

Строки

Библиотека string

- ▶ Библиотечные функции работы со строками объявлены в заголовочном файле **string.h**, который автоматически подключается к исполняемому файлу

Инициализация строк

- ▶ Такая форма определения строки задает пустую строку:

string s;

- ▶ Начальное значение строки можно задать прямо при объявлении:

string s = "Привет!";

- ▶ Новое значение строки записывается с помощью оператора присваивания:

s = "Привет!";

- ▶ Третья форма инициализирует объект типа string другим объектом того же типа:

string st3(st);

Строка st3 инициализируется строкой st

ВВОД И ВЫВОД СТРОК

- ▶ Для того, чтобы ввести из входного потока строку до первого пробела используется метод **cin**:

```
cin >> s;
```

- ▶ Если нужно ввести строку, содержащую пробелы, применяется функция **getline**:

```
getline ( cin, s );
```

- ▶ Вывод выполняется стандартным образом:

```
cout << s;
```

Функции библиотеки

- ▶ В библиотеку **string** входят следующие функции:
 - ▶ копирования строк
 - ▶ объединения строк
 - ▶ сравнения строк
 - ▶ определения длины строки
 - ▶ поиска символов в строке
 - ▶ поиска подстроки в строке

Копирование строк

- ▶ Функция **strcpy (dest, src)**

Копирует содержимое строки **src** в строку **dest**, заменяя старое содержимое этой строки

- ▶ Функция **strncpy (dest, src, n)**

Заменяет первые **n** символов строки **dest** первыми **n** символами строки **src**

Обе функции возвращают обновленную строку **dest**

Объединение строк

- ▶ Функция **strcat (dest, src)**

Добавляет содержимое строки **src** к строке **dest**, заменяя старое содержимое этой строки

- ▶ Функция **strncat (dest, src, n)**

Добавляет первые **n** символов строки **src** к строке **dest**

Обе функции возвращают обновленную строку **dest**

Сравнение строк

- ▶ Функция **strcmp (str1, str2)**

Сравнивает содержимое строки **str1** с содержимым строки **str2**

- ▶ Функция **strncmp (str1, str2, n)**

Сравнивает первые **n** символов строк **str1** и **str2**

Правило сравнения строк

- ▶ Строки сравниваются посимвольно до тех пор, пока не будут обнаружены несовпадающие символы или будет достигнут конец одной из строк
- ▶ Если достигнут конец одной из строк и одновременно достигнут конец другой строки, то строки считаются равными, иначе закончившаяся строка считается меньшей
- ▶ Если обнаружены несовпадающие символы, то меньшей считается строка, которой принадлежит меньший из символов

Определение длины строки

- ▶ Функция **strlen (str1)**

Возвращает длину строки (без нулевого символа конца строки)

При отсутствии нуль-символа в конце строки выдается сообщение об ошибке

Поиск подстроки

- ▶ Функция **strstr (str, sub)**
 - ▶ Ищет первое вхождение подстроки sub в строку str
 - ▶ Если подстрока обнаружена, то функция возвращает указатель на то место в строке str, с которого начинается подстрока
 - ▶ Если подстрока не обнаружена, то возвращается значение NULL

Указатель на подстроку

- ▶ Значение, возвращаемое функцией **strstr** может быть сохранено в отдельной переменной (указателе)
- ▶ Такая переменная объявляется следующим образом:

char* <имя переменной>;

- ▶ Например:

char* w; w = strstr (s, substr);

- ▶ В дальнейшем с этой переменной можно работать как с обычной строкой

ПОИСК СИМВОЛОВ В СТРОКЕ

- ▶ Группа функций:
 - ▶ **strchr (str, ch)** – ищет первое вхождение символа ch в строке str;
 - ▶ **strrchr(str, ch)** - поиск символа ch, начиная с конца строки str;
 - ▶ **strcspn (sr1, str2)** – возвращает индекс первого символа в строке str1, который не принадлежит множеству символов str2, т.е. длину начальной подстроки в строке str1, которая полностью состоит из символов строки str2;



Для определения длины строки `s` используется запись **`s.size()`**

- ▶ Такая запись означает, что метод **`size`** применяется к объекту `s` типа **`string`**.
- ▶ В данном случае **`size`** – это функция (метод), связанная с типом данных **`string`**.

Конкатенация строк

- ▶ Для конкатенации строк используется операция сложения (+) или операция сложения с присваиванием (+=).
- ▶ Пусть даны две строки:

```
string s1 ( "hello, " );  
string s2( "world\n" );
```

- ▶ Мы можем получить третью строку, состоящую из конкатенации первых двух, таким образом:

```
string s3 = s1 + s2;
```

- ▶ Если же мы хотим добавить s2 в конец s1, мы должны написать:

```
s1 += s2;
```

МЕТОДЫ КЛАССА STRING

- ▶ Метод **substr** – выделение части строки

```
s = "0123456789";  
s1 = s.substr ( 3, 5 );  
cout << s1 << endl;
```

- ▶ Фрагмент копирует в строку s1 пять символов строки s (с 3-го по 7-й).
- ▶ Этот метод принимает два параметра: номер начального символа и количество символов.
- ▶ Если второй параметр при вызове substr не указан, метод возвращает все символы до конца строки. Например,

```
s = "0123456789";  s1 = s.substr ( 3 );  
//вернёт «3456789».
```

МЕТОДЫ КЛАССА STRING

- ▶ Метод **erase** – удаление части строки

```
s = "0123456789";
```

```
s.erase ( 3, 6 );
```

- ▶ В строке `s` остаётся значение «0129» (удаляются 6 символов, начиная с 3-го).
- ▶ Обратите внимание, что процедура `erase` изменяет строку.

МЕТОДЫ КЛАССА STRING

- ▶ Метод **insert** - вставка символов в строку

```
s = "0123456789";  
s.insert ( 3, "ABC" );
```

Переменная s получит значение «012ABC3456789».

МЕТОДЫ КЛАССА STRING

- ▶ Метод **find** - поиск в строке
- ▶ Эта функция возвращает номер найденного символа (номер первого символа подстроки) или -1 , если найти нужный фрагмент не удалось.

- ▶ Пример:

```
string s = "Здесь был Вася.";
int n;
n = s.find ( 'c' );
if ( n >= 0 )
    cout << "Номер первого символа 'c': " << n << endl;
else cout << "Символ не найден " << endl;
```

Преобразование строки в число

- ▶ В C++ нет методов преобразования строки в число. Но можно воспользоваться функциями языка C. В этом случае сначала необходимо преобразовать тип `string` в обычную (анси) строку, а потом использовать функции **`atoi`** и **`atof`**, прототип которых находится в **`stdlib.h`**:

```
string s = "123.456";
```

```
int N;
```

```
double X;
```

```
N = atoi ( s.c_str() ); // N=123
```

```
X = atof ( s.c_str() ); // X = 123.456
```

- ▶ Метод `c_str()` преобразует строку `string` в строку с завершающим нулем.
- ▶ Преобразование происходит до первого символа, не относящегося к числу.