

Лабораторная работа №5

Тема: «Динамические массивы»

Цель работы: сформировать навыки и умения обработки структурированных типов данных, организованных в виде матрицы.

Время выполнения: 4 часа.

Теоретические сведения

Кроме отдельных динамических объектов в языке C++ мы можем использовать динамические массивы. Для выделения памяти под динамический массив также используется оператор `new`, после которого в квадратных скобках указывается, сколько массив будет содержать объектов:

```
int *numbers {new int[4]}; // динамический массив из 4 чисел
// или так
// int *numbers = new int[4];
```

Причем в этом случае оператор `new` также возвращает указатель на объект типа `int` - первый элемент в созданном массиве.

В данном случае определяется массив из четырех элементов типа `int`, но каждый из них имеет неопределенное значение. Однако мы также можем инициализировать массив значениями:

```
int *numbers1 {new int[4]{} }; // массив состоит из чисел 0, 0, 0, 0
int *numbers2 {new int[4]{ 1, 2, 3, 4 } }; // массив состоит из чисел 1, 2, 3,
4
int *numbers3 {new int[4]{ 1, 2 } }; // массив состоит из чисел 1, 2, 0, 0
// аналогичные определения массивов
// int *numbers1 = new int[4]{}; // массив состоит из чисел 0, 0, 0,
0
// int *numbers1 = new int[4](); // массив состоит из чисел 0, 0, 0, 0
// int *numbers2 = new int[4]{ 1, 2, 3, 4 }; // массив состоит из чисел 1, 2,
3, 4
// int *numbers3 = new int[4]{ 1, 2 }; // массив состоит из чисел 1, 2, 0,
0
```

При инициализации массива конкретными значениями следует учитывать, что если значений в фигурных скобках больше чем длина массива, то оператор `new` потерпит неудачу и не сможет создать массив. Если переданных значений, наоборот, меньше, то элементы, для которых не предоставлены значения, инициализируются значением по умолчанию.

Стоит отметить, что в стандарт C++20 добавлена возможность вывода размера массива, поэтому, если применяется стандарт C++20, то можно не указывать длину массива:

```
int *numbers {new int[] { 1, 2, 3, 4 } }; // массив состоит из чисел 1, 2, 3, 4
```

После создания динамического массива мы сможем с ним работать по полученному указателю, получать и изменять его элементы:

```
int *numbers {new int[4]{ 1, 2, 3, 4 } };
```

```
// получение элементов через синтаксис массивов
```

```
std::cout << numbers[0] << std::endl;    // 1
```

```
std::cout << numbers[1] << std::endl;    // 2
```

```
// получение элементов через операцию разыменования
```

```
std::cout << *numbers << std::endl;      // 1
```

```
std::cout << *(numbers+1) << std::endl;  // 2
```

Причем для доступа к элементам динамического массива можно использовать как синтаксис массивов (`numbers[0]`), так и операцию разыменования (`*numbers`)

Соответственно для перебора такого массива можно использовать различные способы:

```
unsigned n{ 5 }; // размер массива
```

```
int* p{ new int[n] { 1, 2, 3, 4, 5 } };
```

```
// используем индексы
```

```
for (unsigned i{ }; i < n; i++)
```

```
{
```

```
    std::cout << p[i] << "\t";
```

```
}
```

```
std::cout << std::endl;
```

```
// добавляем к адресу в указателе смещение
```

```
for (unsigned i{ }; i < n; i++)
```

```
{
```

```
    std::cout << *(p+i) << "\t";
```

```
}
```

```
std::cout << std::endl;
```

```
// проходим по массиву с помощью вспомогательного указателя
for (int* q{ p }; q != p + n; q++)
{
    std::cout << *q << "\t";
}
std::cout << std::endl;
```

Обратите внимание, что для задания размера динамического массива мы можем применять обычную переменную, а не константу, как в случае со стандартными массивами.

Для удаления динамического массива и освобождения его памяти применяется специальная форма оператора delete:

```
delete [] указатель_на_динамический_массив;
Например:
```

```
#include <iostream>
```

```
int main()
{
    unsigned n{ 5 }; // размер массива
    int* p{ new int[n] { 1, 2, 3, 4, 5 } };

    // используем индексы
    for (unsigned i{ }; i < n; i++)
    {
        std::cout << p[i] << "\t";
    }
    std::cout << std::endl;
```

```
    delete [] p;
}
```

Чтобы после освобождения памяти указатель не хранил старый адрес, также рекомендуется обнулить его:

```
delete [] p;
p = nullptr; // обнуляем указатель
Многомерные массивы
```

Также мы можем создавать многомерные динамические массивы. Рассмотрим на примере двумерных массивов. Что такое по сути двумерный массив? Это набор массив массивов. Соответственно, чтобы создать динамический двумерный массив, нам надо создать общий динамический массив указателей, а затем его элементы - вложенные динамические массивы. В общем случае это выглядит так:

```
#include <iostream>

int main()
{
    unsigned rows = 3;    // количество строк
    unsigned columns = 2; // количество столбцов
    int** numbers{new int*[rows]{} }; // выделяем память под двумерный
массив
    // выделяем память для вложенных массивов
    for (unsigned i{ }; i < rows; i++)
    {
        numbers[i] = new int[columns]{};
    }

    // удаление массивов
    for (unsigned i{ }; i < rows; i++)
    {
        delete[] numbers[i];
    }
    delete[] numbers;
}
```

Вначале выделяем память для массива указателей (условно таблицы):

```
1
int** numbers{new int*[rows]{} };
```

Затем в цикле выделяем память для каждого отдельного массива (условно строки таблицы):

```
1
numbers[i] = new int[columns]{};
```

Освобождение памяти идет в обратном порядке - сначала освобождаем память для каждого отдельного вложенного массива, а затем для всего массива указателей.

Пример с вводом и выводом данных двухмерного динамического массива:

```
#include <iostream>

int main()
{
    unsigned rows = 3;    // количество строк
    unsigned columns = 2; // количество столбцов
    int** numbers{new int*[rows]{}}; // выделяем память под двухмерный
массив
    for (unsigned i{ }; i < rows; i++)
    {
        numbers[i] = new int[columns]{};
    }

    // вводим данные для таблицы rows x columns
    for (unsigned i{ }; i < rows; i++)
    {
        std::cout << "Enter data for " << (i + 1) << " row" << std::endl;
        // вводим данные для столбцов i-й строки
        for (unsigned j{ }; j < columns; j++)
        {
            std::cout << (j + 1) << " column: ";
            std::cin >> numbers[i][j];
        }
    }

    // вывод данных
    for (unsigned i{ }; i < rows; i++)
    {
        // выводим данные столбцов i-й строки
        for (unsigned j{ }; j < columns; j++)
        {
            std::cout << numbers[i][j] << "\t";
        }
        std::cout << std::endl;
    }

    for (unsigned i{ }; i < rows; i++)
```

```

    {
        delete[] numbers[i];
    }
    delete[] numbers;
}

```

Индивидуальные задания к лабораторной работе №5

Динамические массивы

1. Дан двумерный массив 5×5 . Заменить нулем элементы, расположенные под главной диагональю.
2. Дан двумерный массив целых чисел. Вычислить сумму элементов, расположенных на главной и побочной диагоналях.
3. Дан двумерный массив целых чисел. Найти произведение максимального и минимального элементов.
4. Написать программу, которая вычисляет сумму элементов двумерного массива по столбцам.
5. Найти среднее арифметическое элементов двумерного массива.
6. Написать программу, которая определяет номер строки квадратной матрицы, сумма элементов которой максимальна (минимальна).
7. Дан массив размерностью $n \times m$. Поделить все элементы каждой строки на наибольший по модулю элемент этой строки.
8. Дан двумерный массив 5×5 . Заменить нулем элементы, расположенные над главной диагональю.
9. Написать программу, которая вычисляет сумму элементов двумерного массива по строкам.
10. Написать программу, которая определяет номер столбца квадратной матрицы, сумма элементов которого максимальна.
11. Дан двумерный массив 5×5 . Заменить нулем все его отрицательные элементы.
12. Дан массив размерностью $n \times m$. Поделить все элементы каждого столбца на наименьший по модулю элемент этого столбца.
13. Дан двумерный массив 5×5 . Заменить нулем элементы, расположенные над побочной диагональю.
14. Дан двумерный массив. Получить новый массив путем деления всех элементов массива на наименьший по модулю элемент.
15. Дан двумерный массив. Определить количество строк, в которых присутствует хотя бы один отрицательный элемент.
16. Дан двумерный массив. Получить новый массив путем деления всех элементов строки на наименьший по модулю элемент строки. Если

наименьший элемент некоторой строки равен 0, то элементы этой строки оставить без изменений.

17. Дан двумерный массив размерностью 5×5 . Вычислить сумму элементов, номера строк у которых четные.

18. Присвоить переменной t значение `true`, если среднее арифметическое элементов двумерного массива больше произведения элементов, находящихся на главной диагонали. В обратном случае присвоить значение `false`.

19. Дан двумерный массив размерностью $n \times m$, в котором не все элементы равны нулю. Получить новый массив путем деления всех элементов массива на его наименьший элемент. Если наименьший элемент равен нулю, то вывести соответствующее сообщение, а элементы массива оставить без изменения.

20. Дан квадратный массив размерностью 5×5 . Найти произведение элементов массива, расположенных в строках, которые начинаются с отрицательного элемента.

21. Дан двумерный массив размерностью 5×5 . Вычислить сумму элементов, номер столбца которых нечетный.

22. Дан двумерный массив вещественных чисел размерностью $n \times m$. В каждой строке найти минимальный элемент, а среди них – максимальный.

23. Найти среднее арифметическое максимального и минимального по модулю элементов массива.

24. Дан двумерный массив размерностью 5×5 . Получить новый массив путем деления всех элементов массива на наибольший по модулю элемент.

25. Дан двумерный массив размерностью 5×5 . Заменить нулевые элементы, расположенные под побочной диагональю.

26. Дан двумерный массив размерностью 5×5 . Заменить все элементы массива = C нулями. В случае если таких элементов нет, вывести соответствующее сообщение.

27. Дан двумерный массив вещественных чисел размерностью $n \times m$. В каждом столбце найти максимальный элемент, а среди них – минимальный.

28. Дан двумерный массив 5×5 . Заменить нулевые элементы, расположенные над побочной диагональю, на -1.

29. Дан двумерный массив размером $n \times m$, заполненный случайным образом. Определить, есть ли в данном массиве строки, в которых есть отрицательные элементы.

30. Дан двумерный массив размерностью 3×4 , заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен произведению четных положительных элементов.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть лабораторной работы.
2. Реализовать индивидуальное задание по вариантам, представленные в теоретических сведениях, сделать скриншоты работающих программ. Написать комментарии.
3. Написать отчет, содержащий:
 1. Титульный лист, на котором указывается:
 - а) полное наименование министерства образования и название учебного заведения;
 - б) название дисциплины;
 - в) номер практического занятия;
 - г) фамилия преподавателя, ведущего занятие;
 - д) фамилия, имя и номер группы студента;
 - е) год выполнения лабораторной работы.
 2. Индивидуальное задание из раздела «Теоретические сведения» с кодом, комментариями и скриншотами работающих программ.
 3. Построение блок-схем.