

Answer on Question #45804 – Programming – C#

Program.cs

```
using System;
using System.IO;
using System.Text;

namespace task
{
    class Program
    {
        static void Main(string[] args)
        {
            do
            {
                Console.Clear();
                Console.WriteLine("Choose appropriate line (ESC-EXIT):");
                switch (MyPatterns.Singleton.CreateMenu.ShowMenu("Settings", "Show Content",
"Add data"))
                {
                    case 0:

                        Console.WriteLine("Please enter the path to the folder with
Cus_detail.txt:");

                        string path = Console.ReadLine();
                        if (!Directory.Exists(path))
                        {
                            Console.WriteLine("The folder doesn't exist! Press any key to
continue...");

                            Console.ReadKey();
                            continue;
                        }

                        StreamWriter swtr = new StreamWriter(Directory.GetCurrentDirectory() +
@"\conf_file.txt", false, Encoding.GetEncoding(1251));

                        swtr.Write(path + @"\Cus_detail.txt");
                        swtr.Close();

                        if (!File.Exists(path + @"\Cus_detail.txt"))
                        {
                            FileStream fs = new FileStream(path + @"\Cus_detail.txt",
FileMode.Create);

                            fs.Close();
                            // Console.WriteLine(path + @"\Cus_detail.txt" + " was successfully
created");

                            Console.WriteLine("Please enter the path to the folder with
Cus_detail.txt:");

                            path = Console.ReadLine();
                            if (!Directory.Exists(path))
                            {
                                Console.WriteLine("The folder doesn't exist! Press any key to
continue...");

                                Console.ReadKey();
                                continue;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

    }

    Console.ReadKey();
    continue;
}

Console.WriteLine("Saved! Press any key to continue...");
Console.ReadKey();
break;

case 1:

    if (!File.Exists(Directory.GetCurrentDirectory() + @"\conf_file.txt"))
    {
        StreamWriter strw = new
StreamWriter(Directory.GetCurrentDirectory() + @"\conf_file.txt", false);
        strw.Close();
        Console.WriteLine("Configuration file is empty! Press any key...");
        Console.ReadKey();
        continue;
    }

    StreamReader strd = new StreamReader(Directory.GetCurrentDirectory() +
@"\conf_file.txt");

    string file_path = strd.ReadToEnd();
    strd.Close();
    if (File.Exists(file_path))
    {
        strd = new StreamReader(file_path);
        Console.WriteLine("The content from file:");

        Console.WriteLine("////////////////////////////////////////");
        Console.WriteLine(strd.ReadToEnd());

        Console.WriteLine("////////////////////////////////////////");
        strd.Close();
        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
    else
    {
        Console.WriteLine(file_path + "   Doesn't exist. Press any key to
continue...");

        Console.ReadKey();
        continue;
    }
    break;

case 2:
    if (!File.Exists(Directory.GetCurrentDirectory() + @"\conf_file.txt"))
    {
        File.Create(Directory.GetCurrentDirectory() + @"\conf_file.txt");
        Console.WriteLine("Configuration file is empty! Press any key...");
        Console.ReadKey();
        continue;
    }

```

```

StreamReader strd1 = new StreamReader(Directory.GetCurrentDirectory() +
@"\conf_file.txt");

string file_path1 = strd1.ReadToEnd();
strd1.Close();
if (File.Exists(file_path1))
{
    StreamWriter swr = new StreamWriter(file_path1, true,
Encoding.GetEncoding(1251));
    Console.WriteLine("Please enter the data to add:");
    swr.Write(Console.ReadLine());
    swr.Close();
    Console.WriteLine("Successfully saved. Press any key to
continue...");
    Console.ReadKey();
}
else
{
    Console.WriteLine("Firstly Use \"Settings\" menu to fill config
file.");
    Console.ReadKey();
}

break;

default: return;
}

} while (true);
}
}
}

```

Singleton.cs

```
using System;

namespace MyPatterns
{
    class Singleton
    {
        private static Singleton Menu;

        public static Singleton CreateMenu
        {
            get
            {
                if (Menu == null) Menu = new Singleton();
                return Menu;
            }
        }

        private Singleton()
        {
        }

        public int ShowMenu(params string[] lines)
        {
            if (lines.Length < 2) return -1;

            int starPos = 0;

            for (int i = 0; i < lines.Length; i++)
                if (lines[i].Length > starPos) starPos = lines[i].Length;

            starPos += 5;

            int upLinePos = Console.CursorTop + 1;
            int lowLinePos = upLinePos + lines.Length - 1;
            int curLine = upLinePos;
            Console.WriteLine();

            foreach (var line in lines)
            {
                Console.WriteLine(line);
            }

            Console.SetCursorPosition(starPos, upLinePos);
            Console.Write("*");

            ConsoleKeyInfo key;
            do
            {
                key = Console.ReadKey(true);

                switch (key.Key)
                {
                    case ConsoleKey.UpArrow:
                        Console.SetCursorPosition(starPos, curLine);
```

```

        Console.Write(" ");
        if (curLine == upLinePos) curLine = lowLinePos;
        else curLine--;
        Console.SetCursorPosition(starPos, curLine);
        Console.Write("*");
        break;

    case ConsoleKey.DownArrow:
        Console.SetCursorPosition(starPos, curLine);
        Console.Write(" ");
        if (curLine == lowLinePos) curLine = upLinePos;
        else curLine++;
        Console.SetCursorPosition(starPos, curLine);
        Console.Write("*");
        break;

    case ConsoleKey.Escape:
        Console.SetCursorPosition(0, lowLinePos + 2);
        return -1;
    }
} while (key.Key != ConsoleKey.Enter);

Console.SetCursorPosition(0, lowLinePos + 2);
return curLine - upLinePos;
}

}

```