

# Программирование в C++

Преподаватель  
Колотова Людмила Павловна

# Содержание 8-го занятия по практике

---

## Строки.

- Строковые переменные. Считывание строк.  
Копирование строк
- Массивы строк. Стандартный класс `string` языка C++
- Ввод/вывод для объекта класса `string`
- Поиск, модификация, сравнение объектов класса `string`

# Строки

---

В **C++** используются два вида строк:

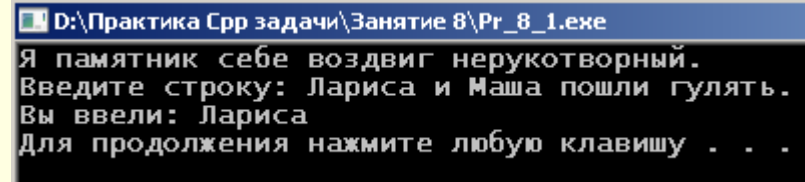
- ❑ строка как массив символов типа **char**  
(или строковый тип)
- ❑ строка как объект класса **string**

## Задача 8-1

# Строковые константы и переменные

В данной задаче одну строку инициализируем постоянным значением при ее определении, а другую строку определим как строковую переменную. Программа просит пользователя ввести строку, помещает эту строку в строковую переменную, а затем выводит эту строку на дисплей.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251);           // установить на консоле
    SetConsoleOutputCP(1251);     // шрифт Lucida Console
    char st0[] = "Я памятник себе воздвиг нерукотворный.";
    cout << st0 << endl;
    const int MAX = 80;
    char str[MAX];
    cout << "Введите строку: ";
    cin >> str;
    cout << "Вы ввели: " << str << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



```
D:\Практика Сpp задачи\Занятие 8\Pr_8_1.exe
Я памятник себе воздвиг нерукотворный.
Введите строку: Лариса и Маша пошли гулять.
Вы ввели: Лариса
Для продолжения нажмите любую клавишу . . .
```

Строковая константа полностью напечатана, а в строковой переменной только одно слово (операция >> считает пробел нулевым символом).

# Строки

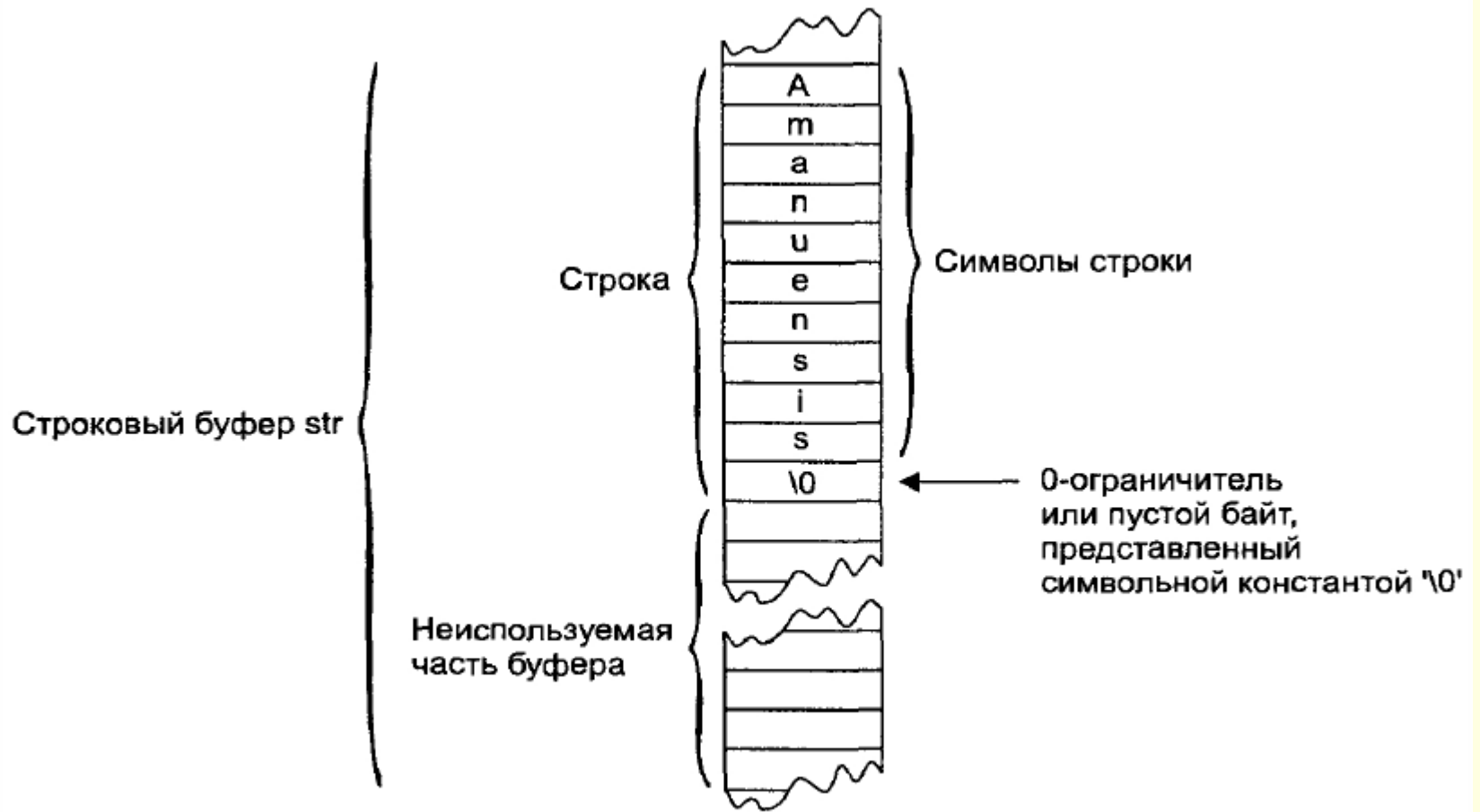


Рис. Строка, хранящаяся в строковой переменной

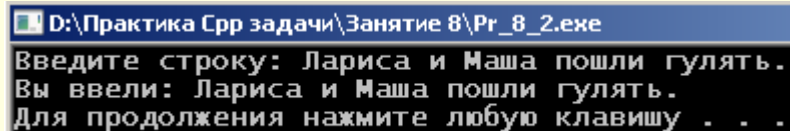
## Задача 8-2

# Считывание строки с пробелами

Для считывания строк, содержащих пробелы, используется другой метод — **cin.get()**. Этот синтаксис означает использовать метод `get()` класса `stream` для его объекта `cin`.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int MAX = 80;
    char str[MAX];
    cout << "Введите строку: ";
    cin.get(str, MAX);
    cout << "Вы ввели: " << str << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Первый аргумент метода `cin.get()` — это адрес массива, куда будет помещена введенная строка. Вторым аргументом определяет максимальный размер массива, автоматически предупреждая, таким образом, его переполнение.



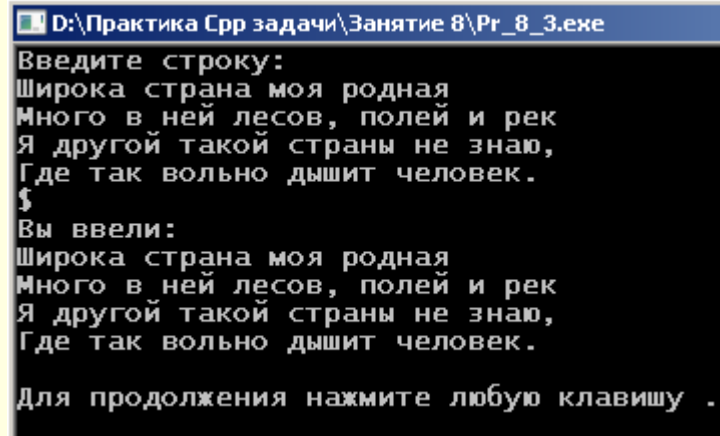
```
D:\Практика Сpp задачи\Занятие 8\Pr_8_2.exe
Введите строку: Лариса и Маша пошли гулять.
Вы ввели: Лариса и Маша пошли гулять.
Для продолжения нажмите любую клавишу . . .
```

## Задача 8-3

# Считывание нескольких строк

Для считывания нескольких строк метод **cin.get()** применяется, но уже с тремя аргументами. Третий аргумент определяет символ, на котором метод завершает считывание строки. По умолчанию значением этого аргумента является символ новой строки ('\n'). В нашей задаче вызовем метод **cin.get()** с символом доллара ('\$') в качестве третьего аргумента.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int MAX = 2000;
    char str[MAX];
    cout << "Введите строку: \n";
    cin.get(str, MAX, '$');
    cout << "Вы ввели: \n" << str << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



```
D:\Практика Сpp задачи\Занятие 8\Pr_8_3.exe
Введите строку:
Широка страна моя родная
Много в ней лесов, полей и рек
Я другой такой страны не знаю,
Где так вольно дышит человек.
$
Вы ввели:
Широка страна моя родная
Много в ней лесов, полей и рек
Я другой такой страны не знаю,
Где так вольно дышит человек.

Для продолжения нажмите любую клавишу .
```

## Задача 8-4

# Копирование строк

Строки можно копировать в цикле символ за символом, для определения длины строки используем функцию **strlen()**, в конец строки вставляем нулевой символ **'\0'**. Другой способ копирования – использование библиотечной функции **strcpy()**, первым аргумент – строка, куда будут копироваться данные.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    char str1[] = "Маленькой елочке холодно зимой.";
    const int MAX = 80;
    char str2[MAX];
    for (int j = 0; j < strlen(str1); j++)
        str2[j] = str1[j];
    str2[strlen(str1)] = '\0';
    cout << str2 << endl;
    char str3[] = "Уронили мишку на пол, оторвали мишке лапу!";
    char str4[MAX];
    strcpy(str4, str3);
    cout << str4 << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

D:\Практика Сpp задачи\Занятие 8\Pr\_8\_4.exe

Маленькой елочке холодно зимой.  
Уронили мишку на пол, оторвали мишке лапу!  
Для продолжения нажмите любую клавишу . . .

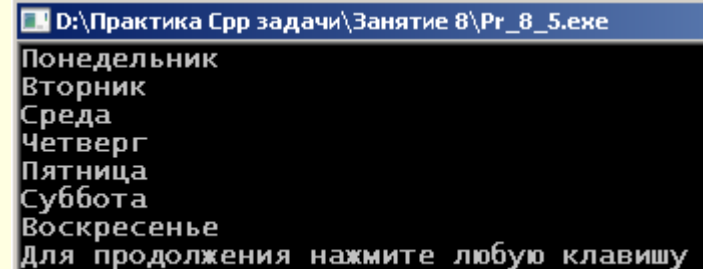


## Задача 8-5

# Массивы строк

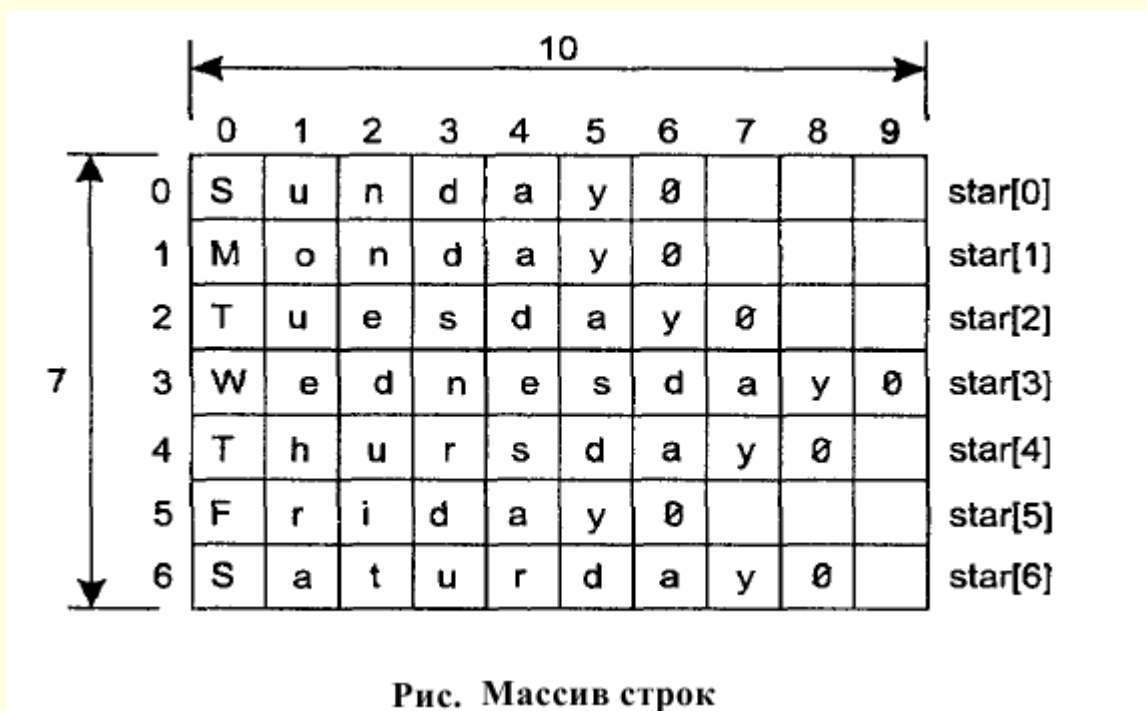
Рассмотрим в данной задаче применение массива строк. Названия дней недели помещаем в массив строк и далее выводим на печать.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    const int DAYS = 7; // количество строк в массиве
    const int MAX = 12; // максимальная длина каждой из них
    // массив строк
    char star [ DAYS ][ MAX ] =
    {
        "Понедельник", "Вторник", "Среда", "Четверг",
        "Пятница", "Суббота", "Воскресенье"
    };
    // вывод всех строк на экран
    for ( int j = 0; j < DAYS; j++ )
        cout << star [ j ] << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



```
D:\Практика Cpp задачи\Занятие 8\Pr_8_5.exe
Понедельник
Вторник
Среда
Четверг
Пятница
Суббота
Воскресенье
Для продолжения нажмите любую клавишу
```

# Массивы строк



Так как двумерный массив — это массив массивов, то мы можем получить доступ к элементам «внешнего» массива, каждый из которых в отдельности является массивом (в нашем случае строкой). Для того чтобы сделать это, нам не нужен второй индекс. Поэтому `star[j]` — это строка под номером `j` из массива строк.

## Задача 8-6

# Строки как члены классов

Строки часто используют в качестве членов классов. В следующем примере используются строки, содержащие названия частей изделий

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
using namespace std;
class Part
{ private: char partname[30]; int number; double cost;
  public:
    void setpart ( char pname [ ], int pn, double c )
      { strcpy(partname,pname); number=pn; cost=c;}
    void showpart ( )
      { cout << "Название=" << partname << ", номер="
        << number << ", цена= " << cost << endl;}
};
int main(int argc, char *argv[])
{ SetConsoleCP(1251); SetConsoleOutputCP(1251);
  Part part1, part2;
  part1.setpart ( "муфта", 4473, 217.55 );
  part2.setpart ( "вороток", 9924, 419.25 );
  cout << "Первая деталь: \n"; part1.showpart ( );
  cout << "Вторая деталь: \n"; part2.showpart ( );
  system("PAUSE");
  return EXIT_SUCCESS;
}
```

D:\Практика Сpp задачи\Занятие 8\Pr\_8\_6.exe

```
Первая деталь:
Название=муфта, номер=4473, цена= 217.55
Вторая деталь:
Название=вороток, номер=9924, цена= 419.25
Для продолжения нажмите любую клавишу . . .
```

## Задача 8-7 (не обязательно)

# Определенные пользователем типы строк

Создадим класс **String**. Он содержит массив типа **char** размера **SZ** символов, два конструктора и методы: **display()**, **concat()**.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
class String
{ private:    // enum { SZ = 80; };
    static const int SZ = 80; char str [ SZ ];
public: String ( )      { str [ 0 ] = '\0'; }
    String ( char s [ ] ){ strcpy ( str, s ); }
    void display ( )    { cout << str; }
    void concat ( String s2 ) // сложение строк
        { if ( strlen ( str ) + strlen ( s2.str ) < SZ )
            strcat ( str, s2.str );
          else cout << "\nПереполнение!"; }
};

int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    String s1 ( "С Новым годом!  " );
    String s2 = "С новым счастьем!";
    String s3; cout << "\ns1="; s1.display ( );
    cout << "\ns2="; s2.display ( );
    cout << "\ns3="; s3.display ( );
    s3 = s1;  cout << "\ns3="; s3.display ( );
    s3.concat ( s2 );    cout << "\ns3="; s3.display ( );
    cout << endl; system("PAUSE"); return EXIT_SUCCESS;
}
```

D:\Практика Сpp задачи\Занятие 8\Pr\_8\_7.exe

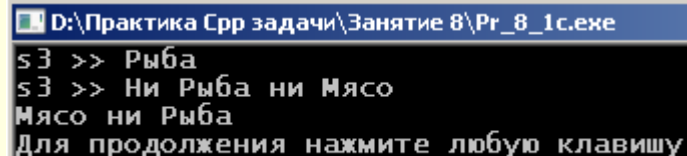
```
s1=C Новым годом!
s2=C новым счастьем!
s3=
s3=C Новым годом!
s3=C Новым годом! С новым счастьем!
Для продолжения нажмите любую клавишу
```

## Задача 8-1с

# Стандартный класс string языка C++

Определение объектов класса **string** и присваивание им значений. В этом примере нам встретится первый метод класса string – **swap()**, который меняет местами значения двух строковых объектов.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    string s1 ( "Рыба" );
    string s2 = "Мясо";
    string s3;
    s3 = s1;
    cout << "s3 >> " << s3 << endl;
    s3 = "Ни " + s1 + " ни ";
    s3 += s2;
    cout << "s3 >> " << s3 << endl;
    s1.swap(s2);    //метод класса string
    cout << s1 << " ни " << s2 << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



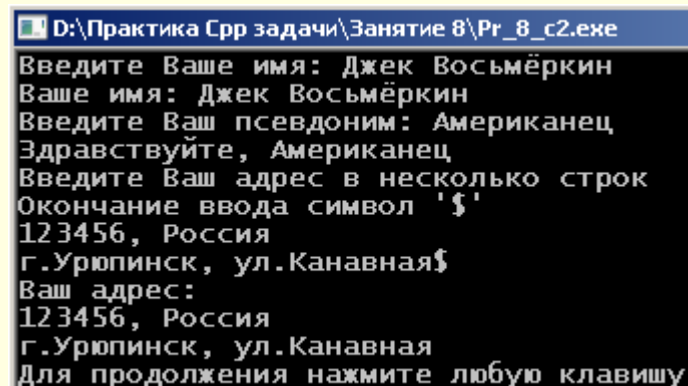
```
D:\Практика Сpp задачи\Занятие 8\Pr_8_1с.exe
s3 >> Рыба
s3 >> Ни Рыба ни Мясо
Мясо ни Рыба
Для продолжения нажмите любую клавишу
```

## Задача 8-2с

# Ввод/вывод для объекта класса string

Ввод и вывод осуществляются путем, схожим с применяемым для строкового типа. Метод **getline()** принимает ввод, который может содержать пробелы или несколько строк.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    string full_name, nickname, address;
    string str ( "Здравствуйтесь, " );
    cout << "Введите Ваше имя: ";
    getline ( cin, full_name );
    cout << "Ваше имя: " << full_name << endl;
    cout << "Введите Ваш псевдоним: ";
    cin >> nickname;
    str += nickname;
    cout << str << endl;
    cout << "Введите Ваш адрес в несколько строк\n";
    cout << "Окончание ввода символ '$'\n";
    getline ( cin, address, '$' );
    cout << "Ваш адрес: " << address << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



```
D:\Практика Сpp задачи\Занятие 8\Pr_8_c2.exe
Введите Ваше имя: Джек Восьмёркин
Ваше имя: Джек Восьмёркин
Введите Ваш псевдоним: Американец
Здравствуйтесь, Американец
Введите Ваш адрес в несколько строк
Окончание ввода символ '$'
123456, Россия
г.Урюпинск, ул.Канавная$
Ваш адрес:
123456, Россия
г.Урюпинск, ул.Канавная
Для продолжения нажмите любую клавишу
```

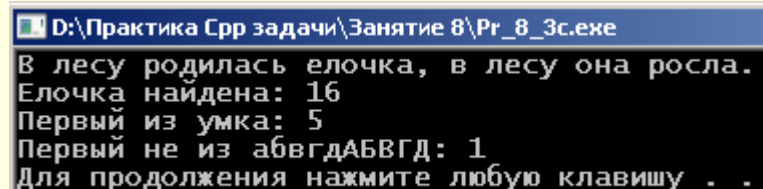
## Задача 8-3с

# Поиск объектов класса string

Класс **string** включает в себя различные методы поиска строк и фрагментов строк в объектах класса **string**. Метод **find()** – поиск подстроки в строке, для которой вызван метод. Метод **find\_first\_of()** – поиск любого символа из группы и возвращает позицию первого найденного. Метод **find\_first\_not\_of()** ищет первый символ в строке, который не входит в определенную группу символов.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    string s1 = "В лесу родилась елочка, в лесу она росла.";
    cout << s1 << endl;
    int n;
    n = s1.find ( "елочка" );
    cout << "Елочка найдена: " << n << endl;
    n = s1.find_first_of ( "умка" );
    cout << "Первый из умка: " << n << endl;
    n = s1.find_first_not_of ( "абвгдАБВГД" );
    cout << "Первый не из абвгдАБВГД: " << n << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Варианты методов: **rfind()**, переворачивает строку, **find\_last\_of()**, ищет последний символ, совпадающий с группой заданных символов и **find\_last\_not\_of()**. Все они равны -1, если цель не найдена.



```
D:\Практика Сpp задачи\Занятие 8\Pr_8_3с.exe
В лесу родилась елочка, в лесу она росла.
Елочка найдена: 16
Первый из умка: 5
Первый не из абвгдАБВГД: 1
Для продолжения нажмите любую клавишу . .
```

## Задача 8-4с

# Модификация объектов класса string

Метод **erase()** удаляет фрагмент из строки (первый аргумент - позиция первого символа фрагмента, второй - длина фрагмента). Метод **replace()** заменяет часть строки на другую строку (первый аргумент - позиция начала замены, второй - количество символов исходной строки, которое должно быть заменено, третий аргумент - строка для замены). Метод **insert()** вставляет строку, определенную

во втором аргументе, на место, определенное в первом аргументе. Метод **size()** возвращает количество символов в объекте класса **string**. Метод **append()** вставляет количество знаков в конец предложения.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <cstring>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    string s1 ( "Все хорошо, прекрасная маркиза." );
    string s2 ( "принцесса" ); string s3 ( "Приветствую " );
    cout << "s1: " << s1 << endl;
    s1.erase ( 0, 12 ); s1.insert ( 0, s3 );
    cout << "s1: " << s1 << endl; s1.replace ( 23, 7, s2 );
    s1.erase(s1.size()-1, 1); s1.append ( 3, '!' );
    cout << "s1: " << s1 << endl;
    int x = s1.find(' '); // ищем пробелы заменяем на слэш
    while(x<s1.size()) {s1.replace(x,1,"/"); x=s1.find(' ');}
    cout << "s1: " << s1 << endl;
    system("PAUSE");    return EXIT_SUCCESS;
}
```

D:\Практика Сpp задачи\Занятие 8\Pr\_8\_4с.exe

```
s1: Все хорошо, прекрасная маркиза.
s1: Приветствую прекрасная маркиза.
s1: Приветствую прекрасная принцесса!!!
s1: Приветствую/прекрасная/принцесса!!!
Для продолжения нажмите любую клавишу . . .
```



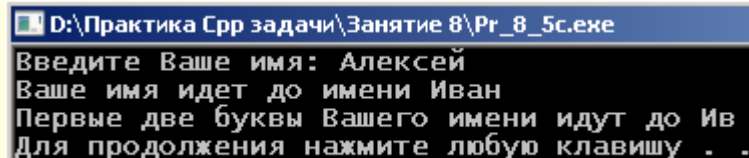
## Задача 8-5с

# Сравнение объектов класса `string`

Метод **`compare()`** для сравнения объектов класса **`string`**. Задача состоит в том, чтобы определить, являются ли строки идентичными или какая из них предшествует другой в алфавитном порядке. Во второй части программы метод **`compare()`** сравнивает только первые

две буквы слова «Иван» с первыми двумя буквами имени, (`userName`). Аргументами метода **`compare()`** являются начальная позиция `userName`, число символов для сравнения, строка (`aName`), начальная позиция и количество символов в строке `aName`.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    string aName = "Иван";    string userName;
    cout << "Введите Ваше имя: "; cin >> userName;
    if ( userName == aName ) cout << "Привет, Иван\n";
    else if ( userName < aName )
        cout << "Ваше имя идет до имени Иван\n";
    else cout << "Ваше имя идет после имени Иван\n";
    int n = userName.compare ( 0, 2, aName, 0, 2 );
    cout << "Первые две буквы Вашего имени ";
    if ( n == 0 ) cout << "совпадают ";
    else if ( n < 0 ) cout << "идут до ";
    else cout << "идут после ";
    cout << aName.substr ( 0, 2 ) << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



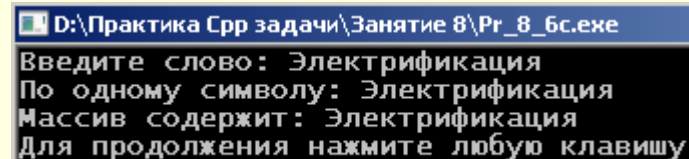
```
D:\Практика Сpp задачи\Занятие 8\Pr_8_5с.exe
Введите Ваше имя: Алексей
Ваше имя идет до имени Иван
Первые две буквы Вашего имени идут до Ив
Для продолжения нажмите любую клавишу . .
```

## Задача 8-6с

# Доступ к символам в объектах класса string

В данном примере используется метод **at()**, который остановит программу, если индекс не входит в допустимые границы.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    char mas[80];    string word;
    cout << "Введите слово: "; cin >> word;
    int wlen = word.length();    // длина строки
    cout << "По одному символу: ";
    for ( int j = 0; j < wlen; j++ )    cout << word.at(j);
    word.copy(mas, wlen, 0 ); // копируем строку в массив
    mas[wlen] = '\0'; cout << "\nМассив содержит: " << mas << endl;
    system("PAUSE");    return EXIT_SUCCESS;
}
```



```
D:\Практика Сpp задачи\Занятие 8\Pr_8_6с.exe
Введите слово: Электрификация
По одному символу: Э л е к т р и ф и к а ц и я
Массив содержит: Электрификация
Для продолжения нажмите любую клавишу
```

Вы можете также использовать перегруженную операцию `[]`, которая позволяет рассматривать объект класса `string` как массив. Однако операция `[]` не предупредит вас, если вы попытаетесь получить доступ к символу, лежащему за пределами массива (например, после конца строки). Операция `[]` ведет себя здесь так, как обращается с настоящим массивом, и это более эффективно. Но это может привести к тяжелым для диагностики программным ошибкам.

# Список задач 8-го занятия

---

1. Строковые константы и переменные
2. Считывание строки с пробелами
3. Считывание нескольких строк
4. Копирование строк
5. Массивы строк
6. Строки как члены классов
7. Определенные пользователем типы строк
- 1с. Стандартный класс `string` языка C++
- 2с. Ввод/вывод для объекта класса `string`
- 3с. Поиск объектов класса `string`
- 4с. Модификация объектов класса `string`
- 5с. Сравнение объектов класса `string`
- 6с. Доступ к символам в объектах класса `string`

## Резюме 8-го занятия

---

- Строковый тип представляет собой массив элементов типа **char**. Последний символ такой строки должен быть нулевым, **'\0'**. Строковая константа имеет специальную форму, она может быть записана удобным для нас способом: это текст, заключенный в двойные кавычки. Для работы со строками используются различные библиотечные функции. Массив строк — это массив массивов типа **char**. Создавая строковую переменную, нужно быть уверенным, что массив имеет достаточный размер для помещения в него строки. Строки, используемые в качестве аргументов в библиотечных функциях, вы сможете найти в старых программах. Обычно такое использование строкового типа в современных программах не рекомендуется

## Резюме 8-го занятия

---

- Предпочтительнее использовать для работы со строками объекты класса **string**. Эти объекты могут работать с перегруженными операциями и методами класса. Пользователю не нужно заботиться об управлении памятью при использовании объектов класса **string**.

# Контрольные вопросы

**Истинно ли следующее утверждение: операция >> прекращает считывание строки при обнаружении пробела?**

- ☐ Ложно
- ☐ Истинно

**Вы можете считывать ввод, который содержит несколько строк или текст, используя:**

- ☐ обыкновенную комбинацию cout <<
- ☐ метод cin.get() с одним аргументом
- ☐ метод cin.get() с двумя аргументами
- ☐ метод cin.get() с тремя аргументами

**Истинно ли следующее утверждение: предпочтительнее использовать строковый тип вместо стандартного класса string в своих программах?**

- ☐ Ложно
- ☐ Истинно

# Контрольные вопросы

## Строка в C++ — это:

- ☐ массив типа char
- ☐ массив типа int
- ☐ не массив

## Объекты класса string:

- ☐ заканчиваются нулевым символом
- ☐ могут быть скопированы с операцией присваивания
- ☐ не требуют управления памятью
- ☐ не имеют методов класса

## Напишите выражение, которое определяет строковую переменную city, содержащую строку длиной до 20 символов

- ☐ char city[21]
- ☐ char city[20]
- ☐ char city[19]

# Задание на дом – самостоятельная работа

---

Решить одну ( на четыре ) или две задачи ( на пять ) из предложенных 2-х задач программирования с использованием строк (СП\_2 и СП\_3)



## Задача СП\_2

Вычисление суммы. Напишите программу, которая вычисляет введенное пользователем значение выражения типа **AZAZ...A**, где **A** — целое число от 0 до 9, а **Z** — знак «+» или «-». Ниже представлен рекомендуемый вид экрана во время работы программы.

Введите строку для вычисления  
цифра знак цифра знак цифра...  
**1+3-2+5+7-9**  
ответ = 5

## Задача СП\_3

---

Вычисление количества цифр в строке. Написать программу, которая подсчитывает количество цифр во введенной строке слов.

Ниже представлен рекомендуемый вид экрана во время работы программы.

Введите строку слов и цифр

**На полке лежат 5 книг. Количество страниц = 1278.**

ответ = 5

# Наиболее «популярные» ошибки

<b>xxx.h: No such file or directory</b>	не найден заголовочный файл 'xxx.h' (неверно указано его имя, он удален или т.п.)
<b>'xxx' undeclared (first use this function)</b>	функция или переменная 'xxx' неизвестна
<b>missing terminating " character</b>	не закрыты кавычки "
<b>expected ;</b>	нет точки с запятой в конце оператора <b>в предыдущей строке</b>
<b>expected }</b>	не закрыта фигурная скобка

# Литература

---

1. Роберт Лафоре. Объектно-ориентированное программирование в С++
2. В. И. Шупляк. С++ Практический курс
3. В. В. Подбельский. Язык С++. Базовый курс

# Конец

---