C++ course

name: Виктория Александровна e-mail: kewtree1408@gmail.com

source: https://bitbucket.org/Kewtree/coursecpp/src

login: coursecpp password: 12345

General Plans

- 1. Inception
- 2-3. Statements
- 4. Functions
- 5. Pointers
- 6. Arrays
- 7. Structures
- 8. Files, OOP: encapsulation
- 9. OOP: inheritance
- 10. OOP: polimorph
- 11. Test
- 12. Game over

Arrays

- 1. Definitions
- 2. Link with pointers
- 3. Multidimensional array
- 4. Dynamic memory: new & delete
- 5. Dynamic array and matrix
- 6. Sort
- 7. Strings

Объявления массивов

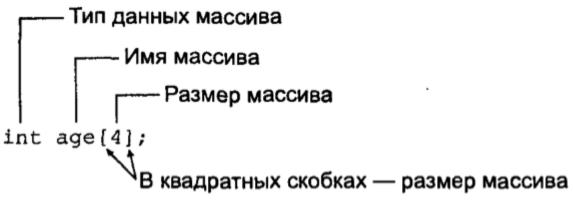


Рис. 7.1. Синтаксис определения массива

Нумерация индексов массивов в С++ всегда начинается с о!

Инициализация массивов

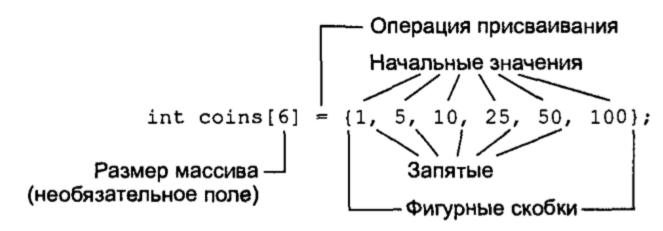


Рис. 7.3. Синтаксис инициализации массива

Нельзя целиком присваивать один массив другому.

```
init_mass.cpp
#include <iostream> // подключение заголовочного файла
using namespace std;
int main() {
    //~ объявление массивов
     float vec[10];
                                     //диапазон индексов 0..9
     short int Prise[12];
                                     //диапазон индексов 0..11
     int nArr[3];
                                     //создается массив из трех элементов
     nArr[0] = 7;
                       //присвоение значения первому элементу
     nArr[1] = 8;
     nArr[2] = 9;
    //~ объявление и инициализация массива
     int Days[12] = \{31,28,31,30,31,30,31,30,31,30,31\};
     //~ что лежит в НЕ инициализированном массиве?
     cout <<"1) NON initializing array vec:\n";</pre>
     cout << vec[0] << " " << vec[1] << " " << vec[2] << " ...\n\n";</pre>
     //~ что лежит в инициализированном массиве
     cout <<"2) Initializing array Days: \n";</pre>
     cout << Days[0] << " " << Days[1] << " " << Days[2] << " ...\n\n";</pre>
     int LeftHand[5] = \{1,2,3,4,5\}; //так объявлять можно, а присваивать нельзя !!!
     int RigthHand[5];
```

```
//~ цикл печати содержимого массива
cout << "3) All elements of Days " << endl;</pre>
for (int i = 0; i < 12; i++)
       cout <<"\tMonth "<< i+1 <<" has " <<Days[i]<<" days.\n";</pre>
//~ если массив инициализирован ЧАСТИЧНО, не до конца, то компилятору все равно
int DaysP[12] = \{31,28,31,30,31\}; //только 5 элементов
cout << "4) Partically Initializing array DaysP: "<< endl;</pre>
//~ цикл печати содержимого массива
for (int i = 0; i < 12; i++)
       cout <<" Month "<< i+1 <<" has " <<DaysP[i]<<" days.\n";</pre>
//~ если список инициализации больше размера массива, то компилятор ругается
//\sim int DaysF[12] = \{31,28,31,30,31,30,31,30,31,30,31,44,55\};
//~ описание и инициализация массива без ЯВНОГО указания его длины (по факту)
int DaysU[] = \{31,28,31,30,31,30,31,31,30\}; //только 9 элементов
//~ что лежит в инициализированном массиве неизвестной длины
cout <<"5) UnKnown size of array DaysU\n";</pre>
for (unsigned int i = 0; i < sizeof DaysU/sizeof(int); i++)</pre>
       cout <<" Month "<< i+1 <<" has " <<DaysU[i]<<" days.\n";</pre>
//~ выход за границы массива НЕ КОНТРОЛИРУЕТСЯ !!!
cout << "6) Days OUT of RANGE !!\n";</pre>
for (int i = 0; i < 22; i++)
       cout <<" Month "<< i+1 <<" has " <<Days[i]<<" days.\n";</pre>
notunn a.
```

Example: birthday.cpp

Определить, сколько дней осталось до нашего дня рождения. На вход программа принимает текущую дату (день и месяц) и дату нашего дня рождения, а на выходе выдает кол-во дней, которое стоит подождать до Дня Рождения. Чтобы упростить задачу, будем считать, что все годы у нас не високосные.

Link with pointers

```
int main(){
        int ages[50]; //массив на 50 элементов
        convert (ages);
}
int convert (int years[]){
        //каков размер массива?
int years[]; и int *years // синонимы
Как узнать размер массива?
```

Link with pointers

Example: sqrt_func.cpp

Пример функций, которые принимают на вход массив.

Multidimensional array

Определение.

Многомерный массив размерности N можно представить как одномерный массив из массивов размерности N-1.

Многомерные массивы обыно не задаются константами:

Dynamic memory

Зачем нужно динамическое выделение памяти?

```
ИмяТипа *ИмяУказателя = new ИмяТипа;

Hanpumep:
int *pn = new int; int *pn; pn=new int; delete pn;
```

Каждому оператору new должен соответствовать свой оператор delete. Иначе могут возникнуть "утечки памяти".

Dynamic memory

Example:

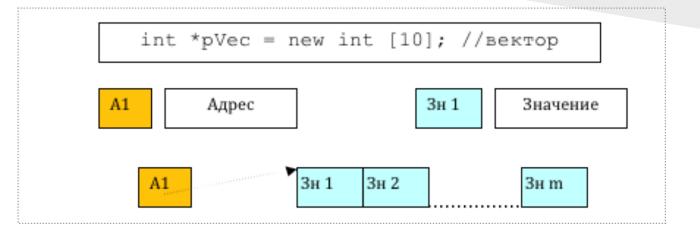
new_delete.cpp

Пример на динамичсекое выделение памяти и ее удаление.

Dynamic array

```
ИмяТипа *ИмяУказателя = new ИмяТипа
[ЧислоЭлементов];
Пример:
int *pnArr = new int [10];//блок на 10
значений типа int
delete [ ] pnArr;
```

Dynamic array



Dynamic array

```
int **ppMatr; //двойной указатель на матрицу n*m
ppMatr
int **ppMatr; //двойной указатель на матрицу n*m
ppMatr = new int * [n];//массив указателей
ppMatr_
int **ppMatr; //двойной указатель на матрицу
ppMatr = new int * [n];//массив указателей
for (i=0; i<n: ++i)
   ppMatr[i] = new int [m]; //массив чисел
                        Зн 2
                                        Зн т
                           3н 1
                                 Зн 2
                                                Зн т
                      3н 1
                            Зн 2
                                           Зн т
```

Sort

- 1. Bubble sort пузырьковая
- 2. Shaker sort шейкерная
- 3. Changed sort обменная
- 4. Quick sort быстрая
- 5. Shell sort сортировка Шелла
- 6. Heap sort пирамидальная
- 7. Radix sort поразрядная

• • •

Introspection sort in C++

Quick Sort

Алгоритм быстрой сортировки:

Краткое описание алгоритма

- 1. выбрать элемент, называемый опорным.
- 2. сравнить все остальные элементы с опорным, на основании сравнения разбить множество на три «меньшие опорного», «равные» и «большие», расположить их в порядке меньшиеравные-большие.
- 3. повторить рекурсивно для «меньших» и «больших».

Быстрая сортировка использует стратегию «разделяй и властвуй».

Quick Sort

Example: quick_sort.cpp

Строка - массив символов.

- 2 способа:
- 1. С-строка
- 2. Класс String из STL-библиотеки

```
Пример:
char dog[6] = {'c','h','a','p','p','i'}; //HE

строка
char cat[5] = {'f','i','s','h','\0'};
//строка
```

Example: init_cstring.cpp

Работа со строками

```
Example strip.cpp
Написать функцию, которая удаляет начальные и конечные пробелы из введенной с клавиатуры строки. (в Python эта функция уже готова и называется strip())
```

Какие проблемы возникают с С-строками?

- 1. массив
- 2. работа с динамичсекой памятью
- 3. сложность операций
- 4. специальные функции из <string.h>,
- <ctype.h>

Таблица 9.1. Некоторые функции из библиотеки обработки символов стуре.h

Прототип функции	Описание функции	
int isdigit(int c)	Возвращает значение true , если c является цифрой, и false в других случаях	
int isalpha(int c)	Возвращает значение true , если c является буквой, и false в других случаях	
int isalnum(int c)	Возвращает значение true , если c является цифрой или буквой, и false в других случаях	
int islower(int c)	Возвращает значение true , если c является буквой нижнего регистра, и false в других случаях	
int isupper(int c)	Возвращает значение true , если c является буквой верхнего регистра, и false в других случаях	
int tolower(int c)	Если с является буквой верхнего регистра, то результат — буква нижнего регистра, в других случаях возвращается аргумент без изменений	
int toupper(int c)	Если с является буквой нижнего регистра, то результат — буква вехнего регистра, в других случаях возвращается аргумент без изменений	

Таблица 9.2. Некоторые функции преобразования строк из библиотеки stdlib.h

Прототип функции	Описание функции
double atof(const char *s)	Преобразует строку s в тип double
int atoi(const char *s)	Преобразует строку s в тип int
long atol(const char *s)	Преобразует строку s в тип long int

Таблица 9.3. Некоторые функции из библиотеки string.h

Прототип функции	Описание функции
size_t strlen(const char *s)	Вычисляет длину строки s в байтах
char *streat(char *s1, const char *s2)	Присоединяет строку s1 в конец строки s2
char *strcpy(char *s1, const char *s2)	Копирует строку s1 в место памяти, на которое указывает s2
char *strncat(char *s1, const char *s2, size_t maxlen)	Присоединяет строку maxlen символов строки s2 в конец строки s1
char *strncpy(char *s1, const char *s2, size_t maxlen)	Копирует maxlen символов строки s2 в место памяти, на которое указывает s1
char * strstr(char *s1, char *s2)	отыскивает позицию первого вхождения строки s2 в строку s1
int stremp(const char *s1, const char *s2)	сравнивает две строки в лексикографическом порядке с учетом различия прописных и

Hometask

- 1) Написать программу, которая сортирует введенный массив символов, с помощью пузырьковой сортировки.
- 2) В введенной матрице найти из каждого столбца количество максимальных элементов и само значение максимального элемента, а из каждой строки -- минимальные.
- 3) Задан числовой массив A[1:m, 1:n]. Составить программу построения одномерного массива, содержащего сначала все отрицательные, затем положительные элементы входного массива.
- 4) Задан числовой массив A[1:m, 1:n]. Составить программу поиска нахождения минимального и максимального элементов массива расположенных на главной диагонали с дальнейшей переменой местами соответствующих столбцов.
- 5) Написать программу, которая проверяет, образуют ли элементы введенного с клавиатуры массива неубывающую последовательность.
- 6) Сформировать матрицу A[1:n, 1:n], элементы которой определяются формулой: cos(i+j)*x. Из матрицы A получить матрицу B, переставляя в каждой строке наибольший по абсолютной величине (первый, если их несколько) с элементом, стоящим на главной диагонали.
- 7) Дана матрица A[1:n, 1:m]. Сформировать матрицу B[1:n, 1:m],, где bij = (aij aij) * aij . Найти *max суммы bij* по *j*-строкам. В матрице *B=(bij) mxn* поменять первую строку со строкой, содержащей максимальное значение суммы ее элементов
- 8*) Написать программу, которая сортирует введенный массив символов, с помощью сортировки вставками.
- 9*) Написать программу, которая сортирует введенный массив символов, с помощью сортировки Шелла.

Hometask

0)Программа, которая в данной последовательности строк(str1) ищет введенную нами строку(str2). Если str2 найдено, то вывести строку(str1), аналогичную первоначальной, но вместо найденной str2 отобразить звездочки -- *.

Если слово не найдено, написать "Not found".

- 1) Написать 0ую программу, без использования string. А с использованием С-строк и С-функций. (и без использования пространства имен using namespace std). +
- 2) Написать программу, которая считывает матрицу, созданную динамическим образом и выводит на экран матрицу, каждый элемент которой возводится в квадрат (матрица вещественных чисел).
- 3) Написать программу, используя класс <string>. Программа должна выяснить, какие слова в строке являются палиндромными и вывести их как результат.
- 4) Написать программу, которая проверяет, является ли введенная с клавиатуры строка двоичным числом.
- 5) Написать программу, которая преобразует введенное с клавиатуры двухразрядное шестнадцатеричное число в десятичное.
- 6) Написать программу, которая проверяет, является ли введенная с клавиатуры строка отрицательным дробным числом.
- 7) Написать функцию, которая возвращает значение True, если символ, полученный функцией в качестве аргумента, не является буквой английского алфавита.