

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.1 Алгоритм функции main.....	8
3.2 Алгоритм метода GetA класса test.....	9
3.3 Алгоритм метода SetA класса test.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
5 КОД ПРОГРАММЫ.....	11
5.1 Файл main.cpp.....	11
5.2 Файл test.cpp.....	12
5.3 Файл test.h.....	12
6 ТЕСТИРОВАНИЕ.....	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	15

# 1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется указатель на массив целого типа.

Конструктору объекта передается целочисленный параметр. Параметр должен иметь значение больше 4. По значению параметра определяется размерность целочисленного массива из закрытой области и каждому элементу присваивается это же значение.

Объект имеет функциональность, по которой выводит содержимое целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Функциональность объекта можно расширить по усмотрению разработчика не более чем на два метода.

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если значение меньше или равно 4, то создание системы прекращается и выводится сообщение. Если система построена, то посредством параметризованного конструктора создаются объекты.

Далее система функционирует по алгоритму:

1. ...
2. Первому объекту присвоить второй объект.
3. ...
4. С первой строки вывести содержимое массива первого объекта.
5. ...
6. Со второй строки вывести содержимое массива второго объекта.

## 1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число»

Пример.

5  
8

## 1.2 Описание выходных данных

Если система была построена, то в первой строке:

«Целое число» «Целое число» . . .

Во второй строке:

«Целое число» «Целое число» . . .

Если система не была построена, то в первой строке выводится некорректное значение и вопросительный знак:

«Целое число»?

**Пример вывода.**

5 5 5 5 5  
8 8 8 8 8 8 8 8

## 2 МЕТОД РЕШЕНИЯ

Класс test:

- свойства/поля:
  - поле поле-указатель на массив:
    - наименование — a;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - метод GetA — возвращение значения a;
  - метод SetA — присваивание значению a valueA.

## 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 функции main

Функционал: запуск программы.

Параметры: нет.

Возвращаемое значение: код ошибки (int).

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменных x и y типа int	2
2		ввод значения переменной x с клавиатуры	3
3	x<=4	ввод значения переменной y с клавиатуры	4
		вывод на экран "(значение переменной x)?"	12
4	y<=4		5
		вывод на экран "(значение переменной y)?"	12
5		создание объектов example1 и example2 класса test	6
6		конструктору объекта example1 передается аргумент x, конструктору объекта example 2 передается аргумент y	7
7		инициализация указателя b на массив a через метод GetA() объекта example1	8
8		присваивание объекта example2 объекту example1	9
9		массиву a объекта example1 передается значение указателя b через метода SetA()	10

№	Предикат	Действия	№ перехода
10		вызов метода print объекта example1	11
11		вызов метода print объекта example2	12
12		возврат значения 0	∅

### 3.2 Алгоритм метода GetA класса test

Функционал: возвращение значения a.

Параметры: нет.

Возвращаемое значение: a.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода GetA класса test

№	Предикат	Действия	№ перехода
1		возврат значения a	∅

### 3.3 Алгоритм метода SetA класса test

Функционал: присваивание значению a valueA.

Параметры: нет.

Возвращаемое значение: -.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода SetA класса test

№	Предикат	Действия	№ перехода
1		присваивание a valueA	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

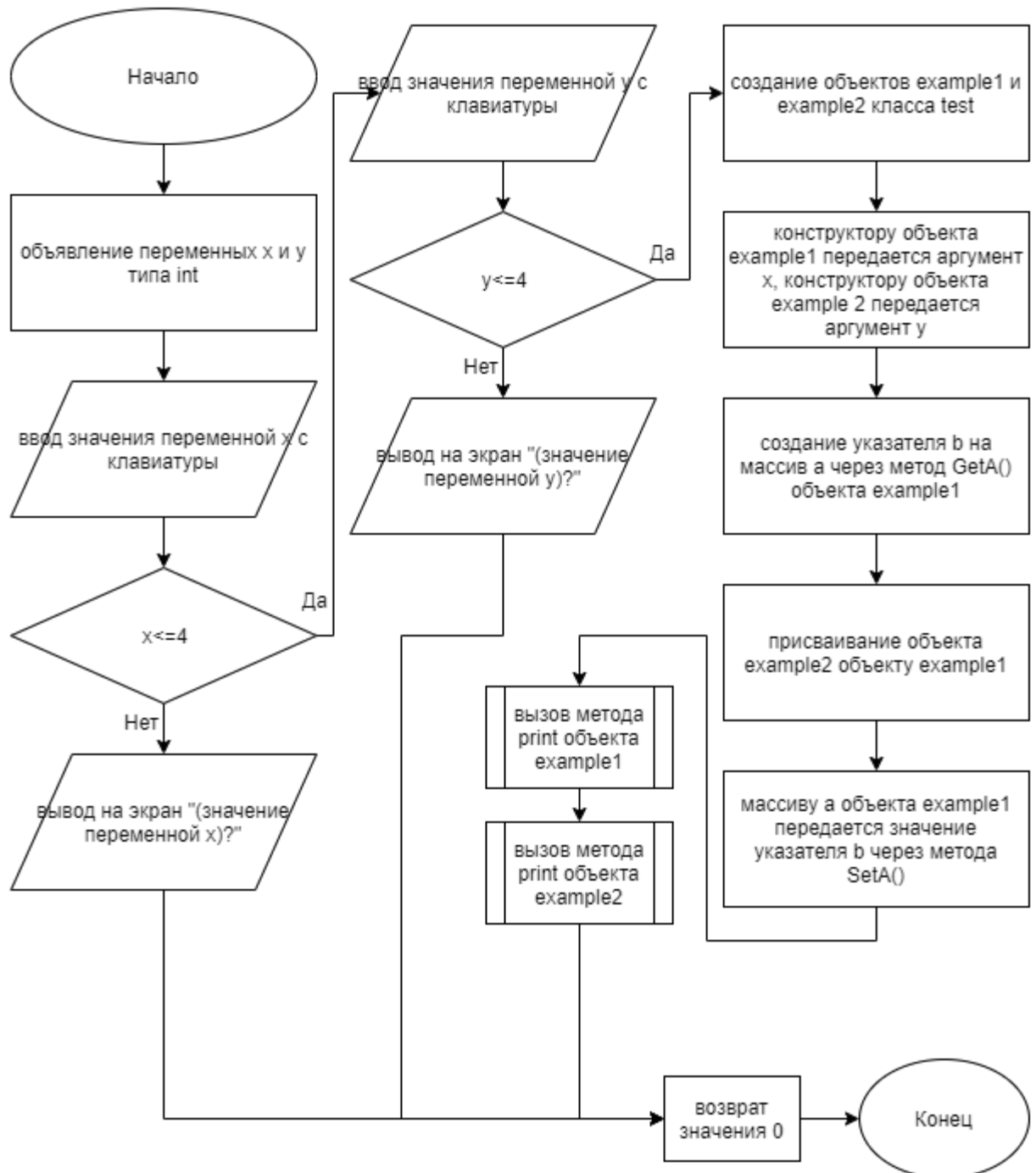


Рисунок 1 – Блок-схема алгоритма



## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

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

using namespace std;

int main()
{
    int x, y;
    cin>>x;
    if (x<=4)
    {
        cout<<x<<"?";
    }
    else
    {
        cin>>y;
        if (y<=4)
        {
            cout<<y<<"?";
        }
        else
        {
            test example1(x), example2(y);
            int* b = example1.GetA();
            example1=example2;
            example1.SetA(b);
            example1.print();
            example2.print();
        }
    }
    return(0);
}
```

## 5.2 Файл test.cpp

*Листинг 2 – test.cpp*

```
#include "test.h"
#include <iostream>
using namespace std;

test::test(int value)
{
    a=new int [value];
    for(int i=0; i<value; i++)
    {
        a[i]=value;
    }
}

int* test::GetA()
{
    return a;
}

void test::SetA(int* valueA)
{
    a=valueA;
}

void test::print()
{
    for (int i=0; i<a[0]; i++)
    {
        if (i<a[0]-1)
        {
            cout<<a[i]<<" ";
        }
        else
        {
            cout<<a[i];
        }
    }
    cout<<endl;
}
```

## 5.3 Файл test.h

*Листинг 3 – test.h*

```
#ifndef __TEST__H
```

```
#define __TEST__H

class test
{
    private:
        int* a;
    public:
        test(int x);
        int* GetA();
        void SetA(int* valueA);
        void print();
};

#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5	0?	0?
5 5 5	5 5 5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5 5 5

## 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).