Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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 myClass предназначен для демонстрация возможности объектов ;
- объект obj2 класса myClass предназначен для демонстрация возможности объектов.

Класс myClass:

- функционал:
    - метод get — возврат указателя на массив;
    - метод set — установка размера указателя, а так же заполнение его значений.

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 метода get класса myClass

Функционал: возврат указателя на массив.

Параметры: нет.

Возвращаемое значение: int* - указатель на массив.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода get класса myClass

№	Предикат	Действия	№ перехода
1		возвращение указателя Array на массив целых значений	∅

3.2 Алгоритм метода set класса myClass

Функционал: установка размера указателя, а также заполнение его значениями.

Параметры: int* - указатель на массив целых значений.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода set класса myClass

№	Предикат	Действия	№ перехода
1		присвоению полю Array указатель x	∅

3.3 Алгоритм функции main

Функционал: демонстарция возможности объектов одного класса присвоения значения друг друга.

Параметры: нет.

Возвращаемое значение: int - индикатор успешности выполнения программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленных переменных a,b	2
2		ввод значения a	3
3	a > 4		4
		вывод "a?"	13
4		ввод значения b	5
5	b > 4	создание объекта obj1 класса myClass с параметром a	6
		вывод "b?"	13
6		создание объекта obj2 класса myClass с параметром b	7
7		инициализация указателя pArray посредством вызова метода get	8
8		присвоение obj1 значения obj2	9
9		вызов метода set объекта obj1 с передачей указателя pArray	10
10		вызов метода out объекта obj1	11
11		переход на новую строку	12
12		вызов метода out объекта obj2	13
13		возврат 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

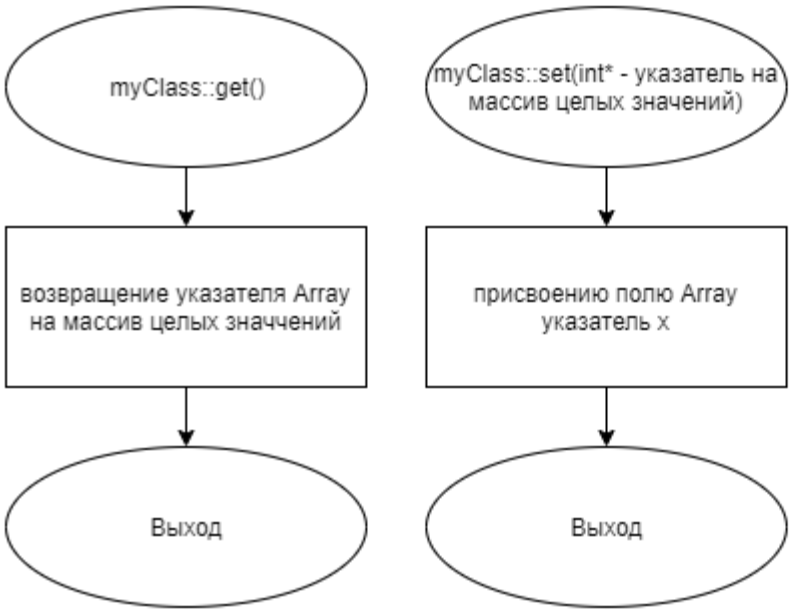

Рисунок 1 – Блок-схема алгоритма

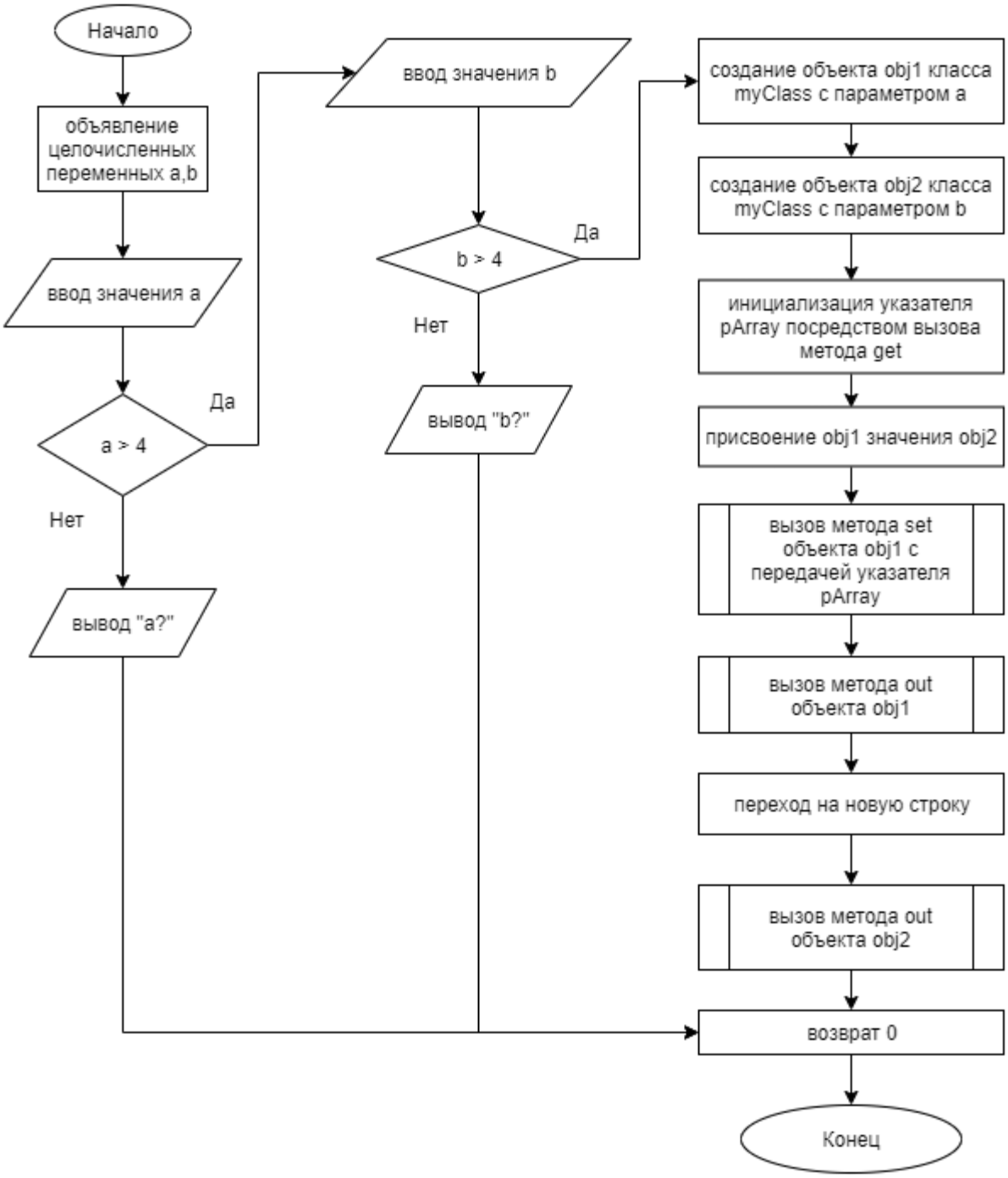

Рисунок 2 – Блок-схема алгоритма

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 "myClass.h"

int main()
{
   int a, b;
   std::cin >> a;
   if (a > 4){
      std::cin >> b;
      if (b > 4){
         myClass obj1(a);
         myClass obj2(b);
         int* pArray = obj1.get();
         obj1 = obj2;
         obj1.set(pArray);
         obj1.out();
         std::cout << "\n";
         obj2.out();
      }
      else{
         std::cout << b << "?\n";
      }
   }
   else{
      std::cout << a << "?\n";
   }
   return 0;
}
```

5.2 Файл myClass.cpp

Листинг 2 – myClass.cpp

```
#include "myClass.h"
#include <iostream>

myClass::myClass(int x){
   Array = new int[x];
   for (int i = 0; i < x; i++){
      Array[i] = x;
   }
};
void myClass::out(){
   for (int i = 0; i < Array[0]-1; i++){
      std::cout << Array[i] << "  ";
   }
   std::cout << Array[0];
}
int* myClass::get(){
   return Array;
}
void myClass::set(int* x){
   Array = x;
}
myClass::~myClass(){
   delete[]Array;
}
```

5.3 Файл myClass.h

Листинг 3 – myClass.h

```
#ifndef __MYCLASS__H
#define __MYCLASS__H

class myClass{
public:
   myClass(int x);
   int* get();
   void set(int* x);
   void out();
   ~myClass();
private:
   int* Array = nullptr;
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**
5 6	5 5 5 5 5 6 6 6 6 6 6	5 5 5 5 5 6 6 6 6 6 6

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-avrora.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-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

4. Шилдт Г. C++: 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).