

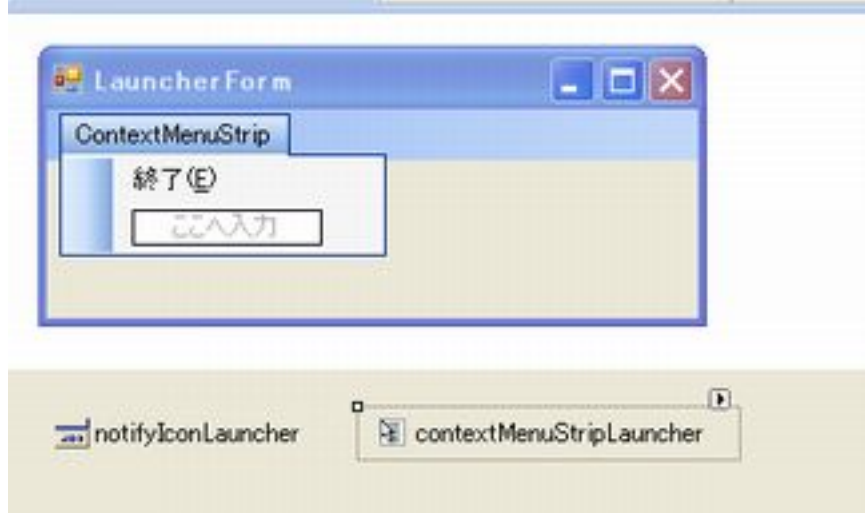
C# バックグラウンドで動く

[C#][Visual Studio][C# タスクトレイ]

起動後、画面を指定時間経過後に非表示にしてタスクトレイに常駐させる

画面のデザイン

- ・ NotifyIcon と ContextMenuStrip を使って、タスクトレイアプリケーションを作成する。



コーディング

Program.cs

- ・ Application.Run に Form ではなく、ApplicationContext のインスタンスを渡すように修正

```
using System;
using System.Windows.Forms;

namespace MyApp
{
    static class Program
    {
        /// <summary>
        /// アプリケーションのメイン エントリ ポイントです。
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            // Form ではなく、ApplicationContext のインスタンスを渡すように修正
            Application.Run(new MyAppContext());
        }
    }
}
```

ApplicationContext を派生させてクラスを作成

- ・ System.Windows.Forms.ApplicationContext

```
using System;
using System.Windows.Forms;

namespace MyApp
```

```

{
    class MyAppContext : ApplicationContext
    {
        public MyAppContext()
        {
            LauncherForm launcherForm = new LauncherForm();
            // 起動する画面の Close イベントをこのクラスの OnLauncherFormClosed でハンドリングする
            launcherForm.FormClosed += new FormClosedEventHandler(this.OnLauncherFormClosed);

            launcherForm.Show();
            // 画面表示させて 1 秒待機
            System.Threading.Thread.Sleep(1000);
            // 画面を非表示に
            launcherForm.Hide();
        }
        private void OnLauncherFormClosed(object sender, EventArgs e)
        {
            // アプリケーションの終了
            ExitThread();
        }
    }
}

```

Form を作成

```

using System;
using System.Windows.Forms;

namespace MyApp
{
    public partial class LauncherForm : Form
    {
        public LauncherForm()
        {
            InitializeComponent();
        }
        private void toolStripMenuItemExit_Click(object sender, EventArgs e)
        {
            // コンテキストメニューの終了メニューで、フォームを閉じる
            this.Close();
        }
    }
}

```