



Díleňská praxe

A4		Samostudium - I		
Petřík Vít			1/4	Známka:
16. 3. 2020		Datum odevzdání:	27. 3. 2020	Odevzdáno:



Zadání:

Vyhledáte na Internetu vhodné studijní podklady pro téma:

- tvorba vlastních událostí

Zpracujete:

- nalezená témata do podoby souhrnné zprávy s uvedením odkazů na zdroje
- navrhnete a vyřešíte příklad, na kterém budete demonstrovat použití (implementaci) vlastních událostí
- posoudíte vhodné oblasti nasazení vlastních událostí

Zpracované téma bude obsahovat dvě části:

- vlastní technická zpráva
- vyřešená úloha

Teorie:

Systém událostí je založen na eventech a delegátech. Tento systém můžeme připodobnit ke vztahu vydavatelství – odběratel.

Vydavatel přijme adresu našich odběratelů. Při vzniku události, v tomto případě vydání nového čísla časopisu, rozešle vydavatel stejný časopis všem odběratelům. V tomto přirovnání má funkci delegáta odběratelova adresa a vydavatel je event.

Delegát:

- Prakticky ukazatel na metodu
- Vytvoří se zapisem vyvolání funkce, akorát vynecháme argument funkce

```
Action delegat = trida.Metoda;
```

- Delegát se pak chová jako běžná funkce a dá se spustit pomocí `delegat()`
- Přidání delegáta k eventu se zapisuje pomocí této konstrukce

```
trida.event += metoda
```

- „+“ se píše, protože zapisujeme do listu. Kdybychom „+“ vynechali, zmizeli by předešlí delegáti

Závěr:

Jako příklad jsem zvolil konzolovou aplikaci, která plní funkci jednoduchého čítače ovládaného pomocí šipek nahoru a dolů. Tento program jsem zvolil, protože podobnou třídu jsem už realizoval v jazyku C++, v úlohách s PWM nebo taktovacím signálem.

Znalost práce s událostmi považuji za velice užitečnou a při realizaci větších projektů je nezbytné vytvářet vlastní událost.



Zdroje:

<https://docs.microsoft.com/en-us/dotnet/standard/events/how-to-raise-and-consume-events?redirectedfrom=MSDN>

<https://stackoverflow.com/questions/803242/understanding-events-and-event-handlers-in-c-sharp>

<https://www.intertech.com/Blog/c-sharp-tutorial-understanding-c-events/>

Výpis programu:

```
using System;

namespace ConsoleApp1
{
    class Program
    {
        static Citac citac;
        static void Main(string[] args)
        {
            //úvodní zprávy
            Console.WriteLine("Zadejte limit čítače: ");

            int cislo = Int32.Parse(Console.ReadLine());
            citac = new Citac(cislo);
            citac.limitDosazen += limitDosazen;

            Console.WriteLine("Stiskněte šipku nahoru pro inkrementaci čítače");
            Console.WriteLine("Stiskněte šipku dolů pro dekrementaci čítače");
            while (true)
            {
                //reakce na stisk kláves
                switch(Console.ReadKey().Key)
                {
                    //šipka nahoru
                    case ConsoleKey.UpArrow:
                        Console.WriteLine("Inkrementace, aktuální stav: " + citac.getState());
                        citac.scitani(1);
                        break;
                    //šipka dolů
                    case ConsoleKey.DownArrow:
                        Console.WriteLine("Dekrementace, aktuální stav: " + citac.getState());
                        citac.scitani(-1);
                        break;
                }
            }
        }

        //event listener
        static void limitDosazen(object sender, argumenty e)
        {
            //výpis na konzoli
            Console.WriteLine("Čítač by přeplněn v hodnotě " + e.hodnota + " v čase " + e.datum);
            Console.WriteLine("program ho vyresetuje");
            //reset citac
            citac.Reset();
        }
    }

    class Citac
    {
        private int limit;
        private int stav;
```



```
//konstruktor třídy
public Citac(int passedLimit)
{
    limit = passedLimit;
}

private void test() {
    if (Math.Abs(stav) >= limit)
    {
        argumenty args = new argumenty();
        args.hodnota = stav;
        args.datum = DateTime.Now;
        //vyvolani eventu
        dosazeni(args);
    }
}

//scitani
public void scitani(int x)
{
    stav += x;
    this.test();
}

//reset
public void Reset()
{
    stav = 0;
}

//vrácení stavu
public int getState() {
    return stav;
}

//vznik eventu
protected virtual void dosazeni(argumenty e)
{
    EventHandler<argumenty> handler = limitDosazen;
    if (handler != null)
    {
        handler(this, e);
    }
}

//definice eventu
public event EventHandler<argumenty> limitDosazen;
}

//třída argumentů
public class argumenty : EventArgs
{
    public int hodnota { get; set; }
    public DateTime datum { get; set; }
}
}
```