- Vinicius Bruno Gribel

Codigo refatorado:

Main

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Globalization;

using Hotelaria;

namespace Hotel

{

class Program

{

static public int Menu()

{

int opc;

while (true)

{ // para caso ter algum erro ele dar retorno no menu, caso nao tenha ocorre o break dentro do if

try

{

// print menu

Console.WriteLine("[1] - Fazer reserva \n[2] - Alterar datas da Reserva\n[3] - Mostrar Todas Reservas\n[4] - Sair\nR:");

opc = int.Parse(Console.ReadLine());

if (opc >= 1 && opc <= 4)

{

break;

}

else

{

Console.WriteLine("\nNão consegui entender tente novamente por favor");

Console.WriteLine("\n\n");

continue;

}

}

catch(Exception e)

{

Console.WriteLine($"\n{e.Message}");

}

}

return opc;

}

static void Main(string[] args)

{

List<SistemaDeReserva> Reservations = new List<SistemaDeReserva>(); // lista para armazenar os dados

int w1 = 1;

while (w1 ==1)

{ // while para o programa ficar em lup

int opc = Menu(); // instanciando a função menu

Task.Delay(2500);

Console.Clear();

switch (opc)

{

case 1: // fazer reserva

SistemaDeReserva TempReserva = new SistemaDeReserva();

Reservations.Add(TempReserva); // criando a reserva adionando ela em uma lista

Task.Delay(3000);

Console.Clear();

break;

case 2: // alterara reserva

Console.WriteLine("\n Qual o numero do quarto que você quer alterar ?");

int NumberRoom = int.Parse(Console.ReadLine());

for (int NumberReservation = 0; NumberReservation < Reservations.Count; NumberReservation++) // for run reservation in lists

{

if (Reservations[NumberReservation].GetRoom() == NumberRoom)

{

Reservations[NumberReservation] = Reservations[NumberReservation].ChangeReservation(Reservations[NumberReservation]); // if found room change reservation

Console.WriteLine("\nReserva alterada!");

Task.Delay(1500);

}

else if(NumberReservation == Reservations.Count -1){

Console.WriteLine("Não foi possivel encontrar Esse quarto");

Console.WriteLine("\n\n");

break;

}

}

break;

case 3: // listar todas as reservas

foreach(SistemaDeReserva Reservation in Reservations)

{

// print reservas

Console.WriteLine($"\n Room:{Reservation.GetRoom()}\nCheckIn: {Reservation.GetCheckIn().ToString("dd/MM/yyyy")}\nCheckout: {Reservation.GetCheckOut().ToString("dd/MM/yyyy")}\nTotal de dias: {Reservation.GetTotalDays()}");

Console.WriteLine("---------------------");

}

Console.WriteLine("\n\n\n\n");

Task.Delay(5000);

break;

case 4: // encerrar programa

w1 = 0;

Console.WriteLine("Encerrando programa");

break;

}

}

}

}

}

Classe Reserva

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Globalization;

namespace Hotelaria

{

public class SistemaDeReserva

{

private int Room;

private DateTime CheckIn;

private DateTime CheckOut;

private int TotalDays;

public SistemaDeReserva() // "struct" para auxiliar no instnaciamento de objetos

{

while (true) // while para caso ter algum erro ele repetir a tela da entrada de dados

{

try

{

DateTime DataAtual = DateTime.Now; // data atual

int TempQuarto; // Variaveis temporarias para auxiliarem na validação de itens

DateTime TempEntrada; // var temporaria para fazer as validações

DateTime TempSaida; // var temooraria para fazer as validações

Console.WriteLine("\nInsira o numero do quarto: ");

TempQuarto = int.Parse(Console.ReadLine());

Console.WriteLine("\nQual da Data de entrada ? Formato dd/MM/yyyy ");

TempEntrada = DateTime.Parse(Console.ReadLine(), new CultureInfo("pt-BR"));

// convertendo as string em DateTime e informando a CultureInfo

Console.WriteLine("\nQual a data de saida ? Formato dd/MM/yyyy ");

TempSaida = DateTime.Parse(Console.ReadLine(), new CultureInfo("pt-BR"));

if (TempEntrada > DataAtual && TempSaida > TempEntrada) // verificando se a datas são validas

{

SetRoom(TempQuarto);

SetCheckIn(TempEntrada);

SetCheckOut(TempSaida);

SetTotalDay();

break;

}

else

{

// mensagem que os dados inseridos foram invalidos

Console.WriteLine("\n");

Console.WriteLine($"Não foi possivel atualizar a reserva, algum dados está errado.\n LEMBRETE: A data de saida deve ser superior da data de Entrada e a data de entrada deve ser superior a data Atual");

Console.WriteLine("\n\nPrecione qualquer tecla para continuar");

Console.ReadKey(); // aguardar o usuario precionar alguma tecla para dar continuidade

Console.Clear();

continue;

}

}

catch (Exception e)

{

Console.WriteLine($"Erro: {e}");

Console.WriteLine("\n\nPrecione qualquer tecla para continuar");

Console.ReadKey(); // aguardar o usuario precionar alguma tecla para dar continuidade

Console.Clear();

}

}

}

public SistemaDeReserva ChangeReservation(SistemaDeReserva Reservation) // metoso para auxiliar a mudança das datas

{

// var temporarias para auxiliar

DateTime TempEntrada;

DateTime TempSaida;

DateTime DataAtual = DateTime.Now;

Console.WriteLine("Dados atuais da sua reserva:");

Console.WriteLine($"\nCheckIn: {Reservation.GetCheckIn().ToString("dd/MM/yyyy")}\nCheckout: {Reservation.GetCheckOut().ToString("dd/MM/yyyy")}\nTotal de dias: {Reservation.GetTotalDays()}");

Console.WriteLine("---------------------");

while (true) // while para caso ter algum erro ele repetir a tela da entrada de dados

{

try

{

Console.WriteLine("\n Qual da Data de entrada ? Formato dd/MM/yyyy ");

TempEntrada = DateTime.Parse(Console.ReadLine(), new CultureInfo("pt-BR"));

// convertendo as string em DateTime e informando a CultureInfo

Console.WriteLine("\n Qual a data de saida ? Formato dd/MM/yyyy ");

TempSaida = DateTime.Parse(Console.ReadLine(), new CultureInfo("pt-BR"));

if (TempEntrada > DataAtual && TempSaida > TempEntrada)

{

Reservation.SetCheckIn(TempEntrada);

Reservation.SetCheckOut(TempSaida);

Reservation.SetTotalDay();

//chamando a função na classe que muda a instancia

break;

}

else

{

// mensagem de erro

Console.WriteLine("\n");

Console.WriteLine($"Não foi possivel atualizar a reserva, algum dados está errado.\nLEMBRETE: A data de saida deve ser superior da data de Entrada e a data de entrada deve ser superior a data Atual");

Console.WriteLine("\n\nPrecione qualquer tecla para continuar");

Console.ReadKey();

Console.Clear();

continue;

}

}

catch(Exception e)

{

Console.WriteLine($"\n Erro:{e.Message}"); // capturando o erro

}

}

return Reservation; // dando o retorno o objeto

}

// metodos Sets para setar os atributos

public void SetRoom(int NewRoom)

{

this.Room = NewRoom;

}

public void SetCheckIn(DateTime CheckIn)

{

this.CheckIn = CheckIn;

}

public void SetCheckOut(DateTime CheckOut)

{

this.CheckOut = CheckOut;

}

public void SetTotalDay()

{

this.TotalDays = int.Parse(CheckOut.Subtract(CheckIn).ToString("dd")); // conversão do periodo da estadia para string e dps para inteiro

}

//gets para poder visualizar os atributos somente por meio deles

public int GetRoom()

{

return Room;

}

public DateTime GetCheckIn()

{

return CheckIn;

}

public DateTime GetCheckOut()

{

return CheckOut;

}

public int GetTotalDays()

{

return TotalDays;

}

}

}