

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 конструктора класса Test.....	8
3.2 Алгоритм метода output класса Test.....	8
3.3 Алгоритм деструктора класса Test.....	9
3.4 Алгоритм метода get_mass класса Test.....	9
3.5 Алгоритм метода change_mass класса Test.....	10
3.6 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	15
5.1 Файл main.cpp.....	15
5.2 Файл Test.cpp.....	15
5.3 Файл Test.h.....	16
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

# 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

Для решения задачи используется:

- объект `obj1` класса `Test` предназначен для хранения и вывода целочисленного массива;
- объект `obj2` класса `Test` предназначен для хранения и вывода целочисленного массива;
- Объект стандартного потока ввода с клавиатуры `cin`;
- Объект стандартного потока вывода на экран `cout`;
- Условный оператор `if..else`;
- Оператор цикла `for`.

Класс `Test`:

- свойства/поля:
  - поле Указатель на целочисленный массив:
    - наименование — `mass`;
    - тип — `int*`;
    - модификатор доступа — `private`;
- функционал:
  - метод `Test` — Конструктор;
  - метод `get_mass` — Получение закрытого указателя `mass`;
  - метод `change_mass` — Изменение закрытого указателя `mass`;
  - метод `output` — Метод вывода массива на экран;
  - метод `~Test` — Деструктор.

## 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 Test

Функционал: Конструктор.

Параметры: Целочисленная переменная  $i$  для инициализации массива.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Test

№	Предикат	Действия	№ перехода
1		Инициализация указателя mass адресом целочисленного массива размера $i$	2
2		Объявление целочисленной переменной счетчика $j$ и инициализация 0	3
3	$j < 1$	Элементу массива с номером $j$ присваивается значение $i$ Увеличение $j$ на 1	3
			Ø

### 3.2 Алгоритм метода output класса Test

Функционал: Метод вывода массива на экран.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *output* класса *Test*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной счетчика <i>j</i> и инициализация 0	2
2	$j < (\text{mass}[0] - 1)$	Вывод на экран <i>j</i> -ого элемента <i>mass</i> и двух пробелов	3
			4
3		Увеличение <i>j</i> на 1	2
4		Вывод на экран элемента массива с номером <i>mass</i> [0]	∅

### 3.3 Алгоритм деструктора класса *Test*

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса *Test*

№	Предикат	Действия	№ перехода
1		Удаление из памяти массива по адресу <i>mass</i>	∅

### 3.4 Алгоритм метода *get\_mass* класса *Test*

Функционал: Получение закрытого указателя *mass*.

Параметры: нет.

Возвращаемое значение: Указатель на целочисленный массив.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *get\_mass* класса *Test*

№	Предикат	Действия	№ перехода
1		Вернуть значение закрытого указателя <i>mass</i>	Ø

### 3.5 Алгоритм метода *change\_mass* класса *Test*

Функционал: Изменение закрытого указателя *mass*.

Параметры: Указатель на целочисленный массив *x* для перезаписи указателя *mass*.

Возвращаемое значение: Указатель на целочисленный массив.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *change\_mass* класса *Test*

№	Предикат	Действия	№ перехода
1		Закрытому указателю <i>mass</i> присваивается значение <i>x</i>	Ø

### 3.6 Алгоритм функции *main*

Функционал: Выполнение действий, описанных в задаче.

Параметры: нет.

Возвращаемое значение: Целочисленное значение.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление переменных целого типа <i>x1</i> и <i>x2</i>	2
2		Ввод с клавиатуры значения переменной <i>x1</i>	3
3		Ввод с клавиатуры значения переменной <i>x2</i>	4
4	<i>x1 &gt; 4</i> и <i>x2 &gt; 2</i>	Создание объекта <i>obj1</i> класса <i>Test</i> с параметром	6



№	Предикат	Действия	№ перехода
		x1 Создание объекта obj2 класса Test с параметром x2	
			5
5	$x1 \leq 4$	Вывод на экран "(x1)?"	∅
		Вывод на экран "(x2)?"	∅
6		Объявление указателя mass1	7
7		Вызов метода get_mass объекта obj1 класса Test и присваивание этого значения указателю mass1	8
8		Объекту obj1 присваивается объект obj2	9
9		Вызов метода change_mass объекта obj1 класса Test с параметром mass1	10
10		Вызов метода output объекта obj1 класса Test	11
11		Вывод переноса строки	12
12		Вызов метода output объекта obj2 класса Test	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

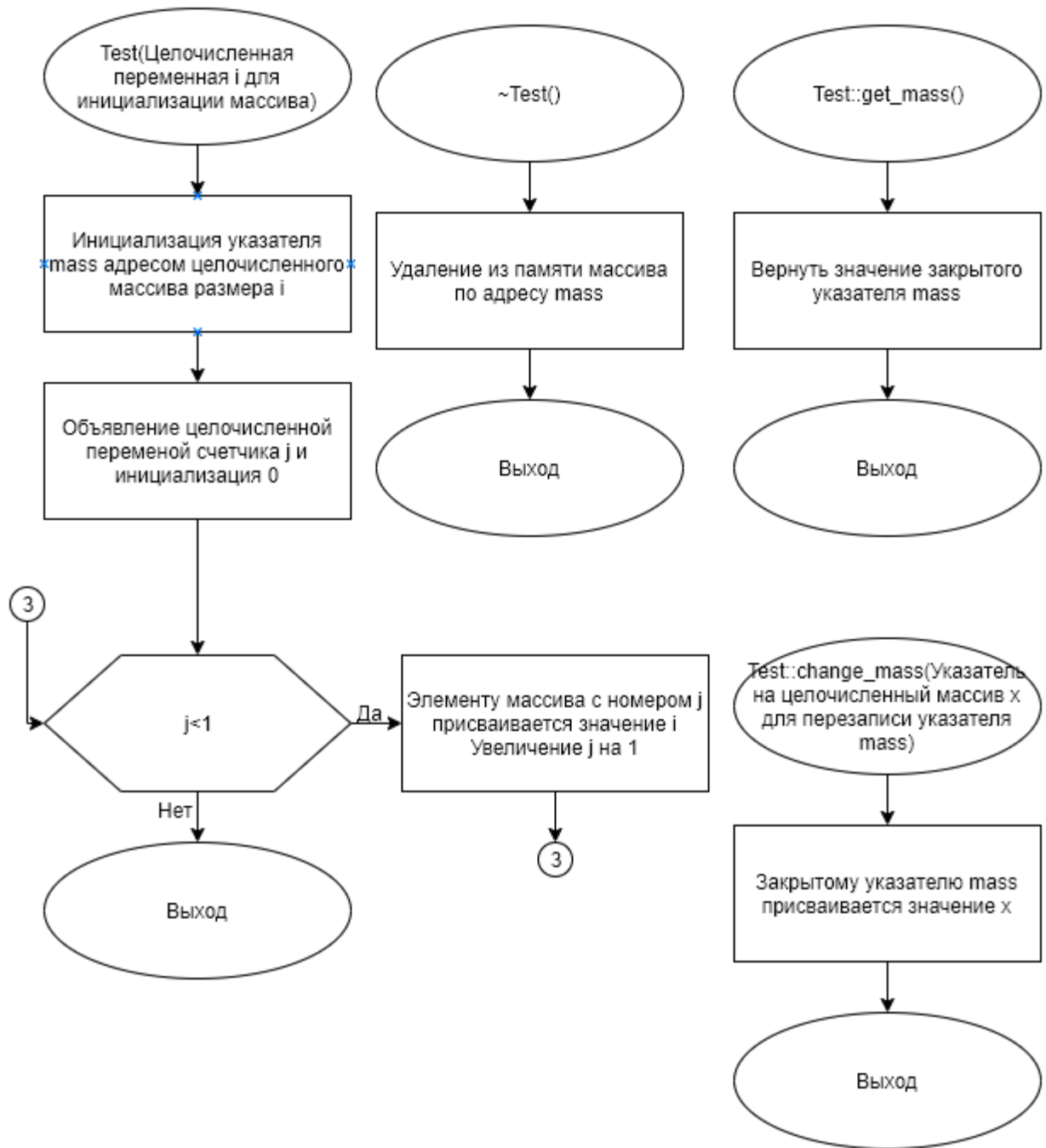


Рисунок 1 – Блок-схема алгоритма

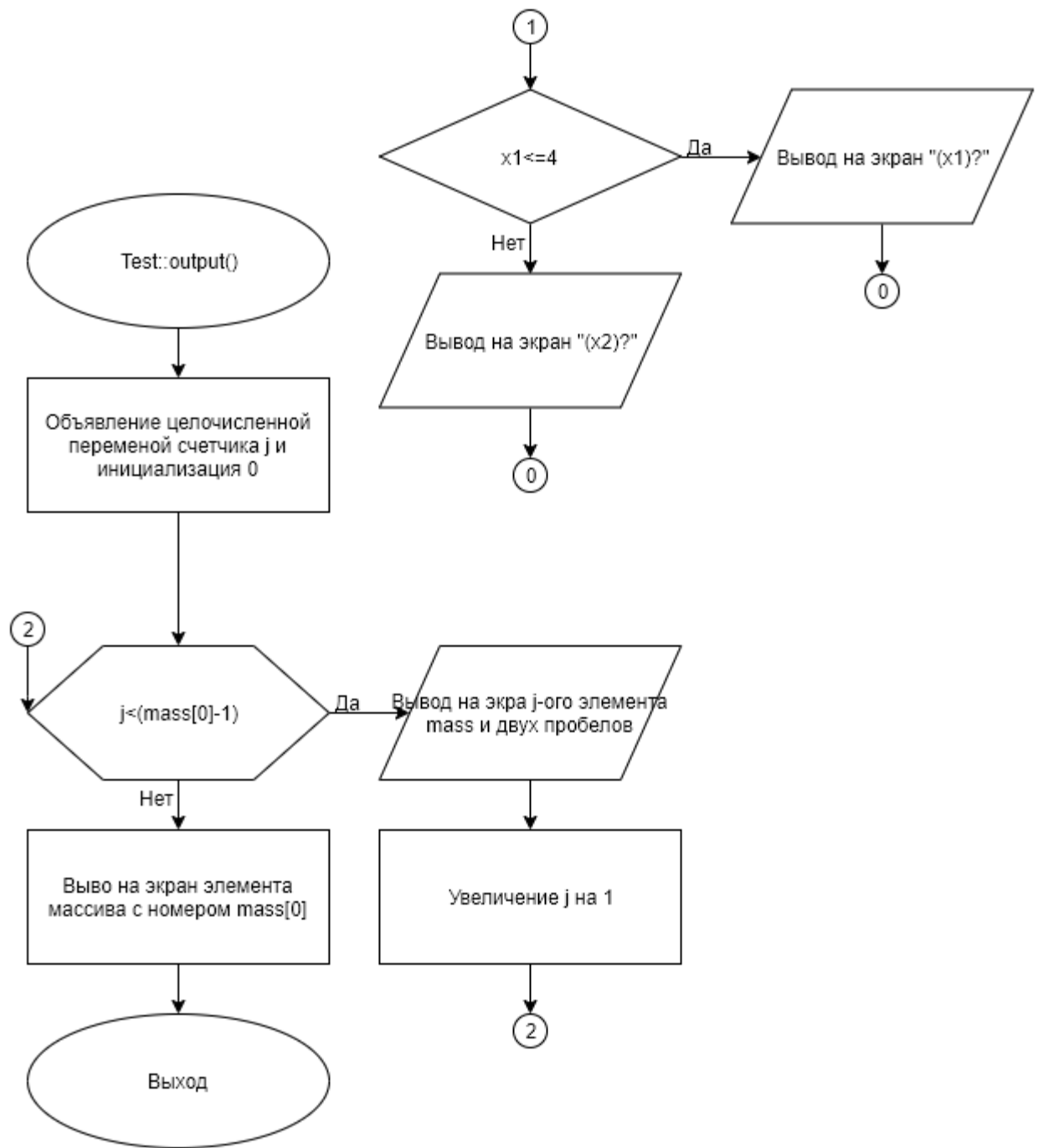


Рисунок 2 – Блок-схема алгоритма

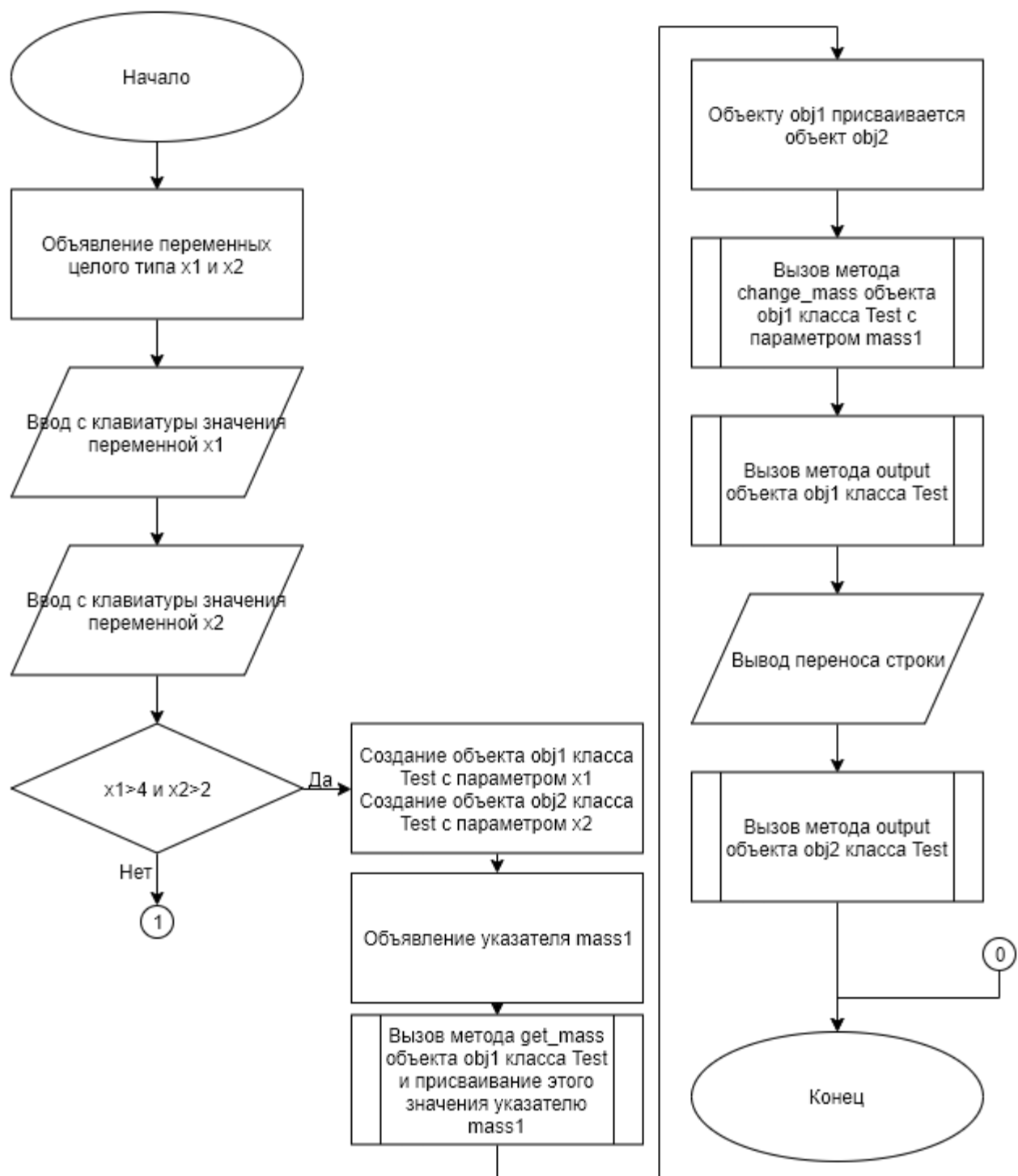


Рисунок 3 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

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

using namespace std;

int main()
{
    int x1, x2;
    cin >> x1;
    cin >> x2;
    if (x1 > 4 && x2 > 4){
        Test obj1(x1), obj2(x2);
        int* mass1 = obj1.get_mass();
        obj1 = obj2;
        obj1.change_mass(mass1);
        obj1.output();
        cout << endl;
        obj2.output();
    }
    else if (x1 <= 4){
        cout << x1 <<"?";
    }
    else {
        cout << x2 <<"?";
    }
}
```

### 5.2 Файл Test.cpp

*Листинг 2 – Test.cpp*

```
#include "Test.h"
```

```

Test::Test(int i)
{
    mass = new int[i];
    for (int j = 0; j < i; j++) {
        mass[j] = i;
    }
}

int* Test::get_mass()
{
    return(mass);
}

void Test::change_mass(int* x)
{
    mass = x;
}

void Test::output()
{
    for (int j = 0; j < mass[0] - 1; j++) {
        cout << mass[j] << " ";
    }
    cout << mass[0];
}

Test::~Test()
{
    delete[] mass;
}

```

## 5.3 Файл Test.h

*Листинг 3 – Test.h*

```

#ifndef __TEST__H
#define __TEST__H
#include <iostream>

using namespace std;

class Test
{
private:
    int* mass;

public:
    Test(int i);

```

```
~Test();  
int* get_mass();  
void change_mass(int* x);  
void output();  
};  
  
#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 8	5 5 5 5 5 8 8 8 8 8 8 8 8	5 5 5 5 5 8 8 8 8 8 8 8 8
1 5	1?	1?
6 4	4?	4?
6 7	6 6 6 6 6 6 7 7 7 7 7 7 7	6 6 6 6 6 6 7 7 7 7 7 7 7
1 2	1?	1?
6 5	6 6 6 6 6 6 5 5 5 5 5	6 6 6 6 6 6 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).