

Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2015

Interfaces (rozhraní)
BI-DNP



```

// Definice a pouziti interfaces
using System;
using System.Windows.Forms;

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

        private void btnProved_Click(object sender, EventArgs e) {
            Konto objKonto = new Konto();

            objKonto.Vlozit(10000);
            objKonto.Save("C:\\Konta\\MojeKonto.dat");
            // objKonto.Encode(12345678); // nelze - implicit private!!

            ISecurity ptrISec; // pointer na Interface
            // do pointeru na interface lze vlozit pointer na instanci tridy,
            // ktera tento interface implementuje!!

            ptrISec = objKonto;
            // takto zprístupnim metody explicit implementující interface:
            ptrISec.Encode(12345678);
            // totez kompaktněji:
            ((ISecurity)objKonto).Encode(12345678);
        }

        // ukazka pouziti interface IDisposable (definovaného v .NET)
        void UkazkaPozitiUsing()
        {
            // pri opusteni bloku using se AUTOMATICKY vyvola metoda Dispose
            // instance objKonto!!
            // Pro pouziti bloku using MUSI trida Konto implementovat
            // interface IDisposable!!!!
            using (Konto objKonto = new Konto()){
                objKonto.Vlozit(1000);
                objKonto.Save("C:\\Konta\\MojeKonto.dat");
            }
        }

        // doporučený přístup - používat interfaces místo tříd. Tedy místo ...
        void UkazkaClassParametru(Konto objKonto){
            // ...
            objKonto.Save("C:\\Konta\\MyKonto.dat"); // ...
        }

        void UkazkaIfcParametru(IDiskoveOperace ptrIDisk){// je obecnější toto
            // ...
            ptrIDisk.Save("C:\\Konta\\MojeKonto.dat"); // ...
        }
    } // konec class Form1

```

```

// definice interface
// v interface je mozne definovat i properties
public interface IDiskoveOperace {
    void Save(string Path); // NESMI mit modifikatory public, private, ...
    void Load(string Path);
}

// definice dalšího interface
public interface ISecurity {
    void EnCode(long key);
    void DeCode(long key);
}

// trida muze dedit z jedine tridy, ale muze implementovat
// libovolny pocet interfaces (soucasne mozno dedit i implementovat)

public class Konto : IDiskoveOperace, ISecurity, IDisposable {
    // ... : IDisk.. -
    // implementuje dany interface

    // bezne metody tridy
    public void Vlozit(decimal Castka) {
        // ...
    }

    public void Vybrat(decimal Castka) {
        // ...
    }

    // metody pro implementaci interface IDiskoveOperace
    public void Save(string Path) {
        // ...
    }

    public void Load(string Path) {
        // ...
    }

    // metody pro explicitni implementaci interface ISecurity
    void ISecurity.EnCode(long key) { // nesmi byt public, private, ...
        // ...
    }

    void ISecurity.DeCode(long key) {
        // ...
    }

    // implementace interface IDisposable (jedina metoda Dispose)
    public void Dispose() {
        // ...
    }
}
}

```