

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Drawing.Imaging;
using System.Net;
using System.IO;
using System.Diagnostics;
using System.Reflection; // Для увеличения NotifyIcon.Text

namespace T_tray
{
    public partial class FormMain : Form
    {
        public static void SetNotifyIconText(NotifyIcon ni, string text) // Для увеличения
        NotifyIcon.Text
        {
            if (text.Length >= 128) throw new ArgumentOutOfRangeException("Text limited to
127 characters");
            Type t = typeof(NotifyIcon);
            BindingFlags hidden = BindingFlags.NonPublic | BindingFlags.Instance;
            t.GetField("text", hidden).SetValue(ni, text);
            if ((bool)t.GetField("added", hidden).GetValue(ni))
                t.GetMethod("UpdateIcon", hidden).Invoke(ni, new object[] { true });
        }

        public String StrMetar = "";
        public String text, Znak, StrNum, StrPlace, StrDateTime, Temp, StrTemp, StrPressure,
WindSpeed, StrWind, StrWeather;
        public int Num;
        public StreamReader SReaderini, SReadertemper;
        public static int Period;
        public static double UTCTimeZone;
        public static String PathINI, PathFile;
        public static FormMain SelfRef //Определяем статическое свойство в классе формы
для вызова из другой формы
        {
            get;
            set;
        }

        public FormMain()
        {

```

```

InitializeComponent();
//NotifyIcon.Text = Str; // задаем всплывающий текст-подсказку иконки в трее
Icon MyIcon = this.Icon;
NotifyIcon.Icon = MyIcon; // устанавливаем значок, отображаемый в трее
NotifyIcon.MouseClick += new MouseEventHandler(NotifyIcon_MouseClick); //
подписываемся на событие клика мышкой по значку в трее
this.Resize += new EventHandler(Form_Resize); // подписываемся на событие
изменения размера формы
NotifyIcon.ContextMenuStrip = ContextMenuStrip; // добавляем контекстное меню
к значку
MenuExit.Click += new EventHandler(MenuExit_Click); // подписываемся на
событие клика по пункту меню "Выход"
this.label2.BackColor = System.Drawing.Color.Transparent;

ToTray();

SelfRef = this; //Для вызова из другой формы
}

//private Icon MyIcon = new Icon("T°tray METAR.ico"); // наша иконка
private FormWindowState OldFormState; // здесь хранится состояние окна до
сворачивания (максимизированное или нормальное)

void MenuExit_Click(object sender, EventArgs e) // обрабатываем клик по кнопке
выхода - завершаем работу приложения
{
    Application.Exit();
}

void NotifyIcon_MouseClick(object sender, MouseEventArgs e) // обрабатываем
событие клика мышью по значку в трее
{
    //проверяем, какой кнопкой было произведено нажатие
    if (e.Button == MouseButtons.Left) // если левой кнопкой мыши
    {
        //проверяем текущее состояние окна
        if (WindowState == FormWindowState.Normal || WindowState ==
FormWindowState.Maximized)//если оно развернуто
        {
            OldFormState = WindowState; // сохраняем текущее состояние
            WindowState = FormWindowState.Minimized; //сворачиваем окно
            // скрываться в трей оно будет по событию Resize (изменение размера),
            // которое сгенерировалось после минимизации строчкой выше
        }
        else // в противном случае
        {
            //Show(); //и показываем на панели задач

```

```
        WindowState = OldFormState; // разворачиваем (возвращаем старое состояние - до сворачивания)
```

```
    }  
}  
}
```

```
void Form_Resize(object sender, EventArgs e) // обрабатываем событие изменения размера
```

```
{  
    if (FormWindowState.Minimized == WindowState) Hide(); //если окно свернуто то скрываем его  
}
```

```
private void MenuSetting_Click(object sender, EventArgs e)
```

```
{  
    Form FormS = new FormSetting();  
    FormS.Show();  
    FormS.ShowInTaskbar = false; //Скрываем вторую форму из панели задач  
    FormS.Left = Screen.PrimaryScreen.WorkingArea.Width - FormS.Width;  
    FormS.Top = Screen.PrimaryScreen.WorkingArea.Height - FormS.Height;
```

```
}
```

```
private void MenuAbout_Click(object sender, EventArgs e)
```

```
{  
    Show();  
    FormMain.ActiveForm.Left = Screen.PrimaryScreen.WorkingArea.Width -  
FormMain.ActiveForm.Width;  
    FormMain.ActiveForm.Top = Screen.PrimaryScreen.WorkingArea.Height -  
FormMain.ActiveForm.Height;  
}
```

```
public void ToTray() // public - обязательно для доступа из другой формы
```

```
{  
    //PathINI = "D:\\- Проекты -\\T°tray METAR\\obj\\Debug\\T°tray METAR.ini"; // Для проекта
```

```
PathINI = "T°tray METAR.ini"; // Для EXE-шника
```

```
try  
{  
    Period = int.Parse(File.ReadLines(PathINI).First()); // читаем 1ую строку из файла
```

```
    UTCTimeZone = Convert.ToDouble(File.ReadLines(PathINI).Skip(1).First()); // читаем 2ую строку из файла - Часовой пояс
```

```
    PathFile = File.ReadLines(PathINI).Skip(2).First(); // читаем 3ую строку из файла - WEB-Путь к файлу с температурой
```

```

WebRequest request = WebRequest.Create(PathFile);
//request.Credentials = new NetworkCredential("admin", "serverpassword");
using (WebResponse response = request.GetResponse())
    using (Stream stream = response.GetResponseStream())
        using (StreamReader reader = new StreamReader(stream))
            StrMetar = reader.ReadToEnd(); // читаем METAR из файла
// --- METAR тестовый ---
//StrMetar = "{\"icao\":\"USNN\", \"name\":\"Нижневартовск. Нижневартовск\", \" +
//\"\"metar\":\"2021-12-15 06:30:00\\nUSNN 150630Z 00000MPS 4900 -SN NSC
M13/M16 Q0030 R21/492032 NOSIG RMK QFE756\", \" +
//\"\"taf\":\"2021-12-15 04:34:00\\nTAF AMD USNN 150434Z 1504/1603
17004G12MPS 3100 -SN BKN017 TEMPO 1504/1512 0700 \" +
//\"\"+SN BKN002 OVC016 BECMG 1512/1513 6000 -SN BECMG 1516/1518
28003G08MPS 9999 NSW\"}";
// --- METAR тестовый с Дальностью видимости ---
//StrMetar = "{\"icao\":\"USNN\", \"name\":\"Нижневартовск. Нижневартовск\", \" +
//\"\"metar\":\"2021-12-21 05:30:00\\nUSNN 210530Z 13006MPS 1800 R03/1800N
R23R/1800D R23L/1400N SHSN BKN029CB M27/M30 Q1023 R03/35//55 TEMPO 1000
SHSN RMK QFE763\", \" +
//\"\"taf\":\"2021-12-21 01:52:00\\nTAF USNN 210152Z 2103/2203 13006G12MPS
6000 -SN BKN016 TEMPO 2103/2115 0800 +SHSN BKN006 BKN015CB\"}";
// --- METAR тестовый с направлением ветра как в TAF ---
//StrMetar = "{\"icao\":\"USNN\", \"name\":\"Нижневартовск. Нижневартовск\", \" +
//\"\"metar\":\"2022-01-17 03:00:00\\nUSNN 170300Z 17008G13MPS 6000 -SN
DRSN OVC022 M04/M05 Q0986 R21/39//36 NOSIG RMK QFE734\", \" +
//\"\"taf\":\"2022-01-17 01:56:00\\nTAF USNN 170156Z 1703/1803 17009G17MPS
3100 -SHSN BLSN BKN010 BKN015CB TEMPO 1703/1712 0800 +SHSN BLSN VV005
BECMG 1705/1707 20011G19MPS BECMG 1721/1722 21009G15MPS 9999 NSW\"}";
// --- METAR тестовый с температурой минус ноль градусов ---
//StrMetar = "{\"icao\":\"USNN\", \"name\":\"Нижневартовск. Нижневартовск\", \" +
//\"\"metar\":\"2022-04-04 04:30:00\\nUSNN 040430Z 15006MPS 9999 DRSN
OVC008 M00/M01 Q1007 R21/450557 NOSIG RMK QFE751\", \" +
//\"\"taf\":\"2022-04-04 02:45:00\\nTAF USNN 040245Z 0403/0503 14007G15MPS
6000 -SNRA OVC007 TEMPO 0403/0409 16010G19MPS 0800 +SHSN +FZRA BKN003
SCT015CB BECMG 0409/0410 FEW010 BKN016 FM041200 18003G08MPS 9999 SCT016
BKN040\"}";
// --- METAR тестовый с температурой " 00/"
//StrMetar = "{\"icao\":\"USNN\", \"name\":\"Нижневартовск.
Нижневартовск\", \"metar\":\"2022-04-04 05:00:00\\nUSNN 040500Z 15007MPS 9999
OVC008 00/M01 Q1007 R21/450557 NOSIG RMK QFE750\", \"taf\":\"2022-04-04
02:45:00\\nTAF USNN 040245Z 0403/0503 14007G15MPS 6000 -SNRA OVC007 TEMPO
0403/0409 16010G19MPS 0800 +SHSN +FZRA BKN003 SCT015CB BECMG 0409/0410
FEW010 BKN016 FM041200 18003G08MPS 9999 SCT016 BKN040\"}";
// --- METAR тестовый с ветром "360"
//StrMetar = "{\"icao\":\"USNN\", \"name\":\"Нижневартовск.
Нижневартовск\", \"metar\":\"2022-07-19 03:30:00\\nUSNN 190330Z 22203MPS CAVOK
20/12 Q1008 R21/0///65 NOSIG RMK QFE751\", \"taf\":\"2022-07-25 01:53:00\\nTAF USNN
250153Z 2503/2603 28003G08MPS 9999 SCT030 BECMG 2506/2508 22003G08MPS\"}";

```

```

// --- METAR тестовый с направлением ветра VRB (переменный ветер)
//StrMetar = "{\"{\"icao\":\"USNN\",\"name\":\"Нижневартовск.
Нижневартовск\",\"metar\":\"2022-08-05 03:30:00\nUSNN 050330Z VRB01MPS 9000
SCT029 19/19 Q1017 R21/0///65 NOSIG RMK QFE758\",\"taf\":\"2022-08-05 01:52:00\nTAF
USNN 050152Z 0503/0603 26003G08MPS 6000 FEW010 BKN016 FM050600
30005G12MPS 9999 BKN016CB TEMPO 0506/0518 VRB20MPS -TSRAGR SQ FEW007
BKN016CB BECMG 0518/0519 32003MPS\"}";
//----- Разбираем строку METAR -----
//--- Город Аэропорт ---
StrPlace = StrMetar.Substring(StrMetar.IndexOf("name")+7,
StrMetar.IndexOf("metar")-StrMetar.IndexOf("name")-10);
StrPlace = StrPlace.Substring(StrPlace.IndexOf(".")+1,
StrPlace.Length-StrPlace.IndexOf(".")-1)
+ "\n Аэропорт \"" + StrPlace.Substring(0, StrPlace.IndexOf(".")) + "\"";
//--- Отделяем только METAR ---
StrMetar = StrMetar.Substring(StrMetar.IndexOf("metar") + 8,
StrMetar.IndexOf("taf") - StrMetar.IndexOf("metar") - 11);
//--- Дата Время ---
StrDateTime = StrMetar.Substring(0, 19);
DateTime DT = new DateTime();
DT = DateTime.Parse(StrDateTime);
DT = DT.AddHours(UTCTimeZone);
StrDateTime = DT.ToString(" dd.MM.yyyy HH:mm");
//--- Ветер ---
if (StrMetar.IndexOf("MPS") != 0) // Метры в секунду
{
    StrWind = StrMetar.Substring(StrMetar.IndexOf("MPS")-9, 8);
    WindSpeed = int.Parse(StrMetar.Substring(StrMetar.IndexOf("MPS")-2,
2)).ToString() + " м/с";
}
else if (StrMetar.IndexOf("KT") != 0) // Узлы
{
    StrWind = StrMetar.Substring(StrMetar.IndexOf("KT")-9, 8);
    WindSpeed = int.Parse(StrMetar.Substring(StrMetar.IndexOf("KT")-2,
2)).ToString() + " уз.";
}
else if (StrMetar.IndexOf("KMH") != 0) // Километры в час
{
    StrWind = StrMetar.Substring(StrMetar.IndexOf("KMH")-9, 8);
    WindSpeed = int.Parse(StrMetar.Substring(StrMetar.IndexOf("KMH") - 2,
2)).ToString() + " км/ч";
}
StrWind = StrWind.Substring(StrWind.IndexOf(" ") + 1, 3);

if (StrWind != "VRB") //переменный ветер
{
    if ((StrWind + WindSpeed).Substring(0, 4) != "0000")
    {

```

```

        int IntWind = int.Parse(StrWind);
        //WindSpeed = int.Parse(WindSpeed).ToString();
        if (IntWind >= 349 || IntWind <= 11) StrWind = "Северный";
        else if (IntWind >= 12 && IntWind <= 33) StrWind = "С-СВ";
        else if (IntWind >= 34 && IntWind <= 56) StrWind = "СВ";
        else if (IntWind >= 57 && IntWind <= 78) StrWind = "В-СВ";
        else if (IntWind >= 79 && IntWind <= 101) StrWind = "Восточный";
        else if (IntWind >= 102 && IntWind <= 123) StrWind = "В-ЮВ";
        else if (IntWind >= 124 && IntWind <= 146) StrWind = "ЮВ";
        else if (IntWind >= 147 && IntWind <= 168) StrWind = "Ю-ЮВ";
        else if (IntWind >= 169 && IntWind <= 191) StrWind = "Южный";
        else if (IntWind >= 192 && IntWind <= 213) StrWind = "Ю-ЮЗ";
        else if (IntWind >= 214 && IntWind <= 236) StrWind = "Ю-З";
        else if (IntWind >= 237 && IntWind <= 258) StrWind = "З-ЮЗ";
        else if (IntWind >= 259 && IntWind <= 281) StrWind = "Западный";
        else if (IntWind >= 282 && IntWind <= 303) StrWind = "З-СЗ";
        else if (IntWind >= 304 && IntWind <= 326) StrWind = "СЗ";
        else if (IntWind >= 327 && IntWind <= 348) StrWind = "С-СЗ";
        StrWind = "Ветер: " + WindSpeed + StrWind;
    }
    else StrWind = "Ветра нет (Штиль)";
}
else StrWind = "Ветер: " + WindSpeed + " перемен-й";
//--- Обрезаем Дальность видимости
//int iDV = StrMetar.IndexOf("/");
while (StrMetar.Substring(StrMetar.IndexOf("/") - 4, 2).IndexOf("R") != -1)
    StrMetar = StrMetar.Substring(StrMetar.IndexOf("/") + 1, StrMetar.Length -
StrMetar.IndexOf("/") - 2);
//--- Температура ---
StrTemp = StrMetar.Substring(StrMetar.IndexOf("/")-3, 3);
if (StrTemp.Substring(1, 2) == "00") StrTemp = "0";
else if (StrTemp.Substring(0, 1) == "M") StrTemp = "-" +
int.Parse(StrTemp.Substring(1, 2)).ToString();
else StrTemp = "+" + int.Parse(StrTemp.Substring(1, 2)).ToString();
Temp = StrTemp;
StrTemp = "Температура: " + StrTemp + "°C";
//--- Давление ---
StrPressure = StrMetar.Substring(StrMetar.IndexOf("/") + 4, 6).Trim(); // Берём с
лишним символом и удаляем все пробелы для учёта с М и без М
if (StrPressure.Substring(0, 1) == "Q") // Давление в гПа
    StrPressure = Math.Round((int.Parse(StrPressure.Substring(1, 4)) *
0.75)).ToString(); // В ммРтСт и округляем до целых
else if (StrPressure.Substring(0, 1) == "A") // Давление в Дюймах рт.ст.
    StrPressure = Math.Round((int.Parse(StrPressure.Substring(1, 4)) *
25.4)).ToString(); // В ммРтСт и округляем до целых
StrPressure = "Давление: " + StrPressure + " мм рт.ст.";
//-----
Timer.Interval = Period * 60000;

```

```

        Bitmap BitMap = new Bitmap(16, 16);
        System.Drawing.Font DrawFont = new System.Drawing.Font("Verdana", 8);
        System.Drawing.SolidBrush DrawBrushRed = new
System.Drawing.SolidBrush(Color.Red);
        System.Drawing.SolidBrush DrawBrushBlue = new
System.Drawing.SolidBrush(Color.Blue);
        System.Drawing.SolidBrush DrawBrushBlack = new
System.Drawing.SolidBrush(Color.Black);
        Pen PenRed = new Pen(Color.Red, 1);
        Pen PenBlue = new Pen(Color.Blue, 1);
        Pen PenBlack = new Pen(Color.Black, 1);
        System.Drawing.Graphics Graph =
System.Drawing.Graphics.FromImage(BitMap);
        Graph.TextRenderingHint =
System.Drawing.Text.TextRenderingHint.SingleBitPerPixel;
        Znak = Temp.Substring(0, 1); // получаем "+" или "-"
        if (Temp == "0") StrNum = Temp;
        else StrNum = int.Parse(Temp.Substring(1, Temp.Length-1)).ToString(); //
выделяем числовую часть
        //--- Задаём всплывающий текст-подсказку иконки в трее ---
        String StrNotify = StrPlace + // Город Аэропорт
            "\n" + StrDateTime + // Дата Время
            "\n" + StrTemp + // Температура
            "\n" + StrPressure + // Давление
            "\n" + StrWind; // Ветер;
        SetNotifyIconText(NotifyIcon, StrNotify);

        Num = int.Parse(StrNum); // переводим текст в число
        if (StrNum == "0")
        {
            Graph.DrawEllipse(PenBlack, 12, 2, 3, 3); // знак градуса
            Graph.DrawString(StrNum, DrawFont, DrawBrushBlack, 4, 5);
        }
        else
        {
            if (Znak == "+") // Знак плюс
            {
                Graph.DrawLine(PenRed, new Point(3, 0), new Point(3, 6)); // знак плюса
                Graph.DrawLine(PenRed, new Point(0, 3), new Point(6, 3)); // знак плюса
            }
            Graph.DrawEllipse(PenRed, 12, 2, 3, 3); // знак градуса
            if (Num < 10) Graph.DrawString(Num.ToString().Substring(0, 1), DrawFont,
            DrawBrushRed, 4, 5); // целое число
            else Graph.DrawString(Num.ToString(), DrawFont, DrawBrushRed, 0, 5); //
число двухзначное целое
        }
    }

```

```

        else if (Znak == "-") // знак минус
        {
            Graph.DrawLine(PenBlue, new Point(0, 4), new Point(6, 4)); // знак плюса
горизонталь
            Graph.DrawEllipse(PenBlue, 12, 2, 3, 3); // знак градуса
            if (Num < 10) Graph.DrawString(Num.ToString().Substring(0, 1), DrawFont,
DrawBrushBlue, 4, 5); // целое число
            else Graph.DrawString(Num.ToString(), DrawFont, DrawBrushBlue, 0, 5); //
число двухзначное целое
        }
    }

```

```

        NotifyIcon.Icon = Icon.FromHandle(Bitmap.GetHicon());
    }
    catch (Exception)
    {
        NotifyIcon.Text = " Температура не получена!\n Ошибка подключения!";
        NotifyIcon.Icon = Icon;
    }
}

```

```

private void MenuRefresh_Click(object sender, EventArgs e)
{
    ToTray();
}

```

```

private void Form1_FormClosing(object sender, FormClosingEventArgs e) // по
закрытию прячем форму
{
    if (this.Visible == true)
    {
        e.Cancel = true;
        this.Hide(); // делаем невидимым форму
    }
}

```

```

private void MenuCopy_Click(object sender, EventArgs e)
{
    Clipboard.SetText(NotifyIcon.Text);
}

```

```

private void MenuExit_Click_1(object sender, EventArgs e)
{
    Close();
}

```



```

private void Imail_MouseEnter(object sender, EventArgs e)
{
    Imail.ForeColor = Color.Blue;
    Imail.Font = new System.Drawing.Font(this.Font.Name, this.Font.Size,
FontStyle.Underline);
}

private void Imail_MouseLeave(object sender, EventArgs e)
{
    Imail.ForeColor = Color.Navy;
    Imail.Font = new System.Drawing.Font(this.Font.Name, this.Font.Size,
FontStyle.Regular);
}

private void Imail_Click(object sender, EventArgs e)
{
    //Process.Start("mailto:digambius@mail.ru&subject=запрос на
программы&body=список программ");
    Process.Start("mailto:digambius@mail.ru");
}

private void Timer_Tick(object sender, EventArgs e)
{
    ToTray();
}

private void LVersion_Click(object sender, EventArgs e)
{
}

}
}

```