// using arrays to pass multiple items since method arrays can only return 1 item

string[] elementsToCheck = new string[2]; // array of data collected from the interface

bool done = false; // flag to keep us in the while loop until user is done

while (!done)

{

Console.WriteLine();

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Welcome to the Parking Permit Site");

Console.WriteLine("If you would like to issue a parking permit, enter a p");

Console.WriteLine("If you would like to see a report of all the issued permits, enter an r");

Console.WriteLine("If you would like to exit, enter an e");

Console.ResetColor();

string userInput = Console.ReadLine();

userInput.ToLower(); // handle both capital and lower letters

Console.Clear();

switch (userInput)

{

case "p": // add a new entry to the permits "database"

//Put values entered into array

Console.ForegroundColor = ConsoleColor.Red;

Console.Write("please enter your name: ");

elementsToCheck[0] = Console.ReadLine(); // name;

Console.Write("please ener your zip code: ");

elementsToCheck[1] = Console.ReadLine(); // zip;

// off course we could have used a wider array and collected more information

Console.ResetColor();

// call our static Class (you do not instantiate a static Class, as there can only be 1 instance

// You just call its methods directly >>>> ClassName.Method(params);

// this is a call to our middle tier

// by using a param of "out" we are saying (1) varaible by ref, and

// (2) the method should not expect it to have an initial value, it must set if before it uses it.

try

{

// call business middle tier, have it validate, and then record if ok

Console.WriteLine("Your permit is recorded.");

}

catch (ApplicationException ex)

{

Console.WriteLine("Sorry, there was a problem processing your applicaiton");

Console.WriteLine(ex.Message);

Console.WriteLine();

}

break;

// User wants a report

case "r":

Console.WriteLine();

Console.Write("Enter the Admin PW: "); // very serious security here :-)

string PW = Console.ReadLine();

// call static method in the middle tier, which also returns an array, with all the data

string[,] data; // must be defined before we get in the Try, as I want to access out of the Try

try

{

data = null;

// data = middle tier GetPermits(PW);

}

catch (ApplicationException ex)

{

Console.WriteLine(ex.Message);

data = null; // without this, compiler says I am using an unassigned variable below

}

Console.ForegroundColor = ConsoleColor.Magenta;

// write out a header for our table of data

if (data != null)

{

Console.WriteLine("{0, -20}, {1,-7}, {2}", "USERNAME", "ZIP", "DATE ISSUED");

for (int i = 0; i < data.GetLength(0); i++) // loop to write out all the data (including the empty cells)

{

Console.WriteLine("{0, -20}, {1,-7}, {2}", data[i, 0], data[i, 1], data[i, 2]);

// remember how to force column widths?

}

}

Console.ResetColor();

break;

case "q": // accept either q or e for quit or exit

case "e":

Console.WriteLine("Goodbye");

done = true;

break;

default:

Console.WriteLine("Not a valid input.");

break;

} // end of switch

} // end of while

=========================================================================================================================

Create middle tier

Validate method, throw exceptions if bad

//Validate name, at least 4 but not longer than 20 characters

//Validate zip is at least 5 characters

// pass good data to data tier with call to SaveNewPermit(username, zip)

//will return a false if failed to save, then

// throw exception database full or other database issue

public static string[,] GetPermitsMT(string pPW) // if bad PW throw exception

// else call data tier to get data and return it

==============================================================================================================================

Create Data Tier

private static string[,] fakeDB = new string[10, 3];

public static bool SaveNewPermit(string userName, int zip)

{

for (int i = 0; i < fakeDB.GetLength(0); i++)

{

if (fakeDB[i,0] == null)

{

fakeDB[i, 0] = userName;

fakeDB[i, 1] = zip.ToString();

fakeDB[i, 2] = DateTime.Now.ToShortDateString();

return true; // success

}

}

return false; // return false if full

}

public static string[,] GetPermitsDT()

{

return fakeDB;

}