

Тајмер

Класа Timer имплементира тајмер (мерач времена) који подиже догађај у ритму кориснички дефинисаних интервала.

Тајмер је оптимизован за употребу у Windows Forms апликацијама и мора се користити као посебан прозор.

Пример: Креирати апликацију која мери време по 5 секунди и по истеку тог интервала пита корисника о изласку из апликације.

```
using System;
using System.Windows.Forms;
namespace ProjekatCS004
{
    public partial class Program : Form
    {
        static Timer tajmer = new System.Windows.Forms.Timer();
        static int brojac = 1;
        static bool izlaz = false;
        private static void TimerEventProcessor(Object objekat,
EventArgs e)
        {
            tajmer.Stop();
            if (MessageBox.Show("Jos?", "Broj ponavljanja do sada: " +
brojac, MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                brojac += 1; //restartovanje tajmera i inkrementiranje
brojaca
                tajmer.Enabled = true;
            }
            else
            {
                izlaz = true;
            }
        }
        public static int Main()
        {
            tajmer.Tick += new EventHandler(TimerEventProcessor);
            tajmer.Interval = 5000; //interval je 5 sekundi
            tajmer.Start();
            while (izlaz == false)
            {
                Application.DoEvents(); //obradjuje dogadjaje po
redosledu
            }
            return 0;
        }
    }
}
```

```
}
```

У примеру се користи тајмер који је дизајниран да ради у оквиру једнонитног окружења где се UI нити користе за извођење обраде.

Овде се користи догађај Tick за операције у коду.

Када год је особеност Enabled постављена на true и када је особеност Interval већа од 0, догађај Tick се подиже над тим интервалима према поставци особености Interval.

Конструктор: Timer() - иницијализује нову инстанцу класе Timer

Особености класе Timer:

- CanRaiseEvents – добија вредност која указује на то да ли компонента може подићи догађај
- Enabled – поставља дозволу за употребу тајмера
- Events – даје листу руковаоца догађајима (event handler) који су закачени на компоненту
- Interval – поставља време у милисекундама пре подизања Tick догађаја релативно у односу на последње подизање Tick догађаја

Методе класе Timer:

- Dispose() – ослобађа све ресурсе које користи компонента
- Equals(Object) – одређује да ли је објекат идентичан са тренутним објектом
- GetType() – добија тип инстанце
- OnTick(EventArgs) – подиже догађај Tick
- Start() – стартује тајмер
- Stop() – зауставља тајмер
- ToString() – враћа стринг који представља Timer

Догађаји класе Timer:

- Disposed – догађа се када је компонента ослобођена позивом Dispose() методе
- Tick – догађа се када се завршио специфицирани временски интервал и тајмер је омогућен

Класа DateTime

Класа DateTimePicker се користи за избор и приказ датума и текућег времена у специфицираном формату на форми.

Класа FlowLayoutPanel се користи за приказ DateTimePicker контроле.

Дефинисана је унутар System.Windows.Forms именског простора.

Може се креирати два различита типа DateTimePicker, попут падајуће листе са датумима представљеним у облику текста или као календар који се појављује када се кликне на стрелицу окренуту надолу на листи.

Особености контроле:

- CalendarFont – поставља стил фонта у календару
- CalendarForeColor – поставља боју фонта у календару
- Format – поставља формат датума и текућег времена приказан на контроли; Custom – уобичајени формат; Long – дуги формат датума који је постављен од стране оперативног система (приказује и датум и текуће време); Short – приказује у скраћеном формату (само датум); Time – приказ времена постављен од стране оперативног система;
- MaxDate – поставља максималну вредност за датум која је дозвољена унутар контроле

- MinDate – поставља минималну вредност за датум која је дозвољена унутар контроле
- ShowUpDown – поставља вредност која указује да ли ће се користити горе-доле дугме за избор датума
- ShowCheckBox – поставља вредност која указује да ли ће се приказати дугме за потврду лево од изабраног датума
- Value – поставља датум/време вредност придружено контроли

Пример: Креирати апликацију која на форми има бирач датума ограничен на неки датум из прошлости до данашњег датума. Избором одговарајућег датума у опсегу, кликом на дугме приказати стринг облик изабраног датума.

```
using System;
using System.Windows.Forms;
namespace ProjekatCS004
{
    public partial class Program : Form
    {
        private DateTimePicker vreme = new DateTimePicker();
        private Button dugme = new Button();
        private Label lejbel = new Label();
        public Program()
        {
            InicijalizacijaForma();
            InicijalizacijaVreme();
            InicijalizacijaLejbel();
            InicijalizacijaDugme();
        }
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Program());
        }
        private void InicijalizacijaForma()
        {
            this.vreme = new System.Windows.Forms.DateTimePicker();
            this.lejbel = new System.Windows.Forms.Label();
            this.dugme = new System.Windows.Forms.Button();
            this.SuspendLayout();
            this.ClientSize = new System.Drawing.Size(600, 600);
            this.Name = "Aplikacija";
            this.Controls.Add(this.vreme);
            this.Controls.Add(this.lejbel);
            this.Controls.Add(this.dugme);
            this.ResumeLayout(false);
            this.PerformLayout();
        }
    }
}
```

```

    }
    private void InicijalizacijaVreme()
    {
        this.vreme.Location = new System.Drawing.Point(20, 20);
        this.vreme.Name = "vreme";
        this.vreme.Size = new System.Drawing.Size(200, 30);
        this.vreme.TabIndex = 0;
        this.vreme.MinDate = new DateTime(1985, 6, 20);
        this.vreme.MaxDate = DateTime.Today;
        this.vreme.CustomFormat = "MMM dd, yyyy - dddd";
        this.vreme.Format = DateTimePickerFormat.Custom;
    }
    private void InicijalizacijaDugme()
    {
        this.dugme.Location = new System.Drawing.Point(50, 150);
        this.dugme.Name = "dugme";
        this.dugme.Size = new System.Drawing.Size(100, 30);
        this.dugme.TabIndex = 1;
        this.dugme.Text = "Potvrda";
        this.dugme.UseVisualStyleBackColor = true;
        this.dugme.Click += new
System.EventHandler(this.DugmeKlik);
    }
    private void InicijalizacijaLejbel()
    {
        this.lejbel.Location = new System.Drawing.Point(50, 200);
        this.lejbel.Name = "lejbel1";
        this.lejbel.Size = new System.Drawing.Size(200, 50);
        this.lejbel.TabIndex = 2;
    }
    private void DugmeKlik(object sender, System.EventArgs e)
    {
        this.lejbel.Text = Convert.ToString(this.vreme.Value);
    }
}
}

```

Задаци за самосталан рад

1. Креирати апликацију која приказује проток времена.
2. Креирати апликацију која рачуна број година између два изабрана датума.