

Динамические массивы

Размер динамического массива можно задавать на этапе выполнения программы. Для создания динамического массива необходимо выделить память под массив. Синтаксис выделения памяти для массива имеет вид **указатель = new тип[размер]**. В качестве размера массива может выступать любое целое положительное значение. После завершения работы с массивом необходимо освободить эту память командой **delete [] массив**.

Пример 1. Создание динамического массива

```
#include <iostream>
using namespace std;

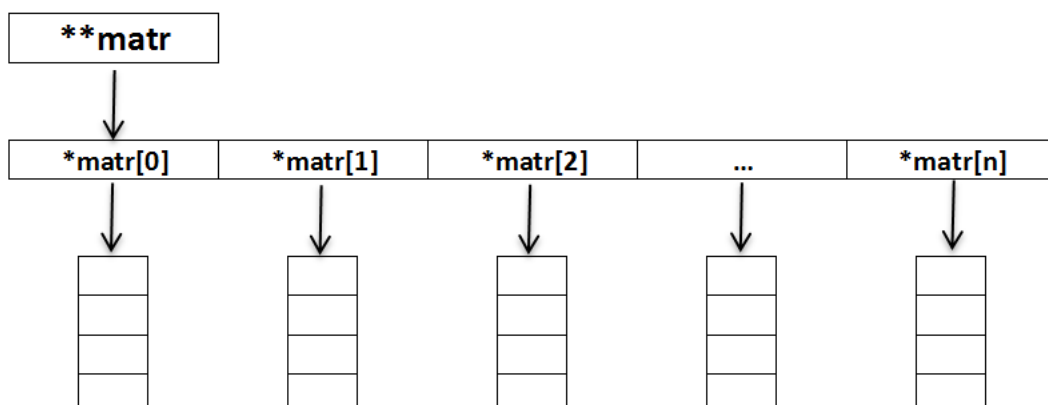
int main()
{
    int n; // размер массива
    cout << "Enter array size: ";
    cin >> n; // получение от пользователя размера массива

    int *arr = new int[n]; // Выделение памяти для массива
    for (int i = 0; i < n; i++) {
        // Заполнение массива и вывод значений его элементов
        arr[i] = i;
        cout << "a[ " << i << " ] = " << arr[i] << endl;
    }
    delete[] arr; // очистка памяти
    return 0;
}
```

1

Для создания двумерного динамического массива вначале нужно распределить память для массива указателей на одномерные массивы, а затем выделить память для одномерных массивов. При динамическом распределении памяти для массивов следует описать соответствующий указатель, которому будет присвоено значение адреса начала области выделенной памяти.

Под объявлением двумерного динамического массива понимают объявление двойного указателя, то есть объявление указателя на указатель.



Выделение памяти с помощью операции **new**:

```
int n, m; // n и m – количество строк и столбцов матрицы
//указатель для массива указателей
float **matr;
//выделение динамической памяти под массив указателей
```

```

matr = new float *[n];
//выделение динамической памяти для массива значений
for (int i = 0; i<n; i++)
    matr[i] = new float[m];

```

При выделении динамической памяти размеры массивов должны быть полностью определены.

После окончания работы с массивом необходимо освободить память, выделенную для его хранения.

Освобождение памяти:

```

//освобождает память, выделенную для массива значений
for (int i = 0; i<n; i++)
    delete matr[i];
//освобождает память, выделенную под массив указателей
delete[] matr;

```

Пример 2. Создание двумерного динамического массива, вывод на экран

```

#include <iostream>
using namespace std;

int main() {
    setlocale(0, "");
    // количество массивов с массиве
    int n = 3;
    // количество элементов в одномерном массиве
    int m = 4;
    // выделение памяти под массив
    int **arr = new int*[n];
    for (int i = 0; i < n; i++) {
        arr[i] = new int[m];
    }
    // заполнение массива
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            arr[i][j] = (i+1) * 100 + (j+1);

    // вывод массива на экран
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << " a[" << i + 1 << j + 1 << " ] = " << arr[i][j];
        }
        cout << "" << endl;
    }
    // очистка памяти
    for (int i = 0; i < n; i++)
        delete[] arr[i];
    delete[] arr;
}

```

При передаче двумерного динамического массива в функцию необходимо передавать его как указатель на указатель и передавать его размер.

Пример 3. Двумерный динамический массив как параметр функции

```

#include <iomanip>
#include <iostream>
using namespace std;

int** newArr(int n, int m)
{
    int **arr;
    arr = new int *[n];
    for (int i = 0; i < n; i++)
        arr[i] = new int[m];
}

```

```
        return arr;
    }

void initArr(int** arr, int n, int m)
{
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            arr[i][j] = i + j + 1;
}

void printArr(int** arr, int n, int m)
{
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++)
            cout << setw(4) << arr[i][j];
        cout << endl;
    }
}

int sumArr(int** arr, int n, int m)
{
    int s = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            s += arr[i][j];

    return s;
}

void deleteArr(int** arr, int n)
{
    for (int i = 0; i < n; i++)
        delete[] arr[i];
    delete[] arr;
}

int main() {
    int n = 3;
    int m = 4;

    int **arr = newArr(n, m);
    initArr(arr, n, m);
    printArr(arr, n, m);
    int s = sumArr(arr, n, m);
    cout << s << endl;
    deleteArr(arr, n);
}
```

Задания

1. Написать функции для одномерного динамического массива, которые выделяют память, заполняют массив случайными целыми числами в диапазоне от -50 до 50, выводят массив на экран.
2. Написать функцию, которая дублирует каждый элемент одномерного динамического массива и возвращает новый массив и его размерность.
3. Написать функцию, которая из заданного одномерного динамического массива формирует два массива – в первом четные элементы, а во втором - нечетные. Функция должна возвращать кроме массивов его размерности.

4. Написать функцию, которая возвращает массив индексов отрицательных элементов исходного массива. Функция должна возвращать кроме массива его размерность.
5. Написать функцию, которая оставляет в одномерном массиве уникальные элементы, удаляя повторяющиеся элементы. Функция должна возвращать новый массив и его размерность.
6. Написать функции для двумерного динамического массива, которые выделяют память, заполняют массив случайными целыми числами в диапазоне от -50 до 50, выводят массив на экран.
7. Написать функцию, которая возвращает главную и побочную диагональ двумерного квадратного массива.
8. Написать функцию, которая меняет в двумерном динамическом массиве строку k со строкой p .
9. Написать функцию, которая перемножает две квадратные матрицы.
10. Написать функцию, которая транспонирует матрицу.