Wiki

Dokumentation • **Database** • **C#** • **Tests** • **Log, Tidsplan** • **Manual**

C#

1. OOA

```
public class Adresse {
  public int Adresseid { get; set; }
  public string Adressestring { get; set; }
  public int Postnr { get; set; }
}
```

```
public class Distrikt {
  public int Distriktid { get; set; }
  public string Omraade { get; set; }
}
```

```
public class Ejer {
  public int Ejerid { get; set; }
  public int Personid { get; set; }
  public string Ejertype { get; set; }
  public string Noter { get; set; }
}
```

```
public class Feriebolig {
  public int Ferieboligid { get; set; }
  public int Distriktid { get; set; }
  public int Adresseid { get; set; }
  public int Ejerid { get; set; }
  public int Opsynsmandid { get; set; }
  public int Stoerrelse { get; set; }
  public int Rum { get; set; }
  public int Senge { get; set; }
  public string Kvalitet { get; set; }
  public double Pris { get; set; }
  public string FeriboligType { get; set; }
  public string Noter { get; set; }
}
```

```
public class Kunde {
  public int Kundeid { get; set; }
  public int Personid { get; set; }
  public string Noter { get; set; }
}
```

```
public class Kundekonsulent {
  public int Kundekonsulentid { get; set; }
  public int Personid { get; set; }
  public string Noter { get; set; }
}


public class Lejekontrakt {
  public int Lejekontrakid { get; set; }
  public int Ferieboligid { get; set; }
  public int Kundeid { get; set; }
  public int Kundekonsulentid { get; set; }
  public int Udlejningskonsulentid { get; set; }
  public DateTime KontraktDato { get; set; }
  public int Aar { get; set; }
  public int Uge { get; set; }
  public double KundePris { get; set; }
  public string UdlejningsKontraktTekst { get; set; }
  public float ElForbrug { get; set; }
}


public class Opsynsmand {
  public int Opsynsmandid { get; set; }
  public int Personid { get; set; }
  public int Distriktid { get; set; }
  public string Noter { get; set; }
}


public class Person {
  public int Personid { get; set; }
  public int Adresseid { get; set; }
  public string Fornavn { get; set; }
  public string Efternavn { get; set; }
  public string Email { get; set; }
  public string Tlf { get; set; }
  public string Password { get; set; }
}


public class PostnrBy {
  public int Postnr { get; set; }
  public string Bynavn { get; set; }
}


public class Saesonkategori {
  public int Ugeid { get; set; }
  public int Kategori { get; set; }
  public string Kategroinavn { get; set; }
  public double Prismodifikator { get; set; }
}


public class Udlejningskonsulent {
  public int Udlejningskonsulentid { get; set; }
  public int Personid { get; set; }
  public int Distriktid { get; set; }
}


public class Udlejningskontrakt {
  public int Udlejningskontraktid { get; set; }
  public int Ferieboligid { get; set; }
  public int Udlejningskonsulentid { get; set; }
  public DateTime KontraktDato { get; set; }
  public int Aar { get; set; }
  public int Uge { get; set; }
  public double PrisEjer { get; set; }
  public string UdlejningsKontraktTekst { get; set; }
}
```

2. OOP

3. Test Console_Screen_max_test2_bufferscrole

```
using System;
using System.Diagnostics;
using System.Runtime.InteropServices;
using System.Text;
using System.IO;

namespace Console_Screen_max_test2_bufferscrole
{
    class Program
    {
        [DllImport("kernel32.dll", ExactSpelling = true)]
        private static extern IntPtr GetConsoleWindow();
        private static IntPtr ThisConsole = GetConsoleWindow();
        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);
        private const int HIDE = 0;
        private const int MAXIMIZE = 3;
        private const int MINIMIZE = 6;
        private const int RESTORE = 9;

        // Scroll vars
        public static int saveBufferWidth;
        public static int saveBufferHeight;
        public static int saveWindowHeight;
        public static int saveWindowWidth;
        public static bool saveCursorVisible;

        static void Main(string[] args)
        {
            Console.SetWindowSize(Console.LargestWindowWidth, Console.LargestWindowHeight);
            ShowWindow(ThisConsole, MAXIMIZE);

            Console.WriteLine("Console_Screen_max_test2\nhttps://www.c-sharpcorner.com/code/448/code-to-

            string m0 = "The current window width is {0}, and \n" +
                "the current window height is {1}.";
            var wzx = Console.WindowWidth;
            var wzy = Console.WindowHeight;
            Console.WriteLine(m0, Console.WindowWidth,
                                  Console.WindowHeight);

            Console.ReadKey(true);
            // Srolle setup

            string m1 = "1) Press the cursor keys to move the console window.\n" +
                "2) Press any key to begin. When you're finished...\n" +
                "3) Press the Escape key to quit.";
            string g1 = "+----";
            string g2 = "|    ";
            string grid1;
            string grid2;
            StringBuilder sbG1 = new StringBuilder();
            StringBuilder sbG2 = new StringBuilder();
            ConsoleKeyInfo cki;
            int y;
            //
            try
            {
                saveBufferWidth = Console.BufferWidth;
                saveBufferHeight = Console.BufferHeight;
                saveWindowHeight = Console.WindowHeight;
                saveWindowWidth = Console.WindowWidth;
                saveCursorVisible = Console.CursorVisible;
                //
                Console.Clear();
                Console.WriteLine(m1);
                Console.ReadKey(true);

                // Set the smallest possible window size before setting the buffer size.
                Console.SetWindowSize(1, 1);
                Console.SetBufferSize((wzx * 2), wzy * 2);
                Console.SetWindowSize(wzx, wzy);

                // Create grid lines to fit the buffer. (The buffer width is 80, but
                // this same technique could be used with an arbitrary buffer width.)
                for (y = 0; y < Console.BufferWidth / g1.Length; y++)
                {
                    sbG1.Append(g1);
                    sbG2.Append(g2);
                }
                sbG1.Append(g1, 0, Console.BufferWidth % g1.Length);
                sbG2.Append(g2, 0, Console.BufferWidth % g2.Length);
                grid1 = sbG1.ToString();
                grid2 = sbG2.ToString();

                Console.CursorVisible = false;
                Console.Clear();
                for (y = 0; y < Console.BufferHeight - 1; y++)
                {
                    if (y % 3 == 0)
                        Console.Write(grid1);
                    else
                        Console.Write(grid2);
                }

                Console.SetWindowPosition(0, 0);
                do
                {
                    cki = Console.ReadKey(true);
                    switch (cki.Key)
                    {
                        case ConsoleKey.LeftArrow:
                            if (Console.WindowLeft > 0)
                                Console.SetWindowPosition(
                                        Console.WindowLeft - 1, Console.WindowTop);
                            break;
                        case ConsoleKey.UpArrow:
                            if (Console.WindowTop > 0)
                                Console.SetWindowPosition(
                                        Console.WindowLeft, Console.WindowTop - 1);
                            break;
                        case ConsoleKey.RightArrow:
                            if (Console.WindowLeft < (Console.BufferWidth - Console.WindowWidth))
                                Console.SetWindowPosition(
                                        Console.WindowLeft + 1, Console.WindowTop);
                            break;
                        case ConsoleKey.DownArrow:
                            if (Console.WindowTop < (Console.BufferHeight - Console.WindowHeight))
                                Console.SetWindowPosition(
                                        Console.WindowLeft, Console.WindowTop + 1);
                            break;
                    }
                }
                while (cki.Key != ConsoleKey.Escape);  // end do-while
            } // end try
            catch (IOException e)
            {
                Console.WriteLine(e.Message);
            }
            finally
            {
                Console.Clear();
                Console.SetWindowSize(1, 1);
                Console.SetBufferSize(saveBufferWidth, saveBufferHeight);
                Console.SetWindowSize(saveWindowWidth, saveWindowHeight);
                Console.CursorVisible = saveCursorVisible;
            }

            // Srolle setup end
        }
    }
}
```

csharp/start.txt · Last modified: 22/03-2020 21:51 by tec