REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 3 |
| Tema a Evaluar | Pilas estáticas con métodos ingresando números |
| Fecha de Entrega | 29 de Septiembre del 2023 |

**Programa Fuente**

| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace P20\_Pilas\_estaticas\_con\_metodos\_ingresando\_numeros  {  internal class Program  {  public static int top = 0;  static void Main(string[] args)  {  Console.Title = "P20 Sueldos con Pilas Estaticas";  /\*Declaración de variables  Variables para la pila \*/  int[] PilaNumeros = new int[100];  //Variables para captura de datos  int numero = 0;  //Variables para control de menú y opciones  char opcion; bool ejecucion = true;  //Despliegue de menú  do  {  Console.Write("\n\t\t:\*\*: MENÚ DE OPCIONES :\*\*:" +  "\n\t| Las opciones disponibles son las siguientes |" +  "\n\n\t[1] - Insertar Números." +  "\n\t[2] - Mostrar Números." +  "\n\t[3] - Eliminar Números." +  "\n\t[4] - Salir del Programa." +  "\n\n\tIngrese el número de la opción deseada: ");  opcion = char.Parse(Console.ReadLine());  Console.Write($"\n\n\tSe ha seleccionado la opción {opcion}" +  $"\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ;  Console.Clear();  //Control de opciones  switch (opcion)  {  case '1':  {  do  { //Caso 1. Insertar número  Console.Write("\n\t\t:\*\*: INSERTAR NÚMEROS :\*\*:" +  "\n\n\tIngrese el número entero a insertar: ");  //Validación de que el dato sea un número entero  if (int.TryParse(Console.ReadLine(), out numero))  {  Console.WriteLine("\n\n\tSe ha capturado el número correctamente.\n\tAhora será insertado en la pila");  //Llamada al método para insertar  Insertar(PilaNumeros, numero);  Console.Write("\n\n\t¿Desea insertar otro número [ S | N ]? ");  opcion = char.ToUpper(char.Parse(Console.ReadLine()));  }  else  {  Console.WriteLine("\n\n\tEl valor leído no es del tipo númerico esperado.\n\tPor favor asegúrese de escribirlo correctamente.");  opcion = 'S';  }  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  } while (opcion == 'S');  }  break;  case '2':  { //Caso 2. Mostrar sueldos  Console.WriteLine("\n\t\t:\*\*: MOSTRAR NÚMEROS :\*\*:" +  "\n\t| Estos son los números guardados en la pila |");  if (top > 0)  {  for (int i = top - 1; i >= 0; i--)  {  Console.WriteLine("\n\t" + PilaNumeros[i]);  }  }  else Console.WriteLine("\n\tLa pila está vacía");  Console.Write("\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '3':  { //Caso 3. Eliminar sueldo  do  {  Console.Write("\n\t\t:\*\*: ELIMINAR NÚMEROS :\*\*:" +  "\n\n\t¿Desea eliminar el último número insertado [ S | N ]? ");  opcion = char.ToUpper(char.Parse(Console.ReadLine()));  if (opcion.Equals('S')) Eliminar(PilaNumeros); //Llamada a método para eliminar  Console.Write("\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  } while (opcion == 'S');  }  break;  case '4':  { //Caso 4. Salida del programa  Console.Write("\n\t\t:\*\*: SALIDA :\*\*:" +  "\n\n\tGracias por utilizar nuestro programa" +  "\n\n\tPresione la tecla <Esc> para salir...");  while (Console.ReadKey().Key != ConsoleKey.Escape) ;  ejecucion = false;  }  break;  default:  { //En caso de ingresar una opción inválida  Console.Write("\n\t\t.: OPCIÓN INVALIDA :." +  "\n\n\tPor favor ingrese una opción existente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  }  } while (ejecucion);  }  //Definición de procedimiento para insertar  public static void Insertar(int[] PilaNumero, int numero)  {  if (VerificarLLena(PilaNumero.Length) == false)  {  PilaNumero[top] = numero;  top++;  }  }  //Definicion de función para comprobar si está llena la pila  public static bool VerificarLLena(int tamanio)  {  if (top == tamanio - 1) return true;  else return false;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\\  //Definición de método para eliminar  public static void Eliminar(int[] PilaNumero)  {  if (VerificarVacia() == false)  {  top--;  Console.WriteLine("\n\tSe ha eliminado el número " + PilaNumero[top]);  PilaNumero[top] = 0;  }  else Console.WriteLine("\n\tLa pila está vacía." +  "\n\tIngrese \"N\" para regresar al menú");  }  //Definición de función para comprobar si está vacía la pila  public static bool VerificarVacia()  {  if (top <= 0) return true;  else return false;  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

**EXPLICA BREVEMENTE QUÉ TE PARECIÓ EL PROBLEMA, QUE SE TE DIFICULTO Y COMO LO SOLUCIONASTE.**

| La verdad es que no tuve complicaciones con este programa, era bastante similar al anterior, incluso diría que más simple. Me sirvió leer la documentación de microsoft para utilizar el TryParse pero fuera de eso no hubo problemas |
| --- |