



Строки. Введение

Урок №28

Разбор домашней работы





ЧТО НАС ЖДЕТ ?

ПЛАН УРОКА

- Повторение: символы
- Теория: строки и символы в них
- Решаем задачи
- Теория: поиск и удаление символов в строке
- Решаем задачи
- Теория: новый for
- Подводим итоги

Повторение

Что здесь происходит?

```
int main() {  
    int count = 0;  
    char c = cin.get();  
    while (c != '.') {  
        if (c != '\n') {  
            count++;  
        }  
        c = cin.get();  
    }  
    cout << count;  
}
```

Теория



15 минут





Строки

Подключение :

```
#include <string>

// подключение библиотеки для работы со строками
// в большинстве случаев подключать не обязательно
```

Использование :

```
string x = "Meow";

// создание переменной и запись в нее текстового значения

cin >> x;

// в переменную вводится текстовое значение до пробела
// либо до нажатия клавиши Enter
```

Пример

```
string name;  
cout << "Enter your name" << endl;  
  
cin >> name;  
  
cout << "Hello, " << name;
```

Ввод: 'John Andrew'

Вывод: Hello, John



Работа со строками

Для чтения целой строки из консоли, включая пробелы:

```
string str;  
getline(cin, str);
```

Первым аргументом **getline** принимает поток ввода, а вторым - строку, в которую считает результат.



Работа со строками

- Строка - массив символов, доступ к символу:

```
str[i] ;
```

При выходе индекса за границы массива символов программа может работать некорректно.

- Узнать длину строки:

```
str.size() ;
```

```
str.length() ;
```



Работа со строками

- Строка - массив символов, доступ к символу:

```
str[i] ;
```

Индексация строки начинается с **нуля**, аналогично массивам.

- Узнать длину строки:

```
str.size() ;  
str.length() ;
```

} эквивалентны



Работа со строками

- Первый символ в строке:

```
str.front()
```

- Последний символ в строке:

```
str.back();
```

Потренируемся?

Пример

Вопрос

Что будет выведено в консоль, если ввести **meme**?

```
int main() {  
    string str;  
    cin >> str;  
    cout << str[2] << str.back() << str.front() << str[1];  
}
```

Пример

Вопрос

Что будет выведено в консоль, если ввести **meme**?

```
int main() {  
    string str;  
    cin >> str;  
    cout << str[2] << str.back() << str.front() << str[1];  
}
```

Ответ: meme

Пример

Вопрос

Что будет выведено в консоль, если ввести **hello**?

```
int main() {  
    string str;  
    cin >> str;  
    cout << str[str.size() - 1] << str.back();  
}
```


Пример

Вопрос

Что будет выведено в консоль, если ввести **hello**?

```
int main() {  
    string str;  
    cin >> str;  
    cout << str[str.size() - 1] << str.back();  
}
```

Ответ: oo

Практика



20 минут



Теория



18 минут





Удаление символов из строки

`str.erase(pos, len)`

Где:

pos - индекс первого удаляемого элемента

len - количество удаляемых символов

(если len не указан, то удаляет все символы до конца строки)

Пример

```
int main(){  
    string str = "Hello, world!";  
    str.erase(5, 1);  
    cout << str << endl;      Вывод: ???  
    str.erase(0, 2);  
    cout << str << endl;      Вывод: ???  
    str.erase(2);  
    cout << str << endl;      Вывод: ???  
}
```

Пример

```
int main() {  
    string str = "Hello, world!";  
    str.erase(5, 1);  
    cout << str << endl;      Вывод: Hello world!  
    str.erase(0, 2);  
    cout << str << endl;      Вывод: ???  
    str.erase(2);  
    cout << str << endl;      Вывод: ???  
}
```

Пример

```
int main() {  
    string str = "Hello, world!";  
    str.erase(5, 1);  
    cout << str << endl;      Вывод: Hello world!  
    str.erase(0, 2);  
    cout << str << endl;      Вывод: llo world!  
    str.erase(2);  
    cout << str << endl;      Вывод: ???  
}
```

Пример

```
int main() {  
    string str = "Hello, world!";  
    str.erase(5, 1);  
    cout << str << endl;    Вывод: Hello world!  
    str.erase(0, 2);  
    cout << str << endl;    Вывод: llo world!  
    str.erase(2);  
    cout << str << endl;    Вывод: ll  
}
```


**Какие ещё операции можно
делать со строкой?**

Поиск в строке



Поиск в строке

| | |
|--------------------------|---|
| find | Позиция первого вхождения символа или подстроки |
| rfind | Позиция последнего вхождения символа или подстроки |
| find_first_of | Позиция первого вхождения символа или набора символов |
| find_first_not_of | Позиция первого вхождения символа не равного символу или набору символов |
| find_last_of | Позиция последнего вхождения символа или набора символов |
| find_last_not_of | Позиция последнего вхождения символа не равного символу или набору символов |



Поиск в строке

Каждая функция поиска может принимать второй аргумент - позиция начала поиска.

Пример:

`str.find("H", 1);` - ищем **H**, начиная со второго символа

Рассмотрим работу функций поиска на примере

Поиск в строке

Hello, world!

Поиск в строке

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Н | е | л | л | о | , | | в | о | р | л | д | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

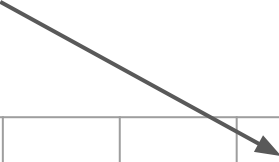
Поиск в строке

```
str.find("o")
```

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

Поиск в строке

```
str.find("o")
```



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

Поиск в строке

```
str.find("o")
```

```
str.rfind("o")
```



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Н | е | l | l | о | , | | w | о | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

Поиск в строке

```
str.find("o")
```

```
str.rfind("o")
```

A diagram illustrating string indexing for the string "Hello, world!". The string is displayed in a red serif font within a table of 13 columns. Below each character is its corresponding index from 0 to 12. Two arrows point from the code examples above to the 'o' characters at indices 4 and 8.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

```
str.find_first_of("wl")
```

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

`str.find_last_of("rl")`

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

`str.find_last_of("rl")`

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

`str.find_last_of("rl")`

`str.find_first_not_of("leH")`

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

`str.find_last_of("rl")`

`str.find_first_not_of("leH")`

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

`str.find_last_of("rl")`

`str.find_first_not_of("leH")`

`str.find_last_not_of("!dl")`

Поиск в строке

`str.find("o")`

`str.rfind("o")`

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| H | e | l | l | o | , | | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

`str.find_first_of("wl")`

`str.find_first_not_of("leH")`

`str.find_last_of("rl")`

`str.find_last_not_of("!dl")`



Поиск в строке

Если функция поиска не нашла символ, она возвращает значение `string::npos`.

Пример:

```
int pos = str.find("H");  
if(pos == string::npos) {  
    cout << "Символ не найден";  
}
```

Пример

```
int main() {  
    string str = "Hello world!";  
    int pos = str.find("H", 1);  
    if (pos == string::npos) {  
        cout << "NO";  
    } else {  
        cout << "YES";  
    }  
}
```

Что будет выведено в консоль?

Пример

```
int main() {  
    string str = "Hello world!";  
    int pos = str.find("H", 1);  
    if (pos == string::npos) {  
        cout << "NO";  
    } else {  
        cout << "YES";  
    }  
}
```

Обратите внимание на
второй аргумент!

Что будет выведено в консоль? **NO**

Пример

```
int main() {  
    string str = "Hello world!";  
    int pos = str.find("H", 1);  
    if (pos == string::npos) {  
        cout << "NO";  
    } else {  
        cout << "YES";  
    }  
}
```

Обратите внимание на
второй аргумент!

Что будет выведено в консоль? **NO**

Практика



15 минут



Теория



3 минут





Перебор элементов строки

Цикл **for** дружит со строками и умеет перебирать их без **i**.

```
for(char ch : str)
```

*Данный тип циклов называется *Range based for loop* и пришёл из стандарта C++11. Благодаря ему можно проходить весь массив от начала до конца, не используя дополнительных условий и счетчиков.*

Что выведет этот код?

```
int main() {  
    string str = "Hello, world!";  
    for(char ch: str){  
        cout << (char)toupper(ch);    -   ???  
    }  
    cout << endl;  
    for(char ch: str){  
        cout << (char)tolower(ch);    -   ???  
    }  
}
```

Что выведет этот код?

```
int main() {  
    string str = "Hello, world!";  
    for(char ch: str){  
        cout << (char)toupper(ch);    - HELLO, WORLD!  
    }  
    cout << endl;  
    for(char ch: str){  
        cout << (char)tolower(ch);    - ???  
    }  
}
```

Что выведет этот код?

```
int main() {  
    string str = "Hello, world!";  
    for(char ch: str){  
        cout << (char)toupper(ch);    - HELLO, WORLD!  
    }  
    cout << endl;  
    for(char ch: str){  
        cout << (char)tolower(ch);    - hello, world!  
    }  
}
```

Что выведет этот код?

```
int main() {  
    string str = "Hello, world!";  
    for(char ch: str){  
        cout << (char)toupper(ch);  
    }  
    cout << endl;  
    for(char ch: str){  
        cout << (char)tolower(ch);  
    }  
}
```

Что будет, если не ставить `(char)` перед функцией?

Итоги урока

- 1) Какой тип используется для строк?
- 2) Как считать целую строку, включая пробелы?
- 3) Как получить символ в строке по индексу?
- 4) Как получить первый и последний символы в строке?
- 5) Как удалить символ из строки?