Занятие 4.7 Массивы

- Декларация массивов и их размещение в памяти.
- Индексация элементов массива.
- Алгоритмы суммирования, поиска и сортировки.

Массивы

- **Массив (индексный)** — простая статическая структура данных, предназначенная для хранения набора элементов данных, каждый из которых идентифицируется индексом или набором индексов.

- **Индекс** — обычно целое число, либо значение типа, приводимого к целому, указывающее на конкретный элемент массива.

- Количество используемых индексов массива может быть различным. Массивы с одним индексом называют *одномерными*, с двумя — *двумерными* и т. д.

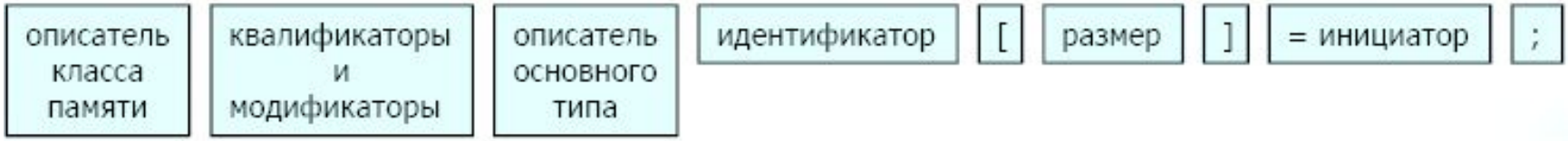
- Одномерный массив соответствует вектору в математике, двумерный — матрице.

Массивы

- **Специфические типы массивов**
  - **Статические массивы.**
    - Статическим называется массив, размер которого не может изменяться во время исполнения программы.
    - *В языке С встроенные массивы являются* ***статическими****.*
  - **Динамические массивы.**
    - Динамическим называется массив, размер которого может меняться во время исполнения программы.
    - Для реализации динамики язык программирования должен предоставлять встроенную поддержку.
    - *Язык С не имеет встроенной поддержки динамических массивов, но ее можно реализовать* ***функциями управления динамической памятью****.*

Декларация массивов и их размещение в памяти

- Оператор описания массива состоит из следующих компонентов:



- Обязательными являются
  - ***идентификатор с квадратными скобками,***
  - ***хотя бы один из предшествующих описателей,***
  - ***точка с запятой***
- Если несколько идентификаторов имеют одинаковый набор описателей, то их можно объединить в одном операторе описания, причем каждый из них может иметь свой инициатор (в т.ч. можно объединять описания массивов и переменных).
  - Пример:

```
int i, a[5];
```

для хранения элементов этого массива резервируется непрерывный участок памяти объемом **5*sizeof(int)** байт.

- Количество элементов массива в С задается *целой константой*, обычно определенной при помощи директивы препроцессора **#define**:

```
#define  SIZE  5
int a[SIZE];
```

Декларация массивов и их размещение в памяти

- Массивы можно *инициализировать при их определении*, заключая список инициаторов в фигурные скобки:

```
int a[SIZE]={2,4,6,8,10}, a0[SIZE]={0};
```

  - Если инициаторов в списке больше, чем элементов в массиве, - это *ошибка*.
  - Если инициирующих значений в списке меньше, чем элементов в массиве, то оставшиеся элементы массива получают *нулевое* значение.
- При инициализации массива можно в определении *не указывать размер* массива, тогда компилятор сам определит размер исходя из числа инициирующих значений:

```
int b[]={1,1,2,3,5,8};
```

  В этом случае массив b будет иметь 6 элементов.

Индексация элементов массива

- Имя массива в С трактуется как *указатель-константа* на начало памяти, отведенной для хранения элементов массива.
- Доступ к отдельным элементам осуществляется *индексированием* имени, причем индексация начинается с *нуля*. Следовательно,
  - первый элемент имеет обозначение **a[0]**,
  - второй - **a[1]**,
  - последний - **a[SIZE-1]**.
- В С нет встроенных средств анализа *выхода индекса за границы массива* ни на этапе компиляции, ни на этапе выполнения программы - вся ответственность за организацию правильной работы возлагается на программиста.
  - Наиболее частая ошибка состоит в присваивании значения несуществующему элементу **a[SIZE]**, что приводит к ***порче других объектов программы***.
  - Типичный цикл для перебора всех элементов массива:

```
for(i=0; i<SIZE; ++i)
{
  … a[i] …
```

Практика

- Напишите программу, в которой массив размера N заполняется случайными числами.

  Требуется вывести эти числа и их сумму на экран

Символьные массивы

- Массивы, состоящие из элементов типа **char**, играют в С особую роль – они используются для представления *символьных строк*, т.к. самостоятельного типа "строка символов" (string) в С нет.
- Каждый символ строки представляется отдельным элементом символьного массива, причем непосредственно после последнего символа строки в массиве должен находиться нулевой код **'\0'**, который служит ограничителем строки.
  - Таким образом, для представления строки, состоящей из **n** символов, массив должен иметь размер не менее чем **n+1**:

```
char text[6]={'H', 'e', 'l','l','o','\0'};
```

    Массив **text** содержит строку **"Hello"**.

- Поскольку инициализация символьных массивов значениями строки символов используется очень часто, то в С для нее разрешена более простая и удобная форма:

```
char text[]="Hello";
```

  - Как и ранее, размер массива определяется в этом случае компилятором по результату инициализации, причем нулевой код автоматически добавляется в массив для ограничения строки в соответствии с принятым соглашением.

Многомерные массивы

- В С допускается описание не только одномерных, но также и многомерных массивов.
  - Наиболее часто используются двумерные массивы, которые можно трактовать как *матрицы*, а массивы размерности выше двух применяются гораздо реже, поскольку требуют для своего хранения значительного объема памяти.
- При описании многомерных массивов размер по каждому измерению заключается в *отдельные квадратные скобки*, например:

```
int a[3][4];
```

  - Такая конструкция рассматривается в С следующим образом: массив **a** имеет три элемента **a[0]**, **a[1]** и **a[2]**, каждый из которых в свою очередь является массивом из четырех целых чисел.

Многомерные массивы

- Как и одномерные, многомерные массивы можно *инициализировать при их определении:*

```
int a[3][4]={ {5,3,-21,42}, {44,15,0,6}, {97,6,81,2} };
```

- Внутренние фигурные скобки можно опустить, кроме того, как и ранее, разрешено опускать размер массива *по первому измерению*:

```
int a[][4]={5,3,-21,42,44,15,0,6,97,6,81,2};
```

тогда пропущенный размер определяется компилятором автоматически по длине инициирующего списка.

- Допускается инициализация лишь *части элементов* многомерного массива

```
int a[3][4]={{1,2},{3,4},{5,6}};
```

В этом примере инициализированы первые два столбца матрицы:
элементы первого столбца имеют нечетные значения,
элементы второго - четные,
а оставшиеся два столбца имеют нулевые значения.

- Если внутренние скобки опустить:

```
int a[3][4]={1,2,3,4,5,6};
```

то элементы первой строки получат значения 1,2,3,4,
первые два элемента второй строки - значения 5 и 6,
а остальные элементы матрицы будут нулевыми.

Практика

- Посмотрите длину сообщения
- Добавьте в конец первой строки вторую
- Запишите 3 сообщения в массив и выведите их на экран
- Вывести номер элемента **x**, который введен с клавиатуры.

Задание на дом

1.Написать программу, которая вводит с клавиатуры в одномерный массив 5 целых чисел, после чего выводит количество ненулевых элементов.

2. Написать программу, которая вычисляет среднее арифметическое ненулевых элементов введенного с клавиатуры массива целых чисел.

3. Написать программу, которая вычисляет среднее арифметическое элементов массива без учета минимального и максимального элементов массива.