Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2014

Indexery
BI-DNP

```
// Definice tridy s indexerem umoznuje pouzivat jeji instance jako pole

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;
// pridano!!
using System.Collections;

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

        // pouziti indexeru
        private void btnZobraz_Click(object sender, EventArgs e)
        {
            // objLide je JEDEN objekt, se kterym lze zachazet jako s polem!!
            Lide objLide = new Lide();

            Clovek objClovek = new Clovek(3, "Anezka", "Prekrasna");

            objLide[2] = objClovek;   // prace s indexerem (index) - prirazeni

            txtVystup.Text = objLide[2].Krestni;  // nacteni

            txtVystup.Text = objLide["Vopicka"].Krestni; // textovy "index"

            txtLide.Clear();

            // foreach vyzaduje, aby objLide implementoval interface IEnumerable
            foreach (Clovek c in objLide){
               txtLide.Text += c.ID + "; " + c.Krestni + "; " +
                               c.Prijmeni + "\r\n";
            }
        }
    } // konec class Form1
```

```
// pomocna trida
public class Clovek
{
    public int ID;
    public string Krestni;
    public string Prijmeni;

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

// trida s indexerem
// interface IEnumerable<Clovek> kvuli foreach
public class Lide : IEnumerable<Clovek>
{
    // pole "natvrdo" ve tride - jen pro jednoduchost
    private Clovek[] poleLidi = {
                new Clovek(1,"Franta","Novak"),
                new Clovek(2,"Venca","Vopicka"),
                new Clovek(3,"Eulalie","Prekrasna"),
                new Clovek(4,"Tonda","Cerny")
    };

    // pomocna funkce
    int NajdiPrijmeni(string Prijmeni){

        int index = -1;
        for (int i = 0; i < poleLidi.Length; i++){
            if (poleLidi[i].Prijmeni == Prijmeni){
                index = i;
                break;
            }
        }
        return index;
     }

    // definice vlastnosti, ktera umoznuje pracovat s objektem
    // jako s polem (indexer) (neni osetren pripad chybneho indexu)

    public Clovek this[int index]
    {
        get{
            return poleLidi[index];
        }
        set {
            poleLidi[index] = value;
        }
```

```
        }

        // druha definice - s "indexem" typu string
        // (neni osetren pripad, kdy se jmeno nenajde)

        public Clovek this[string Name]
        {
            get{
                return poleLidi[NajdiPrijmeni(Name)];
            }
            set{
                poleLidi[NajdiPrijmeni(Name)] = value;
            }
        }

        // implementace interface IEnumerable pomoci yield
        IEnumerator<Clovek> IEnumerable<Clovek>.GetEnumerator()
        {
            for (int i = 0; i < poleLidi.Length; i++)
                yield return poleLidi[i];
        }

        IEnumerator IEnumerable.GetEnumerator()
        {
            return ((IEnumerable<Clovek>)this).GetEnumerator();
        }

        /*
        // pro ne-generické kolekce staci toto:
        IEnumerator IEnumerable.GetEnumerator()
        {
            for (int i = 0; i < poleLidi.Length; i++)
                yield return poleLidi[i];
        }
        */
    }
}


/*
      // "Manualni" implementace interface IEnumerable
      // (pro jednoduchost negenericke)

      // Trida musi implementovat IEnumerable, aby bylo mozno pouzivat foreach
      // IEnumerable - jedina metoda GetEnumerator, vracejici objekt, ktery
      // implementuje interface IEnumerator

      public class Lide : IEnumerable{
           // vnorena trida, implementujici IEnumerator
           public class LideEnumerator : IEnumerator{
                Lide ptrLide;
                int eIndex;
```

```
            public LideEnumerator(Lide ptrLide){
                  this.ptrLide = ptrLide;
                  eIndex = -1;
            }

            public bool MoveNext(){ // metoda IEnumerator
                  eIndex++;
                  if (eIndex >= ptrLide.poleLidi.Length){
                        eIndex--;
                        return false;
                  }
                  else{
                        return true;
                  }
            }

            public void Reset(){ // metoda IEnumerator
                  eIndex = -1;
            }

            // metoda IEnumerator (typ MUSI byt objekt pro ne-generiky!!!)
            public object Current{
                  get {  return ptrLide.poleLidi[eIndex]; }
            }
      } // konec class LideEnumerator

      public IEnumerator GetEnumerator(){ // jedina metoda IEnumerable
            return (IEnumerator) new LideEnumerator(this);
      }
} // konec class Lide
*/
```