

Actividad | #3| Calculo de RFC

Lenguajes de Programación

Ingeniería en Desarrollo de Software

TUTOR: Urbano Francisco Ortega Rivera

ALUMNO: Alejandra Ibarra Carmona

FECHA: 10/08/2024

Índice

Introducción ………………………………………………………3

Descripción ………………………………………………………3

Justificación ………………………………………………………3

Desarrollo …………………………………………………………4

* Codificación……………………………………………….5
* Prueba del sistema………………………………………6

Conclusión………………………………………………………..7

Referencias……………………………………………………….8

Introducción

En esta actividad se estará mostrando como calcular nuestro RFC a partir de un código en lenguaje C++ como veníamos sabiendo este lenguaje esta orientado a objetos en el cual también empleamos una variable. Existen también lo que llamamos arreglos que se conoce como una colección de variables relacionadas a las que se hace referencia a un nombre en común, un arreglo hace referencia a un conjunto de variables que se almacenan en la memoria de manera contigua con el mismo nombre, y para diferenciar los elementos de un arreglo se coloca un índice.

En lenguaje C++ se conoce como tipo de dato compuesto y pueden tener una o varias dimensiones:

* Indicé de un arreglo: indica el numero de posicionamiento de un arreglo y en C++ va desde la posición 0 hasta la -1
* Elemento de un arreglo: se trata de un valor en particular dentro de la estructura del arreglo y para acceder a el es importante indicar el número de posición.

Descripción

Los arreglos de dimensiones es una lista de variables todas del mismo tipo a lo que se hace referencia por medio de un nombre en común. Cuando queremos acceder a un elemento de un arreglo indexándolo por medio de un numero de elemento se dice acceder a su primer elemento y debe usarse el indicé igual a 0, para indexar un arreglo se debe especificar el indicé del elemento que nos interesa dentro de corchetes.

C++ almacena arreglos de una sola dimensión en una localización de memoria contigua en el primer elemento, es la posición más baja, es posible usar el valor del elemento de un arreglo donde se usaría una variable más sencilla o una constante. También tenemos a los arreglos multidimensionales son también conocidos como matrices, en este sentido mencionado se le llama matriz mxn a un conjunto rectangular que este compuesto por n filas y n columnas siendo estos números naturales.

Justificación

Dentro de los arreglos multidimensionales los que mas se utilizan son los datos conocidos como dimensiones, y se conocen como lo mencionamos y son los arreglos bidimensionales o matrices. Un apuntador se define como una variable que contiene una dirección de memoria, y esta puede verse como un conjunto de celdas numeradas y ordenadas cada una de ellas puede almacenar un byte de información, el número de cada celda se conoce como dirección.

Cuando creamos una variable el compilador reserva el numero suficiente de celdas de memoria que son los bytes y se encargan de reservar aquellos que no se traspalen, una variable puede almacenar una dirección de memoria suficiente llamada puntero, para saber dónde se encuentra la memoria almacenada con variable mediante un operador se utiliza de referencia &, un apuntador siempre está asociado con el tipo de dato que apunta, en lenguaje C++ se define una constante especial llamada NULL cabe mencionar que esta no apunta a ningún lugar valido en la memoria como la que se menciono anteriormente con los bytes esta tiene un valor de cero, lo que nos recomiendan es iniciar con apuntadores con una dirección valida o en este caso NULL esto para evitar sobrescribir información.

Desarrollo

* Codificación

#include <iostream>

#include <string>

//Función permite obtener la primera vocal interna de una cadena de texto

char obtenerPrimeraVocalInterna(const std::string& str) {

for (size\_t i = 1; i <str.length(); ++i) {

char c = str [i];

if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U' )

return c;

}

return 'X'; // si no se encuentra ninguna vocal interna se una la X

}

//Función principal para calcular RFC

std::string calcularRFC(const std::string& nombre, const std::string& apellidoPaterno, const std::string& apellidoMaterno, const std::string& fechaNacimiento) {

std::string rfc;

//Se obtiene la letra inicial y la primera vocal interna del apellido paterno

char letraInicial = apellidoPaterno[0];

char primeraVocalInterna = obtenerPrimeraVocalInterna(apellidoPaterno);

//se obtiene la letra inicial del apellido materno o se usa un x si no la tiene

char inicialApellidoMaterno = (!apellidoMaterno.empty())? apellidoMaterno[0] : 'X';

//se obtiene la letra inicial del apellido materno o se usa un x si no la tiene

char inicialNombre = nombre[0];

//Se obtienen los dos últimos dígitos del año de nacimiento

std::string anio = fechaNacimiento.substr(2, 2);

//Se obtiene el mes y el día de nacimiento

std::string mes = fechaNacimiento.substr(5, 2);

std::string dia = fechaNacimiento.substr(8, 2);

//Se construye rfc

rfc = letraInicial;

rfc += primeraVocalInterna;

rfc += inicialApellidoMaterno;

rfc += inicialNombre;

rfc += anio;

rfc += mes;

rfc += dia;

return rfc;// Retornar el RFC

}

int main () {

std::string nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento;

std::cout << "Ingrese su nombre: ";

std::getline (std::cin, nombre);

std::cout << "Ingrese el apellido paterno: ";

std::getline (std::cin, apellidoPaterno);

std::cout << "Ingrese el apellido materno: (si no tienen pulse Enter) ";

std::getline (std::cin, apellidoMaterno);

std::cout << "Ingrese su fecha de nacimiento (YYYY/MM/DD): ";

std::getline (std::cin, fechaNacimiento);

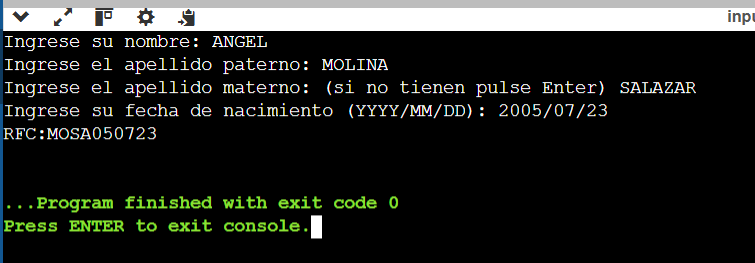
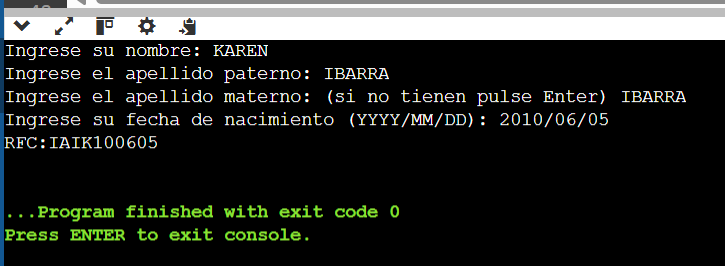
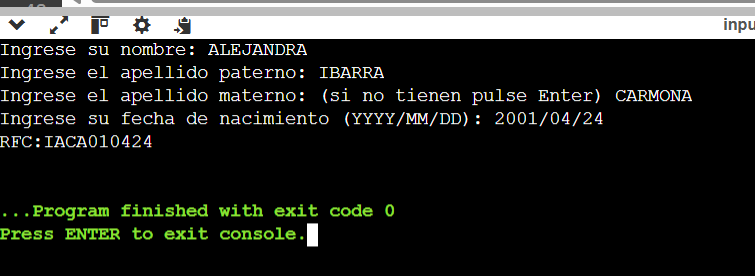
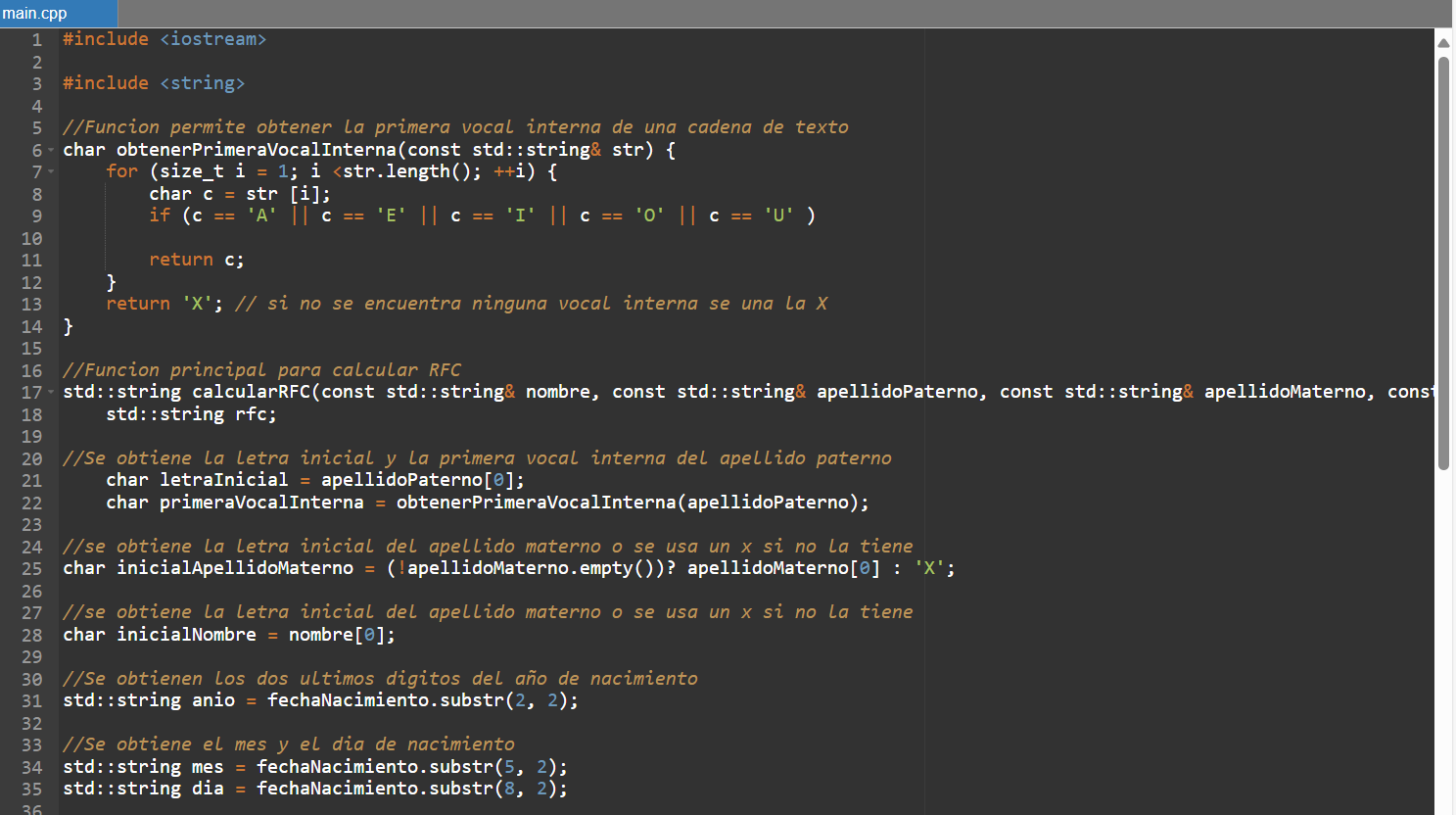
std::string rfc = calcularRFC (nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento);

std::cout << "RFC:" << rfc << std::endl;

return 0;

}

* Prueba en sistema



Conclusión

En esta actividad creamos un código para generar nuestro RFC, durante este proceso tienes que realizar varios string los cuales sirven para seleccionar las letras correspondientes de nuestra información. También hablaremos de las clases que estas se definen como un patrón para construir objetos y como ya sabemos este lenguaje que se empleo durante estas tres actividades fue C++ que esta orientado a objetos y este se define como un tipo de variable de una clase determinada, es importante distinguir entre clases y objetos ya que clase la denominamos como una declaración y no se asocia con ningún objeto de modo que no puede recibir mensajes ni procesarlos cosa que si hacen los objetos, se puede acceder a los datos y métodos de una clase de la misma forma que se accede a un campo de estructura. Las variables y los métodos son de instancia, para usarlos se deben crear instancias de clases que necesita o bien tener una referencia misma.

Referencias