**Класс string C++.** Стандарт языка **С++** включает в себя класс **string**. Этот класс во многом улучшает традиционный *строковый тип.* Для работы со строками предпочтительнее использовать объекты класса **string,**  которые могут работать с перегруженными операциями и функциями класса. Для этого необходимо в программе подключить заголовочный файл **string**:

#include <string>

При работе с объектами класса **string** необходимо знать, что объекты **не заканчиваются** нулевым символом, как это происходит ***в строковом типе***. Поэтому при просмотре строки символ, обозначающий конец строки, не будет найден.

string s1 = "hello";

cout << s1.size() <<endl; // **5**

string s2("\*\*\*\*\*"); // \*\*\*\*\*

Подобно массиву, используя индекс, можно обращаться к отдельным символам объектов класса string, получать и изменять их:

char c = hello[1]; // e

// **Пример 1.** Заменить в натуральном числе **n** с нечётным количеством

// цифр все цифры **3** на цифру **5**.

using namespace std;

#include <iostream>

#include <string>

class **Number**{

long num;

**public:**

Number(long n) { num = n; }

void replace();

void show();

};

void Number::replace(){

strings = **to\_string**(num);

for (**auto**& i : s){

if (i == '3') i +=2;

}

num = **stoi**(s);

}

void Number::show(){

cout << "New number: " << num<<endl;

}

int **main()**{

int n;

cout << "Enter a number "; cin >> n;

Number ob(n);

ob.replace();

ob.show();

}

**Результат:**

Enter a number 123213

New number: 125215

В **Примере 1**. следует обратить внимание на использовании **ссылки** у ключевого слова **auto**. Если её убрать, то результат не получится.

**Функции begin() и end().** Можно вручную определять указатели на начало и конец массива. А можно использовать встроенные библиотечные функции **begin()** и **end()**:

int nums[] = { 1, 2, 3, 4, 5 };

int \*begin = begin(nums); // указатель на начало массива

int \*end = end(nums); // указатель на конец массива

Причем **end** возвращает указатель не на последний элемент, а адрес за последним элементом в массиве.

// **Пример 2.** Функции **begin()**, **end()**

#include <iostream>

using namespace std;

void print(int\*, int\*);

void twice(int\*, int\*);

int **main()**{

int nums1[] = { 1, 2, 3, 4, 5 };

int \*beg = begin(nums1);

int \*en = end(nums1);

**print**(beg, en);

cout<<endl;

int nums2[] = { 1, 2, 3, 4, 5 };

beg = begin(nums2);

en = end(nums2);

**twice**(beg, en);

**print**(beg, en);

}

void **print**(int\* start, int\* last){

for (**auto** \*ptr = start; ptr != last; ++ptr)

cout << \*ptr << ' ';

}

void **twice**(int\* start, int\* last){

for (**auto**\* ptr = start; ptr != last; ptr++)

\*ptr = \*ptr \* 2;

}

**Результат:**

1 2 3 4 5

2 4 6 8 10

Поскольку при передаче массива передаётся фактически указатель на первый элемент, то используя этот указатель, мы можем изменить элементы массива. Если нет необходимости в изменении массива, то лучше параметр-массив определять как константный:

void print(**const** int \*begin, **const** int \*end){

for (const **auto** \*ptr = begin; ptr != end; ptr++)

cout << \*ptr << "\t";

}

В данном случае функция **print()** просто выводит значения из массива, поэтому параметры этой функции помечаются ***как константные***.