/\*

Programming Assignment: Course Project Week 7

Developer: Marvin Ziegler

Date: December 19, 2020

Purpose: Course Project - DJ Program

\*/

#include<iostream> //library for streaming input & output

#include<string> //library to allow strings as input & output

#include<ostream> //constructor header

#include<vector> //library for dynamic arrays

#include<fstream> //contains declarations necessary for file operations

#include<cstdlib> //library to allow usage or rand & srand

#include<ctime> //library for time function

#include <limits.h> //library for defining constants

//namespace - in order to avoid name collisions

//std - official standard library

using namespace std;

int playlistNumber; //variable for user to input playlist number

string playlistName;

string classicalPlaylistUno[] = { "Opus 18 The Tempest (by Tchaikovsky)",

"Adagio maestoso (by Mozart - Posthorn)", "Opus 20 Swan Lake (by Tchaikovsky)",

"Minuetto (by Mozart - Posthorn)", "Orchestral Suite No. 1 (by Tchaikovsky)",

"Concertante (by Mozart - Posthorn)", "Opus 44 Piano Concerto No. 2 in G (by Tchaikovsky)",

"Rondeau (by Mozart - Posthorn)", "Allegro (by Mozart - Eine kleine Nachtmusik)" };

string classicalPlaylistDue[] = { "Opus 45 Capriccio Italien in A (by Tchaikovsky)",

"Andantino in D minor (by Mozart - Posthorn)", "Opus 48 Serenade in C for Strings (by Tchaikovsky)",

"Minuetto - Trio 1 and 2 (by Mozart - Posthorn)", "Romanze: Andante (by Mozart - Eine kleine Nachtmusik)",

"Opus 49 1812 Overture (by Tchaikovsky)", "Finale: Presto (by Mozart - Posthorn)",

"Menuetto: Allegretto (by Mozart - Eine kleine Nachtmusik)", "Opus 71 The Nutcracker (by Tchaikovsky)",

"Rondo: Allegro (by Mozart - Eine kleine Nachtmusik)" };

string popularPlaylistUno[] = { "Surfin\' Safari (Beach Boys)", "Ring Ring (ABBA)",

"County Fair (Beach Boys)", "Ten Little Indians (Beach Boys)", "Disillusion (ABBA)",

"Chug-A-Lug (Beach Boys)", "Another Town, Another Train (ABBA)", "Little Girl (Beach Boys)",

"People Need Love (ABBA)", "409 (Beach Boys)", "I Saw It in the Mirror (ABBA)" };

string popularPlaylistDue[] = { "Surfin\' (Beach Boys)", "Nina, Pretty Ballerina (ABBA)",

"Heads You Win-Tails I Lose (Beach Boys)", "Love Isn\'t Easy (ABBA)",

"Summertime Blues (Beach Boys)", "Me and Bobby and Bobby\'s Brother (ABBA)",

"Cuckoo Clock (Beach Boys)", "He Is Your Brother (ABBA)", "Moon Dawg (Beach Boys)",

"Ring Ring (English Version) (ABBA)", "The Shift (Beach Boys)" };

string fourPlaylists[4][11] = { classicalPlaylistUno[9], classicalPlaylistDue[10],

popularPlaylistUno[11], popularPlaylistDue[11] };

string searchTerm; //variable for user to input search term

string playlists[4] = { "Rhythm & blues", "Rock", "Classical", "Popular" };

int main()

{

char selectIt; //variable to select from main menu

//playlist name

string somethingSpecial[] = { "Stepping out", "Good Time Tonight",

"Take My Heart", "Be My Lady", "Get Down On It", "Pass It On",

"Stand Up and Sing", "No Show" };

string specialThings[] = { "Could I Be Dreaming", "He's So Shy",

"The Love Too Good to Last", "Evil", "Save This Night For Love",

"We\'ve Got the Power", "Where Did the Time Go?", "Special Things",

"Here Is Where Your Love Belongs" };

//main display menu

//function prototypes

void addNewSong();

void removeSongs(string somethingSpecial[], string specialThings[]);

void createPlaylist();

void editPlaylist();

void searchMusic();

void nextCommercial();

void reviewPlaylists();

void saveFile(string classicalPlaylistUno[], string classicalPlaylistDue[],

string popularPlaylistUno[], string popularPlaylistDue[]);

while (true)

{ //conditional statement for while not logged off

cout << "\tWolfman Jack DJ song and playlist.\n";

cout << " Please select an option: \n";

cout << " 1 Add Songs to Collection\n";

cout << " 2 Delete Songs from Collection\n";

cout << " 3 Create Playlist\n";

cout << " 4 Edit Playlist\n";

cout << " 5 Search Music\n";

cout << " 6 Check for Time Before Next Commercial\n";

cout << " 7 Playlist Reports\n";

cout << " 8 Save Playlists to File\n";

cout << " 9 Log Off\n\n";

cout << "Enter a selection 1 through 9\t";

cin.clear(); //resetting error flags

cin >> selectIt;

switch (selectIt)

{

case '1':

addNewSong();

break;

case '2':

removeSongs(somethingSpecial, specialThings);

break;

case '3':

createPlaylist();

break;

case '4':

editPlaylist();

break;

case '5':

searchMusic();

break;

case '6':

nextCommercial();

break;

case '7':

reviewPlaylists();

break;

case '8':

saveFile(classicalPlaylistUno, classicalPlaylistDue, popularPlaylistUno, popularPlaylistDue);

break;

case '9':

"Logging off from application....";

return 0;

break;

default:

cout << "Not a correct entry\n\n";

break;

} //end of switch statement

}

} //end of main method

/\*

bool is\_numeric(string str) //method is convert integer to string

{

for (int i = 0; i < str.length(); i++) //increment through each element

if (isdigit(str[i]) == false) //check to see if element is digit (number)

return false;

return true;

}

\*/

void addNewSong()

{

string songName; //added song

vector<string>addedSongsArray; //array to contain songs from DJ

int numberOfSongs = -1; //initializing current number of songs

bool is\_digit = false; //

while (true)

{

cout << "How many songs would you like to enter?\t";

/\*below two statements clear the stdin buffer\*/

cin.clear(); //resetting error flags

cin.ignore(INT\_MAX, '\n'); //reads and ignores everything until EOF (end of file)

cin >> numberOfSongs;

if (numberOfSongs == 0) {

cout << "The value is a not a number. Please enter a positive number." << endl;

continue;

}

else {

break;

}

//string str = to\_string(numberOfSongs);

//is\_digit = is\_numeric(str);

//if (is\_digit == false)

//{

// cout << "The value is a not a number. Please enter a number." << endl;

// cin >> numberOfSongs;

//}

}

addedSongsArray.resize(numberOfSongs);

cin.ignore();

for (unsigned int d = 0; d < addedSongsArray.size(); d++) //loop to enter requested number of songs

{

cout << "Enter song number " << d + 1 << " out of " << numberOfSongs << ": "; //prompt user for each song

getline(cin, songName); //get the entire name of each song

addedSongsArray[d] = songName; //filling songs in array positions

} //end of for loop

cout << "The list of songs added is as follows:\n\n";

for (unsigned int e = 0; e < addedSongsArray.size(); e++)

{

/\*

memory locations start at 0, not 1

so displaying memory locations.

starting at e + 1 instead of e

\*/

cout << "Song number " << e + 1 << " is: " << addedSongsArray[e] << "\n";

} //end of for loop

} //end of function

void removeSongs(string somethingSpecial[], string specialThings[])

{

char selection;

int count = 0;

int songNumb; //removed Song menu

cout << "Delete songs\n";

cout << "Which album would you like to remove a song from?: \n";

cout << "1 - Something Special (by Kool & the Gang)\n";

cout << "2 - Special Things (by The Pointer Sisters)\n";

cout << "Select a number\t";

cin >> selection; //getting user input

cin.ignore();

switch (selection)

{

case '1': //1st playlist

cout << "Something Special (by Kool & the Gang)\n";

cout << "Which song would you like to remove?: \n";

for (int a = 0; a < 8; a = a + 1) //incrementing through songs in Something Special

{ //displaying songs to the user

cout << "Song number " << a + 1 << " is " << somethingSpecial[a] << endl;

} //end of for loop

cout << "Enter a selection, 1 thru 8: "; //prompting user to enter choice

cin >> songNumb; //getting user input

cout << somethingSpecial[songNumb - 1] << " has been removed.\n"; //notify user of removal

break;

case '2': //2nd playlist

cout << "Special Things (by The Pointer Sisters)\n";

cout << "Which song would you like to remove?: \n";

for (int b = 0; b < 9; b = b + 1) //increment through songs in Special Things

{ //displaying songs to the user

cout << "Song number " << b + 1 << " is " << specialThings[b] << endl;

} //end of for loop

cout << "Enter a selection, 1 thru 9: "; //prompt user to enter choice

cin >> songNumb; //getting user input

cout << specialThings[songNumb - 1] << " haspro been removed\n"; //notify user of removal

break;

default:

cout << "Not a correct entry\n"; //reply in case user doesn't enter 1 or 2

break;

} //end of switch statement

} //end of removeSongs function

void createPlaylist() //function for creating a playlist

{ //display menu to create a playlist

cout << "Create Playlists\n";

cin.ignore();

cout << "The current playlists are as follows:\n";

for (int g = 0; g < 4; g = g + 1) //loop to increment through current playlists

{

if (playlists[g] != "")

{

cout << g + 1 << ": " << playlists[g] << endl; //display playlist name

}

}

cout << "Which additional playlist would you like to create?\t"; //prompt user to enter new playlist

getline(cin, playlistName);

cout << "Creating \"" << playlistName << "\".\n\n"; // notify user that playlist is being created

}

void editPlaylist() //function for editing a playlist

{ //display menu to edit playlist

cout << "Edit Playlists\n";

cout << "The available playlists are as follows:\n";

for (int b = 0; b < 4; b = b + 1) //loop to increment through current playlist

if (playlists[b] != "")

{

//adding 1 to loop counter variable since the computer starts to

//count from 0 instead of 1

cout << b + 1 << ": " << playlists[b] << endl; //display playlist name

}

cout << "Which Playlist would you like to edit?\t"; //prompt user to select a playlist

cin >> playlistNumber;

cout << "Editing \"" << playlists[playlistNumber - 1] << "\" playlist\n\n"; //notify user of playlist being edited

}

void searchMusic() //function for searching for music

{ //display menu to search for music

cout << "Search Music\n";

cout << "Please Enter a Search Term\t";

cin.ignore(); //clears buffer from prev input

getline(cin, searchTerm); //getting a search term from user

cout << "Searching for \"" << searchTerm << "\"\n\n"; //notify user of search

}

void nextCommercial() //function to search for next commercial

{

//constants

const int MAX = 540; //max time before commercial is 9 mins

const int MIN = 29; //minimum time is 29 seconds

//variables

int totalSeconds; //total time not considering minutes and seconds

//get the system time

unsigned seed = time(0);

//seed the random number generator

srand(seed);

//calculate time in seconds

totalSeconds = ((rand() % MAX - MIN + 1) + MIN);

int minutes = totalSeconds / 60; //getting time in minutes

int seconds = totalSeconds % 60; //getting time in minutes

if (totalSeconds < 60) { //if less than 60 secs, 0 mins

minutes = 0;

}

//displaying time before next commercial

cout << "\nThe time before the next commercial is " <<

minutes << " minutes and " << seconds << " seconds.\n\n";

}

void reviewPlaylists() //function to review playlists

{

cout << "Reviewing playlists\n\n";

cout << "Playlists are as follows: \n\n";

cout << "Classical Playlist 1\n";

cout << "Classical Playlist 2\n";

cout << "Popular Playlist 1\n";

cout << "Popular Playlist 2\n";

} //end of review playlists function

void saveFile(string classicalPlaylistUno[], string classicalPlaylistDue[],

string popularPlaylistUno[], string popularPlaylistDue[])

{

char choice; //variable to process user choice

string nameFromArray = ""; //string variable for storing names from array

string filename; //variable to store name of newly created file

bool fileWritten = false; //variable to check to see if a file was written

//display menu

do

{

cout << "\*\*\*\* Save Playlists to File \*\*\*\*\n\n";

cout << "Which playlist would you like to save as a textfile?:\n";

cout << "\t1 - Classical Playlist 1\n";

cout << "\t2 - Classical Playlist 2\n";

cout << "\t3 - Popular Playlist 1\n";

cout << "\t4 - Popular Playlist 2\n";

cout << "\t5 - Return to Main menu\n";

cout << "Enter an option, 1 thru 5:\t";

cin >> choice; //getting users choice

//create a file object named savedPlaylistFile;

ofstream savedPlaylistFile;

switch (choice)

{

case '1':

//creating and opening a new file from file object

savedPlaylistFile.open("Classical Playlist 1.txt");

filename = "Classical Playlist 1.txt"; //variable to hold name of text file

//loop to go through array & save each element to a file

for (int b = 0; b < 9; b = b + 1) {

if (classicalPlaylistUno[b] != "") //what to do if array location not empty

{

//write song name to file and then hit carriage return

savedPlaylistFile << classicalPlaylistUno[b] << endl;

fileWritten = true;

} //end of 'if' statement

else

{

cout << "File creation unsuccessful";

}

} //end of 'for' loop

savedPlaylistFile.close(); //closing the file

break;

case '2':

//creating and opening a new file from file object

savedPlaylistFile.open("Classical Playlist 2.txt");

filename = "Classical Playlist 2.txt"; //variable to hold name of text file

//loop to go through array & save each element to a file

for (int c = 0; c < 10; c = c + 1) {

if (classicalPlaylistDue[c] != "") //what to do if array location not empty

{

//write the song name to file and then hit carriage return

savedPlaylistFile << classicalPlaylistDue[c] << endl;

fileWritten = true;

} //end of 'if' statement

else

{

cout << "File creation unsuccessful";

}

} //end of 'for' loop

savedPlaylistFile.close(); //closing the file

break;

case '3':

//creating and opening a new file from file object

savedPlaylistFile.open("Popular Playlist 1.txt");

filename = "Popular Playlist 1.txt"; //variable to hold name of text file

//loop to go through array and save each element to a file

for (int d = 0; d < 11; d = d + 1)

{

if (popularPlaylistUno[d] != "") //what to do if array location not empty

{

//write the song name to file and then hit carriage return

savedPlaylistFile << popularPlaylistUno[d] << endl;

fileWritten = true;

} //end of 'if' statement

else

{

cout << "File creation unsuccessful";

}

} //end of 'for' loop

savedPlaylistFile.close(); //closing the file

break;

case '4':

//creating and opening a new file from file object

savedPlaylistFile.open("Popular Playlist 2.txt");

filename = "Popular Playlist 2.txt"; //variable to hold name of text file

//loop to go through array and save each element to a file

for (int e = 0; e < 11; e = e + 1)

{

if (popularPlaylistDue[e] != "") //what to do if array location not empty

{

//write the song name to file and then hit carriage return

savedPlaylistFile << popularPlaylistDue[e] << endl;

fileWritten = true;

} //end of 'if' statement

else

{

cout << "File creation unsuccessful";

}

} //end of 'for' loop

savedPlaylistFile.close(); //closing the file

break;

case '5':

cout << "Returning to main menu\n";

break;

default:

cout << "Incorrect entry\n";

break;

} //end of switch statement

if (fileWritten == true)

{

cout << "File creation is successful!\n";

cout << "File was saved to the disk\n";

cout << "File name is \"" << filename << "\".\n\n";

} //end of 'if' statement

} while (choice != '5'); //end of do; beginning of while

} //end of function