**Сопроводительный лист на издание в авторской редакции**

Название работы Двумерные массивы Методические указания к выполнению лабораторной работы № 17 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 «Информатика и вычислительная техника», 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Актуальность и соответствующий   
научно-методический уровень подтверждаю \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись научного редактора)

Рукопись сверена и проверена автором \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись автора)

Рекомендуется к изданию \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись заведующего кафедрой)



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

Утверждаю

Ректор университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Н. Федонин

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018

ПРОГРАММИРОВАНИЕ

ДВУМЕРНЫЕ МАССИВЫ

**Методические указания**

**к выполнению лабораторной работы № 17**

**для студентов очной формы обучения**

**по направлениям подготовки**

**09.03.04 – «Программная инженерия»,**

**09.03.01 «Информатика и вычислительная техника»,**

**02.03.03 «Математическое обеспечение и администрирование информационных систем»**

**Брянск 2018**

**УДК 004.**

Двумерные массивы [Текст] + [Электронный ресурс]: методические указания к выполнению лабораторной работы № 17 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 «Информатика и вычислительная техника», 02.03.03 «Математическое обеспечение и администрирование информационных систем». – Брянск, 2018. – 7 с.

Разработали:

Д. И. Булатицкий

канд. техн. наук, доц.,

Е..В. Коптенок

ассистент

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ

**Методические издания публикуются в авторской редакции**

# Цель работы

Цель работы – научиться работать с двумерными массивами.

Продолжительность работы – 2 часа.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

До этого момента, мы рассматривали только одномерные массивы, то есть, к элементу массива мы обращались через один индекс. Однако, массивы могут быть и двумерными и трехмерными и, даже, n-мерными. Многомерные массивы — это массивы, у которых есть более одного индекса. Вместо одной строки элементов, многомерные массивы можно рассматривать как совокупность элементов, которые распределены по двум или более измерениям.

С позволяет создавать многомерные массивы. Простейшим видом многомерного массива является двумерный массив. Двумерный массив - это массив одномерных массивов. Двумерный массив объявляется следующим образом:

тип имя\_массива[размер второго измерения][размер первого измерения];

Следовательно, для объявления двумерного массива целых с размером 10 на 20 следует написать:

int d[10] [20] ;

Объявление двумерного массива почти ничем не отличается от объявления одномерного, за исключением того, что при объявлении двумерного массива, нужно указывать размер каждого измерения в квадратных скобочках. Например, давайте объявим двумерный массив размером 8\*8, это размер поля для стандартных шашек — 8 строк и 8 столбцов:

int checkers[8][8]; // двумерный массив

Посмотрим внимательно на это объявление. В противоположность другим компьютерным языкам, где размерности массива отделяются запятой, С помещает каждую размерность в отдельные скобки.

Для доступа к элементу с индексами 3, 5 массива d следует использовать d[3] [5].

Двумерные массивы сохраняются в виде матрицы, где первый индекс отвечает за строку, а второй -за столбец. Это означает, что правый индекс изменяется быстрее левого, если двигаться по массиву в порядке расположения элементов в памяти. На рис. показано графическое представление двумерного массива в памяти. Левый индекс можно рассматривать как указатель на строку.

Число байт в памяти, требуемых для размещения двумерного массива, вычисляется следующим  образом:

число байт = размер второго измерения \* размер первого измерения \* sizeof(базовый тип)

Предполагая наличие в системе 2-байтных целых, целочисленный массив с размерностями 10 на 5 будет занимать 10 \* 5 \* 2, то есть 100 байт.

Когда двумерный массив используется как аргумент функции, передается указатель на первый элемент. Функция, получающая двумерный массив, должна, как минимум, определять размер первого измерения, поскольку компилятору необходимо знать длину каждой строки для корректной  индексации массива. Например, функция, получающая двумерный целочисленный массив с размерностями 5, 10, будет объявляться следующим образом:

void func1 (int х[] [10])  
{  
...  
}

Можно определить размер второго измерения, но это не обязательно. Компилятору нужно знать размер первого измерения для правильного выполнения операторов типа.

# ХОД РАБОТЫ

**Пример1.** В следующем примере в двумерный массив заносятся числа от 1 до 12, после чего массив выводится на экран.

#include <stdio.h>

int main(void)

{

int t,i, num[3][4];

/\* загрузка чисел \*/

for(t=0; t<3; ++t)

for (i=0; i<4; ++i)

num[t][i] = (t\*4)+i+1;

/\* вывод чисел \*/

for (t=0; t<3; ++t)

{

for (i=0; i<4; ++i)

printf("%d ",num[t][i]);

printf ("\n");

}

return 0;

}

В данном примере num[0][0] имеет значение 1, num[0][1] имеет значение 2, num[0][2] - 3 и так далее. num[2][3] имеет значение 12.

# ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

**Задание 1. Реализовать следующие функции для работы с массивом:**

1. Ввод массива:
   1. С клавиатуры
   2. Из файла (текстового)
   3. Случайными числами из диапазона
   4. По заданной формуле согласно варианту
2. Вывод массива:
   1. На экран
   2. В файл (текстовый)
3. Оформить листинг лабораторной работы с использованием заголовочных файлов.

Формулы для заполнения массива:

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. 
15. 
16. 
17. 
18. 
19. 
20. 
21. 
22. 
23. 
24. 
25. 
26. 
27. 

**Выполнить над полученным массивом:**

1. Найти среднее арифметическое k-й строки.
2. Найти сумму элементов k-го столбца.
3. Найти сумму наибольшего и наименьшего элементов k-й строки.
4. Найти сумму каждого столбца. Среди полученных сумм найти максимальное значение.
5. Найти сумму каждой строки. Среди полученных сумм найти максимальное значение.
6. В полученной матрице найти среднее арифметическое элементов столбца, в котором находится максимальный элемент матрицы В.
7. Найти число элементов матрицы, лежащих вне интервала (0.4;0.8).
8. Найти количество отрицательных чисел в каждой строке
9. Найти сумму элементов, произведение индексов которых кратно трем
10. Найти максимальный среди минимальных значений по строке
11. Найти среднее арифметическое неотрицательных элементов матрицы
12. Подсчитать, сколько отрицательных элементов в каждой отдельно взятой строке матрицы.
13. Изменить матрицу, заменив каждый отрицательный элемент, лежащий выше главной диагонали, его абсолютной величиной.
14. Найти сумму элементов главной диагонали
15. Найти среднее арифметическое элементов матрицы.
16. Найти сумму элементов тех строк матрицы, в которых отрицателен элемент главной диагонали
17. Найти среднее арифметическое элементов Aij, лежащих выше и ниже главной диагонали матрицы, для которых выполняется условие Aii<Aij
18. Найти среднее арифметическое всех тех элементов матрицы, у которых сумма индексов четная.
19. Найти минимальный элемент, лежащий ниже побочной диагонали.
20. Найти сумму значений квадратов элементов побочной диагонали.
21. Найти минимальный элемент, лежащий выше главной диагонали, и максимальный элемент, лежащий ниже главной диагонали.
22. Если в чётной строке матрицы есть хотя бы один элемент, равный заданному числу, то в следующей нечётной строке найти минимальный элемент среди тех, значение которых попадает на отрезок [C, D].
23. Найти сумму каждой четной строки. Среди полученных сумм найти минимальное значение.
24. В полученной матрице найти среднее арифметическое элементов столбца, в котором находится минимальный элемент матрицы В.
25. Найти число элементов матрицы, принадлежащих интервалу (0.4;0.8).
26. Найти количество отрицательных чисел в каждом нечетном столбце
27. Поменять местами строки матрицы, которые содержат наибольший и наименьший элементы.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое двумерный массив?
2. Каким образом происходит адресация элементов в двумерном массиве?
3. Как передать двумерный массив в функцию?
4. Как организовать перебор всех элементов двумерного массива?

# Список рекомендуемой литературы

## Основная литература

1. Павловская Т.А. C#. Программирование на языке высокого уровня:учеб. Для вузов.-м. [и др.]:Питер,2014.-432
2. Давыдова Н.А.,боровская Е.В. Программирование: учеб. Пособие. М.:Бином. Лаб. Знаний,2011.-238 с.
3. Булатицкий Д.И. Объектно-ориентированное программирование: учеб. пособие.- Брянск: изд-во БГТУ,2008.-120 с.

## Дополнительная литература

1. Тюльпинова Н.В. Алгоритмизация и программирование: учеб. Пособие/Брян. Гос. Техн. Ун-т.- Брянск:изд-во БГТУ,2012.-199 с.-[+ электронная копия].
2. Иванова Г.С. Программирование: основы алгоритмизации и процедурное программирование, объектно ориентированное программирование: учеб. Для вузов.-2-е изд., стер.-м.:Кнорус, 2014.-425 с.

Двумерные массивы [Текст] + [Электронный ресурс]: методические указания к выполнению лабораторной работы № 17 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 «Информатика и вычислительная техника», 02.03.03 «Математическое обеспечение и администрирование информационных систем». – Брянск, 2018. – 7 с.

БУЛАТИЦКИЙ ДМИТРИЙ ИВАНОВИЧ

КОПТЕНОК ЕЛИЗАВЕТА ВИКТОРОВНА

Научный редактор В.Я. Израилев

Компьютерный набор Е.В. Коптенок

Иллюстрации Е.В. Коптенок

Подписано в печать 20.02.2017. Формат 60х84 1/15 Бумага офсетная. Офсетная печать. Усл.печ.л. \_\_\_ Уч.-изд.л. \_\_\_ Тираж 1 экз. Бесплатно

Брянский государственный технический университет

Кафедра «Информатика и программное обеспечение», тел. 56-09-84

241035, Брянск, бульвар 50 лет Октября, 7 БГТУ, тел. 58-82-49