Facultad de Ingeniería y Ciencias Hídricas
FICH
UNL

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

Ingeniería Informática

PROGRAMACIÓN ORIENTADA A OBJETOS

UNIDAD 5

La clase std::string

2011

UNIDAD 5

La clase std::string

Ejercicio 5.1

Proponga una clase con métodos para incorporar un arreglo lineal de N elementos de tipo string. La clase debe tener métodos para obtener el primero elemento de la lista de acuerdo al orden alfabético, y eliminar las palabras que comienzan con la sílaba 'mar'.

Ejercicio 5.2

Escriba una clase *AnalizaFrase* que posea como atributo una frase ingresada por el usuadio y métodos para determinar:

a) La cantidad de vocales de la frase.
b) La cantidad de consonantes.
c) La cantidad de letras.

Proponga un programa cliente que la utilice.

Ejercicio 5.3

Escriba una clase *CortaFrase* que posea como atributo un arreglo dinámico de strings. La clase debe tener un constructor que reciba una frase y coloque en el arreglo generado dinámicamente cada una de las palabras de la frase y un destructor que libere la memoria. Además, implemente una función que reciba un entero y devuelva la palabra que se encuentra en dicha posición del arreglo. Pruebe la clase en un programa cliente.

Ejercicio 5.4

Diseñe una clase llamada *ManejadorMail* que posea como atributo una lista con los nombres y apellidos de docentes y alumnos de la FICH. La clase debe poseer métodos que permitan agregar personas a la lista gestionada por la clase y un método *VerMail(...)* que debe recibir el apellido de la persona y devolver su dirección de correo electrónico (suponga que no existen apellidos repetidos). El dominio asignado a la Facultad para el e-mail es: fich.unl.edu.ar, y el nombre de usuario se forma con la inicial del nombre y el apellido. Por ejemplo: para Marelli Jorge, la dirección de correo es: jmarelli@fich.unl.edu.ar.

Ejercicio 5.6

Escriba un programa C++ que lea una lista de N palabras y las acumule en un único objeto de tipo string de modo que las palabras aparezcan una junto a la otra separadas por un blanco. Mostrar el objeto con las palabras acumuladas.

Ejercicio 5.7

Escriba un programa que permita ingresar un string con una operación matemática simple (suma, resta, multiplicación o división, con solo dos operandos) y muestre el resultado. Por ejemplo, si la cadena ingresada es “7.5+12”, el programa debe reconocer el operador '+' (podría ser '+', '-', '/' o '*'), separar los dos substrings de los números (“7.5” y “12”), convertirlos a double y calcular y mostrar el resultado. Para convertir de string a double puede utilizar la función *atof(...)* de la biblioteca *cstdlib*.

Ejercicio 5.5 (OPCIONAL)

Diseñe una clase para modelar una persona a partir de los datos siguientes: Apellido, Nombres, Calle y Número. Escriba un programa que detecte y muestre los nombres de las personas que viven en la misma calle, indicando primero la calle correspondiente (no incluir aquellas personas que sean únicas en su calle).

Ejercicio 5.8 (OPCIONAL)

Escriba una función llamada *Jerigonza(...)* que reciba una cadena con una frase y retorne la frase en jerigonza. Por ejemplo, si recibe *“Saludos Terricolas!”* debe retornar *“Sapalupudopos Teperripicopolapas!”*.

Ejercicio 5.9

Escriba un programa C++ que permita al usuario ingresar un párrafo de texto y lo guarde en un objeto de tipo string y luego:

a) Reemplace todas la ocurrencias de la frase *“open source”* por *“código abierto”* (notar que puede aparecer más de una vez, o también podría no aparecer nunca).
b) Corrija mayúsculas y minúsculas: las mayúsculas solo van al comienzo de una oración, es decir, en la primer letra y luego de un punto; el resto debe estar en minúsculas.
c) Muestre el resultado.

Ejercicio 5.9 (OPCIONAL)

Escriba una función que reciba como parámetros dos frases. La función debe devolver en una única cadena tantas lineas como palabras haya en la segunda. Cada una de las i-ésimas líneas debe ser una copia de la primer frase con la i-ésima palabra de la segunda frase insertada en la anteúltima posición. Por ejemplo, la cadena:

El gato así hace
El gato es hace
El gato como hace
El gato se hace
El gato mantiene
El gato a hace
El gato un hace
El gato estudiante hace
El gato ocupado hace
El gato por hace
El gato cuarenta hace
El gato segundos hace

debe formarse a partir las frases: *"El gato hace"* y *"así es como se mantiene a un estudiante ocupado por cuarenta segundos".*

Cuestionario

- ¿Qué ventajas tiene el uso de std::strings frente a las cstrings?
- Según su opinión, sin conocer la estructura interna de la clase std::string, ¿qué atributos posee esta clase para llevar a cabo sus funciones?
- ¿Qué devuelve el operados *sizeof()* al ser aplicado a un objeto de tipo std::string?¿Por qué sucede esto?