

Jazyk C# a platforma .NET

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze

© Pavel Štěpán, Helena Wallenfelsová 2015

Delegáty
BI-DNP



```
// Delegaty - bezpecne pointery na funkce
```

```
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;

namespace Delegaty {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();
        }

        Clovek[] Lide = { new Clovek(5, "Franta"),
                        new Clovek(3, "Alena"),
                        new Clovek(7, "Karel"),
                        new Clovek(2, "Jirka"),
                        new Clovek(6, "Vera"),
                        new Clovek(4, "Eva"),
                        new Clovek(8, "Mirek"),
                        new Clovek(1, "Beta") };

        private void Form1_Load(object sender, EventArgs e) {
            Display();
        }

        void Display() {
            txtData.Clear();
            for (int i = 0; i < Lide.Length; i++) {
                txtData.Text += Lide[i].ID + ";" + Lide[i].Jmeno + "\r\n";
            }
        }

        private void btnSort_Click(object sender, EventArgs e) {
            Porovnani dlgCompare;

            // naplneni delegata (bezpecneho pointeru na funkce)
            if (radID.Checked)
                // ziskani adresy vstupniho bodu funkce
                // testuje, zda funkce odpovida delegatu (bezpecny)!!
                // dlgCompare = new Porovnani(JeVetsiID);
                dlgCompare = JeVetsiID; // totez (zkraceny zapis)
            else
                // dlgCompare = new Porovnani(JeVetsiJmeno);
                dlgCompare = JeVetsiJmeno; // totez
            Sort(Lide, dlgCompare);
            Display();
        }
    }
}
```

```

// porovnavaci funkce - obe stejného typu
bool JeVetsiID(object o1, object o2) {
    return ((Clovek)o1).ID > ((Clovek)o2).ID;
}

bool JeVetsiJmeno(object o1, object o2) {
    // .Compare vraci > 0, pokud je první parametr vetsi
    return String.Compare(((Clovek)o1).Jmeno, ((Clovek)o2).Jmeno) > 0;
}

// parametr typu delegata
void Sort(object[] pole, Porovnani porovnej) {
    bool prehozeno;

    do {
        prehozeno = false;
        for (int i = 0; i < pole.Length - 1; i++) {
            object temp;

            // vyvolani funkce (s prislusnymi parametry),
            // jejiz pointer je ulozen v delegatu 'porovnej'

            // if (porovnej.Invoke(pole[i],pole[i + 1])) {

            // totez zkracene
            if (porovnej(pole[i], pole[i + 1])) {
                temp = pole[i];
                pole[i] = pole[i + 1];
                pole[i + 1] = temp;
                prehozeno = true;
            }
        }
    } while (prehozeno);
}
} // konec class Form1

// definice typu delegat - pointeru na funkce, které mají
// dva parametry typu object a vraceji hodnotu typu bool
delegate bool Porovnani(object p1, object p2);

public class Clovek {
    public int ID;
    public string Jmeno;

    public Clovek(int ID, string Jmeno) {
        this.ID = ID;
        this.Jmeno = Jmeno;
    }
}
}

```