

Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2015

Functory II
BI-DNP



```

using System;
using System.Windows.Forms;

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

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

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

        void Display() {
            txtOutput.Clear();

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

        private void btnSort_Click(object sender, EventArgs e) {
            // misto definice funkci pouzity lambda vyrazy
            // (casto pri pouziti functoru)
            if (radID.Checked)
                Sort(Lide, (p1, p2) =>
                    ((Clovek)p1).ID > ((Clovek)p2).ID);
            else
                Sort(Lide, (p1, p2) =>
                    string.Compare(((Clovek)p1).Jmeno, ((Clovek)p2).Jmeno) > 0);

            Display();
        }
    }
}

```

```

// Definice metody Sort s pomoci functoru - BEZ explicitni
// definice delegata

// Functory jsou vlastne preddefinovane parametricke
// (genericke - ve smyslu generiky) delegaty.
// Maji 0 - 16 parametru, které odpovídají parametrum metod,
// pro kterou vytvori delegáta. Poslední parametr (out),
// odpovídající typu, vracenemu z dane metody

// priklady ruznych funkci o nula, jednom, ... parametru
/*
int VratMaxID()
string VratPosledniJmeno() ...

int Mocnina(int x)
string NaVelka(string Nazev) ...
*/

// definice functoru v .NET:
// functor pro funkci bez parametru:
// delegate TResult Func <out TResult> ();
// functor pro funkci s jednim parametrem:
// delegate TResult Func <in T, out TResult> (T arg);
// functor pro funkci se dvema parametry:
// delegate TResult Func <in T1, in T2, out TResult>
//                               (T1 arg1, T2 arg2);
// ... az do T16

void Sort(object[] Pole,
         Func<object,object,bool> porovnej) {
    bool Prehozeno;

    do {
        Prehozeno = false;

        for (int i = 0; i < Pole.Length - 1; i++) {
            if (porovnej(Pole[i],Pole[i + 1])) {
                object Temp;
                Temp = Pole[i];
                Pole[i] = Pole[i + 1];
                Pole[i + 1] = Temp;
                Prehozeno = true;
            }
        }
    } while (Prehozeno);
}
} // konec class Form1

```

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

    public Clovek(int ID, string Jmeno)
    {
        this.ID = ID;
        this.Jmeno = Jmeno;
    }
} // konec class Clovek
} // konec namespace FunctoryII
```