**Código Fuente Ejercicio 2 de la videoconferencia**

//Ejecicio2 de la Videoconferencia

/\*Este ejercicio permite pedir cita ingresando por teclado los datos de la Hora, Minutos,

Segundos, los mismos que son validados para que no se ingrensen datos incorrectos y ademas

en forma de comentarios se va explicando lo que realiza cada linea de codigo relevante \*/

#include <iostream> //Es un componente de la biblioteca estandar de C++ que sirve para la entrada y salida de datos.

#include <iomanip> //Esta libreria permita hacer uso de setfill y setw

#include<stdlib.h>

#include<windows.h> //Sirve para utilizar la funcion system("cls") para limpiar la pantalla

using namespace std;/\*Esta linea de codigo sirva para el ingreso y salida de datos y remplaza a todas las siguientes lineas

using std::cout; //Sirve para imprimir los datos por pantalla

using std::endl; //Sirve para hacer un Enter

using std::cin; //Sirve para el ingreso de datos desde el teclado

using std::setfill; //Sirve para rellenar con un determinado caracter

using std::setw; //Sirve para alinear los datos \*/

// Definición del tipo de dato abstracto (ADT) Tiempo

class Tiempo{ //Declaracion de la clase Tiempo

private: //Los : nos indican que las siguientes lineas son de tipo private

int hora; // Declaracion del atributo hor (0 - 23 formato de reloj de 24 horas)

int minuto; // Declaracion atributo minuto (0 - 59)

int segundo; // Declaracion atributo segundo (0 - 59)

public:

Tiempo( ); //Declaracion del Metodo Constructor y es de tipo vacio

void capturarDatos(); //Declaracion Metodo para ingresar los datos por pantalla

void estableceHora(int,int,int); //Establece hora, minuto, segundo

void imprimeUniversal(); //Imprime el tiempo en formato universal

void imprimeEstandar(); //Imprime el tiempo en formato estándar

}; //Fin de la clase Tiempo

//Metodo Constructor Tiempo el mismo que inicializa cada dato en cero y garantiza que los objetos Tiempo comiencen en un estado consistente.

Tiempo::Tiempo(){

hora = minuto = segundo = 0;

} // Fin del constructor Tiempo

//Metodo EstableceHora\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Tiempo::estableceHora(int \_hora,int \_minuto,int \_segundo){

hora = ( \_hora >= 0 && \_hora < 24 ) ? \_hora : 0; //? Se pregunta si cumple la comdicion si no, establece la hora en cero

minuto = ( \_minuto >= 0 && \_minuto < 60 ) ? \_minuto : 0; //? Se pregunta si cumple la comdicion si no, establece los minutos en cero

segundo = ( \_segundo >= 0 && \_segundo < 60 ) ? \_segundo : 0;//? Se pregunta si cumple la comdicion si no, establece los segundos en cero

} // fin de la función estableceHora

//Metodo para Validar los datos numericos\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int validar\_num(char numero[50]){

int i = 0, sw = 0, j=0; //sw nos indica si la cadena contiene dijitos no numericos

j=strlen(numero); //j es el tamaño de la cadena

while (i<j && sw==0) { //Sentencia while para recorrer la cadena

if(isdigit(numero[i])!=0) { //Mientras sea digito se sigue recorriendo la cadena

i++;

}

else {

sw = 1; //Si al recorrer la cadena encuentra algo que no sea numero, lo valida

}

}

return sw;

}

// Función para capturar datos para Clase\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Tiempo::capturarDatos(){ //Los :: significa que capturarDatos es de la clase Tiempo

char \_hr[10],\_mn[10],\_sg[10]; //Variables de tipo String para poder validarlos

int hr=0, mn=0, sg = 0,sw=0; //Aqui se guardaran las variables una vez validadas

char \_opcion[10]; //Variable \_opcion de tipo char

int opcion=0; //Recoge el valor si la opcion se de tipo numerico

do{ //Este do sirve la repetir mientras la opcion no sea 1 o 2

cout<<"\n\tRESTAURANTE GRUPO1"<<endl;

cout<<"Escoja una opcion."<<endl;

cout<<"1. Desea reservar una cita."<<endl;

cout<<"2. Salir"<<endl;

cout<<"Opcion: " ;

cin.getline(\_opcion,10,'\n'); //Es la mejor opcion para ingresar datos de tipo char

sw = validar\_num(\_opcion); //Esta linea llama al metodo validar\_num() y valida que se elija 1 o 2

if(sw==0){

opcion = atoi(\_opcion); //Si los datos son de tipo int, transforma el dato string

}

system("cls"); //Limpia la pantalla

}while(opcion != 1 && opcion != 2); //Fin do...whwile

switch(opcion){ //Segun la opcion en el elegida en el Menú se ejecuta el codigo

case 1: //Caso1

do{

cout<<"\n\tBIENVENIDO A SU RESTAURANTE GRUPO1 \n\t \t\"Reserve su cita\" "<<endl;

fflush(stdin);

do{

cout <<"\nDigite la Hora (0 - 23): "; //Pide la hora

cin.getline(\_hr,10,'\n'); //Cuando se llega a esta linea hay que ingresar la Hora

sw = validar\_num(\_hr); //Se valida la Hora ingresada

if(sw ==0){

hr = atoi(\_hr); //Si los datos no son numericos nos vuelve a pedir nuevamente la hora

}else{

cout<<"Ingrese la hora correctamente"; //Mensaje para ingresar nuevamente la Hora

}

}while(sw!=0);

do{

cout <<"\nDigite los Minutos (0 - 59): "; //Pide los minutos

cin.getline(\_mn,10,'\n'); //Ingreso de los minutos

sw = validar\_num(\_mn);

if(sw == 0){

mn = atoi(\_mn); //Si los datos no son numericos nos vuelve a pedir nuevamente los minutos

}else{

cout<<"Ingrese los minutos correctamente!"; //Mensaje para ingresar nuevamente los Minutos

}

}while(sw !=0);

do{

cout <<"\nDigite los Segundos (0 - 59): "; //Pide los segundos

cin.getline(\_sg,10,'\n'); //Ingreso de los segundos por teclado

sw = validar\_num(\_sg);

if(sw == 0){

sg = atoi(\_sg); //Si los datos no son numericos nos vuelve a pedir nuevamente los segundos

}else{

cout<<"Ingrese los segundos correctamente!"; //Mensaje para ingresar nuevamente los segundos

}

}while(sw !=0);

//Si los datos son numericos, se validan que esten dentro de los rangos de Hora, Minutos, Segundos, caso contrario se piden nuevamente

if(sg<0||sg>59||mn<0||mn>59||hr<0||hr>23){

system("cls");

cout<<"\n\n¡¡¡¡ATENCION: Algo anda mal "<<endl; //Mensaje que algo esta mal

cout<<"Introduzca los datos correctamente"<<endl; //Mensaje

}

}while(sg<0||sg>59||mn<0||mn>59||hr<0||hr>23); //Se repite el switch si los valores estan fuera de rango

system("cls"); //Limpia la pantalla

estableceHora(hr,mn,sg); //Establece la Hora en su formato correcto

cout<<"\n\n\tBIENVENIDO A SU RESTAURANTE GRUPO1"<<endl; //Mensaje de Pantalla

cout << "\nSu cita a sido reservada EXITOSAMENTE "<<endl; //Si los datos son corretos se imprime su cita

imprimeUniversal(); //Establece el Tiempo en formato Universal

imprimeEstandar(); //Establece el Tiempo en formato Estandar

cout<<"\n\nNo olvide acudir a su cita"<<endl; //Mensaje para acudir a la cita

cout<<"\n\t-----------Le Esperamos!-----------"<<endl; //Mensaje

cout << "\n"<<endl; //Salto de linea

break;

case 2: system("cls"); //Opcion 2

cout<<"\n\n\tRESTAURANTE GRUPO1"<<endl;

cout<<"\nNo a concretado ninguna cita"<<endl; //Mensaje si no se elige una cita

cout<<"\n--------HASTA PRONTO--------"<<endl; //Mensaje de despedida

cout << "\n"<<endl;

break;

default: "...........Adios............"; //No hace nada

}

} // Fin de la funcion capturarDatos

//Funcion para imprimir el Tiempo en formato universal\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Tiempo::imprimeUniversal(){

cout << "\n\tHora universal:";

cout << setfill( '0' ) //setfill rellena con un caracter, en este caso con cero ('0') donde haya datos con un digito justificados a la derecha.

<< setw( 2 ) << hora << ":" //setw Sirve para dar formato de ancho a los numeros ej. (sewt 2 = 00) (sewt 3 = 000)

<< setw( 2 ) << minuto << ":"

<< setw( 2 ) << segundo;

} //Fin de la función imprimeUniversal

//Funcion para imprimir el Tiempo en formato estándar\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Tiempo::imprimeEstandar(){

cout << "\n\tHora estandar :";

cout<< setfill( '0' ) //Rellena los datos con el cero donde haya un solo digito

<< setw( 2 )<<( ( hora == 0 || hora == 12 ) ? 12 : hora % 12 )<< ":" //Si es 00 o 12 nos devuelve un 12 si no modulo % 12

<< setw( 2 ) << minuto << ":" //el formato de hora es de dos digitos y son rellenados por setfill (1 digito)

<< setw( 2 ) << segundo

<< ( hora < 12 ? " AM" : " PM" ); //Mensaje si es la Manaña AM si no PM

} //Fin de la función imprimeEstandar

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menú Principal\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(){

//int hora,minuto,segundo;

Tiempo horaCita1 = Tiempo(); //Instanciacion del objeto horaCita1

// Capturando los datos de la hora de la cena\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

horaCita1.capturarDatos(); //Llama a funcion capturaDatos.

system ("pause");

return 0;

} //Fin de main