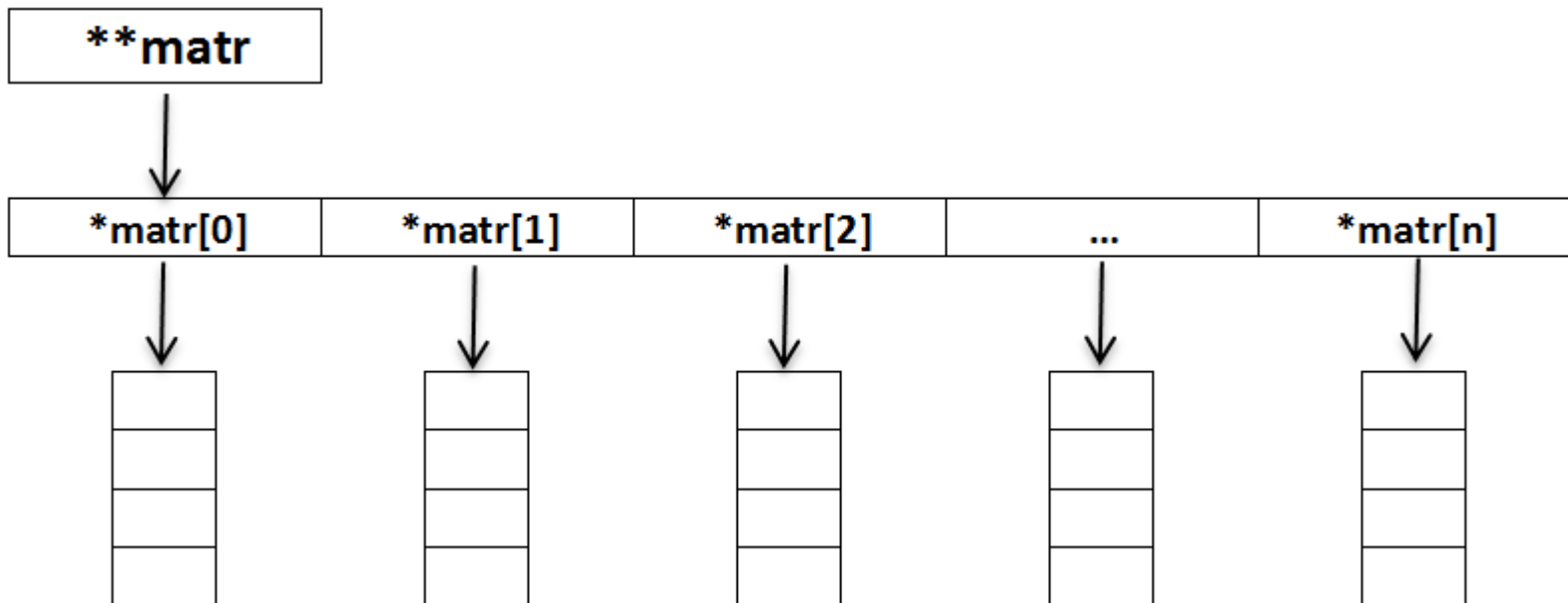


Динамические двумерные массивы

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



Динамические двумерные массивы

Для создания двумерного динамического массива вначале нужно распределить память для массива указателей на одномерные массивы, а затем выделить память для одномерных массивов.

```
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];
```

Динамические двумерные массивы

При выделении динамической памяти размеры массивов должны быть **полностью определены**.

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

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

```
//освобождает память, выделенную под массив указателей  
delete[] matr;
```

Динамические двумерные массивы. Пример 1. Часть 1

```
#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];
    }
}
```

Динамические двумерные массивы. Пример 1. Часть 2

```
// 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;
}
```

Динамические двумерные массивы. Пример 2. Часть 1

```
#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];
    }
}
```

Динамические двумерные массивы. Пример 2. Часть 1

```
// 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;}
```

Динамические двумерные массивы. Передача в функцию

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

```
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;
    }
}
```


Передача массива в функцию. Пример. Часть 1

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

// Выделение памяти для массива
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;
}
```

Передача массива в функцию. Пример. Часть 2

```
// Инициализация массива
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;
}

// Печать массива
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;
    }
}
```

Передача массива в функцию. Пример. Часть 3

```
// Сумма элементов массива
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;
}

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

    delete[]arr;
}
```

Передача массива в функцию. Пример. Часть 4

```
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);  
}
```