Программирование на языке C++

§ 66. Символьные строки

В C++ существует два типа строк:

```
1. char s[10];  // массив символов
```

Переменная хранит в себе только 1 символ, элементы массива – отдельные объекты, сложно работать со строками переменной длины

```
2. string s;  // символьная строка
```

строка

Это специальный класс **string**
Для его подключения в начале программы нужно подключить :
#include <string>

Символьные строки

Начальное значение:

```
string s = "Привет!";
```

Присваивание:

```
s = "Привет!";
```

Вывод на экран:

```
cout << s;
```

А если массив?

Символьные строки

Ввод с клавиатуры:

```
cin >> s;
```

только до пробела!

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

до перевода строки (Enter)

Отдельный символ:

```
s[4] = 'a';
```

Символы в строке нумеруются с нуля!

Длина строки:

```
int n;
...
n = s.size();
```

метод для объектов типа `string`

Символьные строки

Задача: заменить в строке все буквы **'а'** на буквы **'б'**.

```
#include <iostream>
using namespace std;
main()
{
  setlocale(0,"rus");
   string s;
   int i;
   cout << "Введите строку: ";
   getline ( cin, s );
   for ( i=0; i<s.size(); i++)
     if ( s[i] == 'a' )
       s[i] = 'b';
   cout << s;
}
```

цикл по всем символам строки

Операции со строками

s.append(str)	добавляет в конец строки строку str. Можно писать как s.append(переменная), так и s.append("строка")
s.assign(str)	присваивает строке s значение строки str. Аналогично записи s=str
int i=s.begin()	записывает в i индекс первого элемента строки
int i=s.end()	аналогично, но последнего
s.clear()	как следует из названия, очищает строку. Т.е. удаляет все элементы в ней
s.compare(str)	сравнивает строку s со строкой str и возвращает 0 в случае совпадения (на самом деле сравнивает коды символов и возвращает их разность)
s.copy **(куда, сколько, начиная с какого)**	- копирует из строки s в куда (там может быть как строка типа стринг, так и строка типа char). Последние 2 параметра не обязательные (можно использовать функцию с 1,2 или 3 параметрами)

Операции со строками

bool b=s.empty()	если строка пуста, возвращает true, иначе false
s.erase(откуда, сколько)	удаляет n элементов с заданной позиции
s.find(str,позиция)	ищет строку str начиная с заданной позиции
s.insert(позиция,str, начиная, beg, count)	вставляет в строку s начиная с заданной позиции часть строки str начиная с позиции beg и вставляя count символов
int len=s.length()	записывает в len длину строки
s.push_back(symbol)	добавляет в конец строки символ
s.replace(index, n,str)	берет n первых символов из str и заменяет символы строки s на них, начиная с позиции index
str=s.substr(n,m)	возвращает m символов начиная с позиции n
s.swap(str)	меняет содержимое s и str местами
s.size()	возвращает число элементов в строке

Операции со строками

Объединение (*конкатенация*):

```
string s, s1, s2;
s1 = "Привет";
s2 = "Вася";
s = s1 + ", " + s2 + "!";
```

"Привет, Вася!"

Срез (подстрока):

```
s = "0123456789";
s1 = s.substr( 3, 5 );    // «23456"
```

откуда

с какого символа

сколько символов

5

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

Операции со строками

Удаление:

```
s = "0123456789";
s.erase ( 3, 6 ); // "0129"
```

с какого символа

сколько символов

Вставка:

```
s = "0123456789";
s.insert( 3,"ABC" ); // "012ABC3456789"
```

куда

с какого символа

что

Поиск символа в строке

```
string s = "Здесь был Вася.";
int n;
n=s.find( 'c' );  // 3
```

find – искать

Вернёт **-1**, если не нашли!

```
if ( n>=0 )
      cout << "Номер символа 'c': "
           << n << endl;
else cout << "Символ не найден.\n";
```

Поиск подстроки

```
string s = "Здесь был Вася.";
int n;
n=s.find( "Вася" );  // 10
```

```
if ( n>=0 )
  cout << "Слово начинается с s["
       << n << "]\n";
else
  cout << "Слово не найдено.\n";
```

s.rfind() – поиск с конца строки!

Пример обработки строк

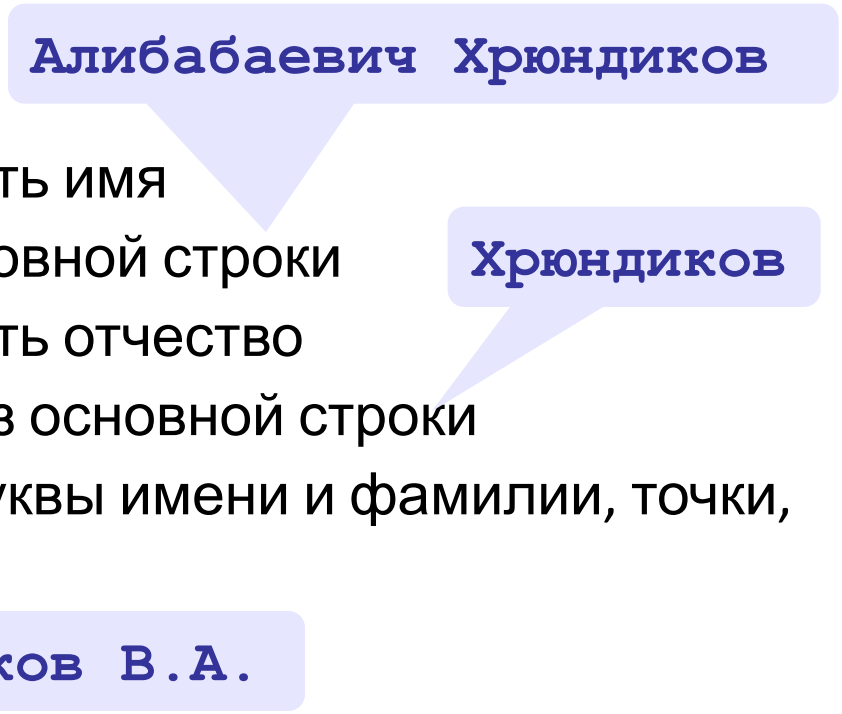
Задача: Ввести имя, отчество и фамилию. Преобразовать их к формату «фамилия-инициалы».

Пример:

```
Введите имя, отчество и фамилию:
Василий Алибабаевич Хрюндиков
Результат:
Хрюндиков В.А.
```

Алгоритм:

- найти первый пробел и выделить имя
- удалить имя с пробелом из основной строки
- найти первый пробел и выделить отчество
- удалить отчество с пробелом из основной строки
- «сцепить» фамилию, первые буквы имени и фамилии, точки, пробелы…



Пример обработки строк

```
main()
{
  string s, name, name2;
  int n;
  cout << "Введите имя, отчество и фамилию: ";
  getline ( cin, s );
  name = s.substr(0,1) + '.';// начало имени
  n = s.find(' ');           // найти пробел
  s = s.substr ( n+1 );      // удалить имя
  n = s.find(' ');           // найти пробел
  name2 = s.substr(0,1) + '.';// начало отчества
  s = s.substr ( n+1 );      // осталась фамилия
  s = s + ' ' + name + name2;// результат
  cout << s;
}
```

Задачи

«А»: Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы.

Пример:

```
Введите фамилию, имя и отчество:
Иванов Петр Семёнович
П.С. Иванов
```

Задачи

«B»: Ввести адрес файла и «разобрать» его на части, разделенные знаком `'/'`. Каждую часть вывести в отдельной строке.

Пример:

```
Введите адрес файла:
C:/Фото/2013/Поход/vasya.jpg
C:
Фото
2013
Поход
vasya.jpg
```

Задачи

«С»: Напишите программу, которая заменяет во всей строке одну последовательность символов на другую.

Пример:

```
Введите строку:
(X > 0) and (Y < X) and (Z > Y) and (Z <> 5)
Что меняем: and
Чем заменить: &
Результат
(X > 0) & (Y < X) & (Z > Y) & (Z <> 5)
```

Преобразования «строка» – «число»

Из строки в число:

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

«12x3» → 12

в строку языка Си

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

Преобразования «строка» – «число»

Из числа в строку:

Идея: направить выходной поток в строку!

```
#include <sstream>
```

строковые потоки

```
ostringstream ss;
string s;
int N=123;
ss << N;
s=ss.str();    // s = "123"
```

строковый поток вывода

из потока в строку

Преобразования «строка» – «число»

Вещественное число в строку:

```
ostringstream ss;
string s;
double X=123.456;
ss.width(10);     // ширина поля
ss.precision(3); // знаков в дробной части
ss << X;
s=ss.str();      // s ="   123.456"
```

Научный формат:

```
ss.str("");       // очистка потока
ss.width(10);     // ширина поля
ss.precision(6); // знаков в дробной части
ss << scientific << X; // научный формат
s=ss.str();      // s = "1.234560E+002"
```

Задачи

«A»: Напишите программу, которая вычисляет сумму трех чисел, введенную в форме символьной строки. Все числа целые.

Пример:

```
Введите выражение:
12+3+45
Ответ: 60
```

«B»: Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются только знаки «+» или «–»). Выражение вводится как символьная строка, все числа целые.

Пример:

```
Введите выражение:
12-3+45
Ответ: 54
```

Задачи

«C»: Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются знаки «+», «–», «*» и «/»). Выражение вводится как символьная строка, все числа целые. Операция «/» выполняется как целочисленное деление (**div**).

Пример:

```
Введите выражение:
12*3+45
Ответ: 81
```

Задачи

«D»: Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются знаки «+», «–», «*» и «/») **и круглых скобок**. Выражение вводится как символьная строка, все числа целые. Операция «/» выполняется как целочисленное деление.

Пример:

```
Введите выражение:
2*(3+45)+4
Ответ: 100
```