Лекция 11

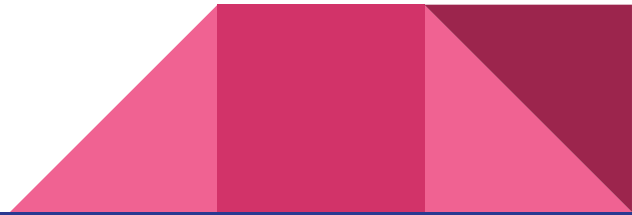
Символы и строки постоянной длины. Класс String

Символы

Символ – элементарная единица, некоторый набор которых несет определенный смысл.

В языке программирования C++ предусмотрено использование символьных констант.

Символьная константа – это целочисленное значение (типа int) представленное в виде символа, заключенного в одинарные кавычки, например 'a'. В таблице ASCII представлены символы и их целочисленные значения.



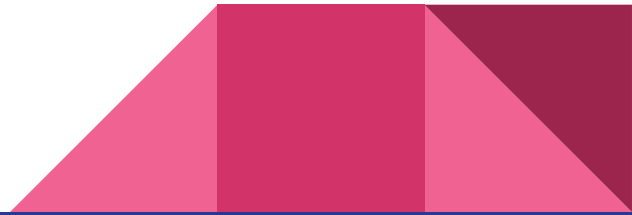
Объявление символьной переменной

```
char symbol = 'a';
//  где symbol – имя переменной типа  char
//  char – тип данных для хранения символов
```

Строки

Строки в C++ представляются как массивы элементов типа char, заканчивающиеся нуль-терминатором \0, называются C-строками или строками в стиле C.

Символьные строки состоят из набора символьных констант заключённых в двойные кавычки. При объявлении строкового массива необходимо учитывать наличие в конце строки нуль-терминатора, и отводить дополнительный байт под него.



Объявление строки

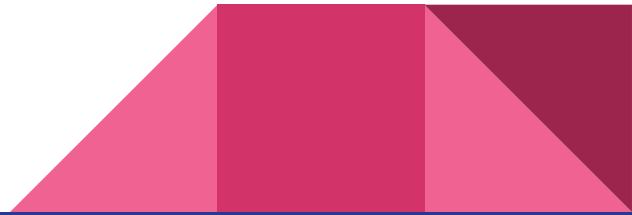
char string[10];

// где string – имя строковой переменной

// 10 – размер массива, то есть в данной строке может поместиться 9 символов , последнее место отводится под нуль-терминатор.

Строка при объявлении может быть инициализирована начальным значением, например, так:

char string[10] = "abcdefghf";

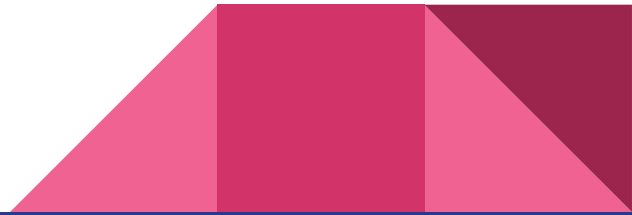


Объявление строки

При объявлении строки не обязательно указывать её размер, но при этом обязательно нужно её инициализировать начальным значением. Тогда размер строки определится автоматически и в конец строки добавится нуль-терминатор.

char string[ ] = "abcdefghf";

Строка может содержать символы, цифры и специальные знаки. В C++ строки заключаются в двойные кавычки. Имя строки является константным указателем на первый символ.



Функции для работы со строками и символами

strlen(имя_строки)	определяет длину указанной строки, без учёта нуль-символа
strcpy(s1,s2)	выполняет побайтное копирование символов из строки s2 в строку s1
strncpy(s1,s2, n)	выполняет побайтное копирование n символов из строки s2 в строку s1. возвращает значения s1
strcat(s1,s2)	объединяет строку s2 со строкой s1. Результат сохраняется в s1
strncat(s1,s2,n)	объединяет n символов строки s2 со строкой s1. Результат сохраняется в s1

strcmp(s1,s2)	сравнивает строку s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2, <0 — если s1>s2 С учётом регистра
strncmp(s1,s2,n)	сравнивает n символов строки s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2, <0 — если s1>s2 С учётом регистра
stricmp(s1,s2)	сравнивает строку s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2, <0 — если s1>s2 Без учёта регистра
strnicmp(s1,s2,n)	сравнивает n символов строки s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2, <0 — если s1>s2 Без учёта регистра

isalnum(c)	возвращает значение true, если с является буквой или цифрой, и false в других случаях
isalpha(c)	возвращает значение true, если с является буквой, и false в других случаях
isdigit(c)	возвращает значение true, если с является цифрой, и false в других случаях
islower(c)	возвращает значение true, если с является буквой нижнего регистра, и false в других случаях
isupper(c)	возвращает значение true, если с является буквой верхнего регистра, и false в других случаях
isspace(c)	возвращает значение true, если с является пробелом, и false в других случаях
toupper(c)	если символ с, является символом нижнего регистра, то функция возвращает преобразованный символ с в верхнем регистре, иначе символ возвращается без изменений.

strchr(s,c)	поиск первого вхождения символа с в строке s. В случае удачного поиска возвращает указатель на место первого вхождения символа с. Если символ не найден, то возвращается ноль.
strcspn(s1,s2)	определяет длину начального сегмента строки s1, содержащего те символы, которые не входят в строку s2
strspn(s1,s2)	возвращает длину начального сегмента строки s1, содержащего только те символы, которые входят в строку s2
strprbk(s1,s2)	возвращает указатель первого вхождения любого символа строки s2 в строке s1

atof(s1)	преобразует строку s1 в тип double
atoi(s1)	преобразует строку s1 в тип int
atol(s1)	преобразует строку s1 в тип long int
getchar(c)	считывает символ с со стандартного потока ввода, возвращает символ в формате int
gets(s)	считывает поток символов со стандартного устройства ввода в строку s до тех пор, пока не будет нажата клавиша ENTER

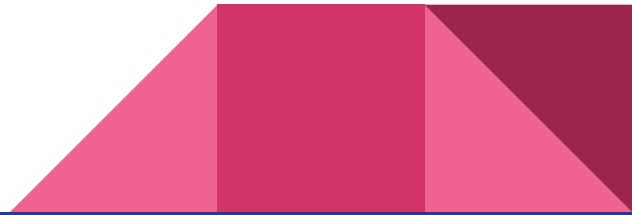
Класс String

В языке C++ для удобной работы со строками есть класс string, для использования которого необходимо подключить заголовочный файл string.

Строки можно объявлять и одновременно присваивать им значения:

string S1, S2 = "Hello";

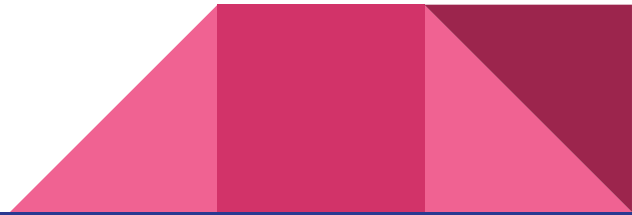
К отдельным символам строки можно обращаться по индексу, как к элементам массива или С-строк. Например S[0] - это первый символ строки.



Конструкторы строк

Строки можно создавать с использованием следующих конструкторов:

- **string()** - конструктор по умолчанию (без параметров) создает пустую строку.
- **string(string & S)** - копия строки S
- **string(size_t n, char c)** - повторение символа с заданное число n раз.
- **string(size_t c)** - строка из одного символа c.
- **string(string & S, size_t start, size_t len)** - строка, содержащая не более, чем len символов данной строки S, начиная с символа номер start.



Арифметические операторы

Со строками можно выполнять следующие арифметические операции:

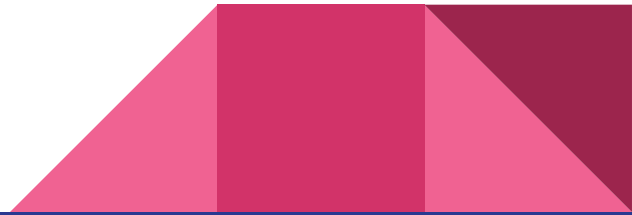
= - присваивание значения.

+= - добавление в конец строки другой строки или символа.

+ - конкатенация двух строк, конкатенация строки и символа.

==, != - посимвольное сравнение.

<, >, <=, >= - лексикографическое сравнение.



Методы строк

1. **size**

Метод **size()** возвращает длину длину строки. Возвращаемое значение является беззнаковым типом. Поэтому нужно аккуратно выполнять операцию вычитания из значения, которое возвращает **size()**.

1. **resize**

S.resize(n) - изменяет длину строки, новая длина строки становится равна n. При этом строка может как уменьшится, так и увеличиться. Если вызвать в виде **S.resize(n, c)**, где c - символ, то при увеличении длины строки добавляемые символы будут равны c.

Методы строк

3. clear

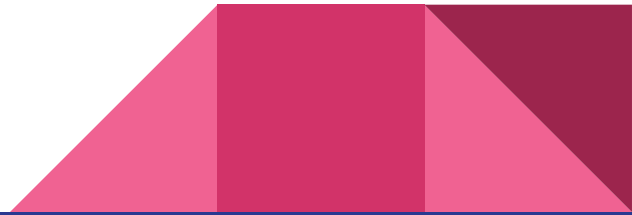
S.clear() - очищает строку, строка становится пустой.

4. empty

S.empty() - возвращает true, если строка пуста, false - если непуста.

5. push_back

S.push_back(c) - добавляет в конец строки символ c, вызывается с одним параметром типа char.



Методы строк

6. append

Добавляет в конец строки несколько символов, другую строку или фрагмент другой строки. Имеет много способов вызова.

S.append(n, c) - добавляет в конец строки n одинаковых символов, равных c. n имеет целочисленный тип, c - char.

S.append(T) - добавляет в конец строки S содержимое строки T. T может быть объектом класса string или C-строкой.

S.append(T, pos, count) - добавляет в конец строки S символы строки T начиная с символа с индексом pos количеством count.

Методы строк

7. erase

S.erase(pos) - удаляет из строки S с символа с индексом pos и до конца строки.

S.erase(pos, count) - удаляет из строки S с символа с индексом pos количеством count или до конца строки, если pos + count > S.size().

8. substr

S.substr(pos) - возвращает подстроку данной строки начиная с символа с индексом pos и до конца строки.

S.substr(pos, count) - возвращает подстроку данной строки начиная с символа с индексом pos количеством count или до конца строки, если pos + count > S.size().

Методы строк

9. insert

Вставляет в середину строки несколько символов, другую строку или фрагмент другой строки.

S.insert(i, n, c) - вставить n одинаковых символов, равных c. n имеет целочисленный тип, c - char.

S.insert(i, T) - вставить содержимое строки T. T может быть объектом класса string или C-строкой.

S.insert(i, T, pos, count) - вставить символы строки T начиная с символа с индексом pos количеством count.

Методы строк

10. replace

Заменяет фрагмент строки на несколько равных символов, другую строку или фрагмент другой строки.

S.replace(pos, count, n, c) - вставить n одинаковых символов, равных c. n имеет целочисленный тип, c - char.

S.replace(pos, count, T) - вставить содержимое строки T. T может быть объектом класса string или C-строкой.

S.replace(pos, count, T, pos2, count2) - вставить символы строки T начиная с символа с индексом pos количеством count.

Методы строк

11. find

Ищет в данной строке первое вхождение другой строки str. Возвращается номер первого символа, начиная с которого далее идет подстрока, равная строке str. Если эта строка не найдена, то возвращается константа string::npos .

Если задано значение pos, то поиск начинается с позиции pos, то есть возращаемое значение будет не меньше, чем pos. Если значение pos не указано, то считается, что оно равно 0 - поиск осуществляется с начала строки.

S.find(str, pos = 0) - искать первое входение строки str начиная с позиции pos. Если pos не задано - то начиная с начала строки S.

S.find(str, pos, n) - искать в данной строке подстроку, равную первым n символам строки str. Значение pos должно быть задано.