 'SecondType' с передачей значений value2, value3 в качестве аргументов	7
7		вызов функции 'findMax', в качестве аргументов передаются объекты 'obj1' и 'obj2', результат функции присваивается переменной 'maxValue'	8
8		вывод на экран 'max', значение переменной 'maxValue'	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

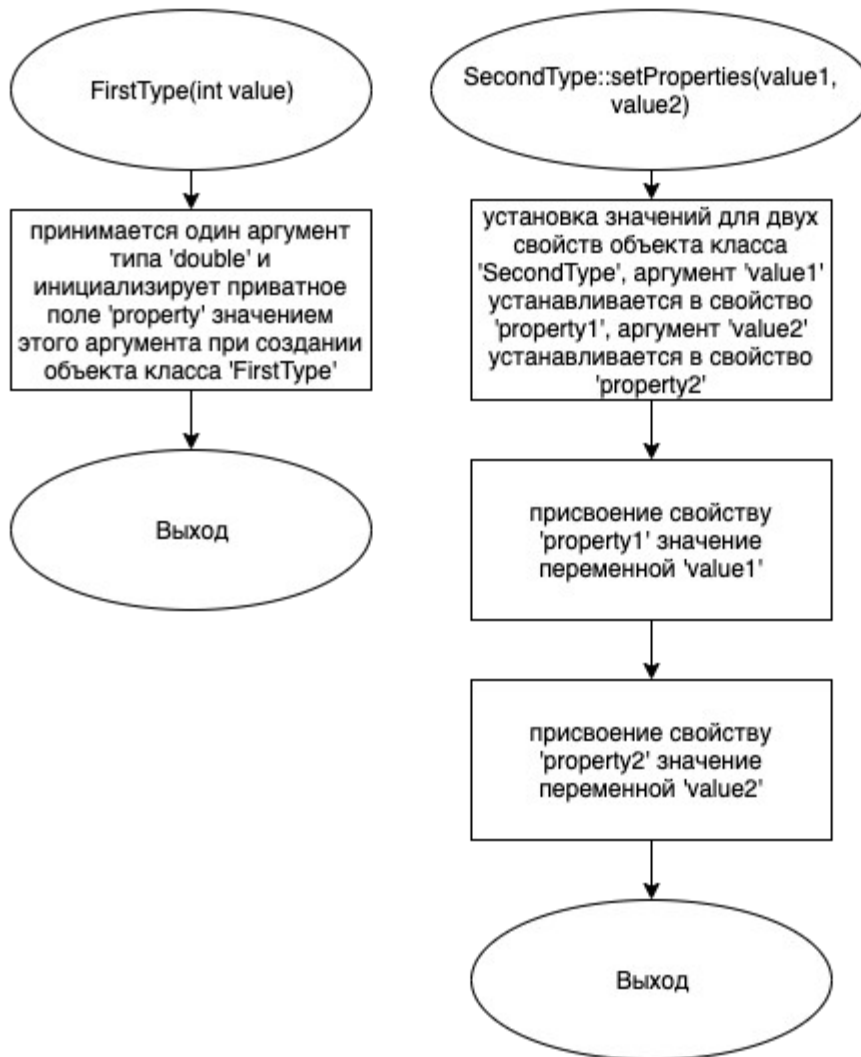


Рисунок 1 – Блок-схема алгоритма

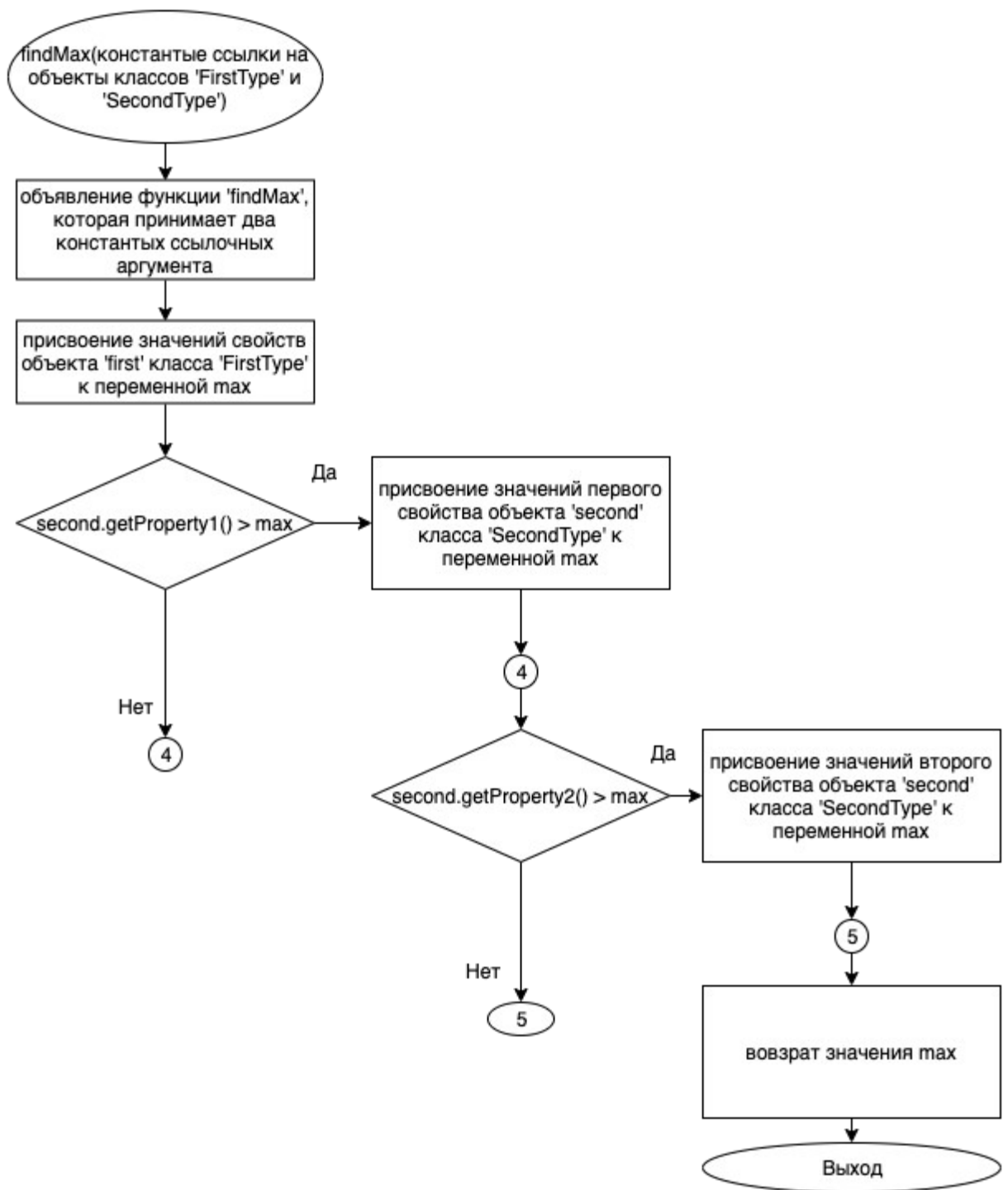


Рисунок 2 – Блок-схема алгоритма



Рисунок 3 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл FirstType.cpp

*Листинг 1 – FirstType.cpp*

```
#include "FirstType.h"

FirstType::FirstType(double value) : property(value) {}
```

### 5.2 Файл FirstType.h

*Листинг 2 – FirstType.h*

```
#ifndef __FIRSTTYPE__H
#define __FIRSTTYPE__H

#include "SecondType.h"

class SecondType;

class FirstType {
friend int findMax(const FirstType& first, const SecondType& second);
private:
    double property;

public:
    FirstType(double value);

};

#endif
```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>

#include "FirstType.h"
#include "SecondType.h"

using namespace std;

int findMax(const FirstType& first, const SecondType& second);

int main() {
    double value1, value2, value3;

    cin >> value1;

    FirstType obj1(value1);

    cin >> value2 >> value3;

    SecondType obj2;

    obj2.setProperties(value2, value3);

    double maxValue = findMax(obj1, obj2);

    cout << "max = " <<  maxValue << endl;

    return 0;
}

int findMax(const FirstType& first, const SecondType& second) {
    double max = first.property;
    if(second.property1 > max) {
        max = second.property1;
    }
    if(second.property2 > max) {
        max = second.property2;
    }
    return max;
}
```

## 5.4 Файл SecondType.cpp

*Листинг 4 – SecondType.cpp*

```
#include "SecondType.h"

void SecondType::setProperties(double value1, double value2) {
    property1 = value1;
    property2 = value2;
}
```

## 5.5 Файл SecondType.h

*Листинг 5 – SecondType.h*

```
#ifndef __SECONDTYPE__H
#define __SECONDTYPE__H

#include "FirstType.h"

class FirstType;

class SecondType{
friend int findMax(const FirstType& first, const SecondType& second);
private:
    double property1;
    double property2;
public:

    void setProperties(double value1, double value2);
};

#endif
```

## 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>
5.0 10.0 20.0	max = 20	max = 20
6.0 10.0 30.0	max = 30	max = 30
25.0 5.0 10.0	max = 25	max = 25



## 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).