

Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2014

Dědičnost a polymorfismus
BI-DNP



```

// Dedicnost a polymorfismus v C#
// (dedena trida i její rodice jsou v samostatném souboru)
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 Dedicnost {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();

            Zam objZam1, objZam2;
            // Sef objSef;

            private void btnNewZam1_Click(object sender, EventArgs e) {
                objZam1 = CreateZam();

                // objSef = new Sef("Boss"); // instance dedene tridy
                // objSef.SetCase(); // volani dedene metody
            }

            private void btnNewZam2_Click(object sender, EventArgs e) {
                objZam2 = CreateZam();
            }

            private void btnShowZam1_Click(object sender, EventArgs e) {
                ShowZam(objZam1);
            }

            private void btnShowZam2_Click(object sender, EventArgs e) {
                ShowZam(objZam2);
            }

            Zam CreateZam() {
                Zam objZam;

                if (chkSef.Checked)
                    // do promenne typu pointer na RODICE MOHU vložit pointer
                    // na DEDICE!!
                    objZam = new Sef(txtJmeno.Text, int.Parse(txtLidi.Text));
                else
                    objZam = new Zam(txtJmeno.Text);

                chkSef.Checked = false;
                txtLidi.Clear();
                txtID.Clear();
            }
        }
    }
}

```

```

        txtJmeno.Clear();
        txtJmeno.Focus();
        return objZam;
    }

    void ShowZam(Zam objZam) {
        chkSef.Checked = false;
        txtLidi.Clear();

        txtID.Text = objZam.ID.ToString();
        txtJmeno.Text = objZam.Jmeno;
        /* 1. varianta
        if (objZam is Sef) {    // obsahuje objZam pointer na instanci
                               // tridy Sef?
            chkSef.Checked = true;
            // pretypovani NUTNE!!
            txtLidi.Text = ((Sef)objZam).Lidi.ToString();
        }
        */
        // 2. varianta
        Sef objSef;
        objSef = objZam as Sef; // je-li v objZam pointer na instanci
                               // tridy Sef, priradi; pokud ne,
                               // priradi null

        if (objSef != null) {    // jednoduchy test
            chkSef.Checked = true;
            txtLidi.Text = objSef.Lidi.ToString();
        }
    }

    private void btnUpper1_Click(object sender, EventArgs e) {
        objZam1.SetCase();
        ShowZam(objZam1);
    }

    private void btnLower1_Click(object sender, EventArgs e) {
        objZam1.SetCase(false);
        ShowZam(objZam1);
    }

    private void btnProper1_Click(object sender, EventArgs e) {
        objZam1.ToProper(); // vola metodu POLYMORFNE - dle typu objZam1!!
        ShowZam(objZam1);
    }

    private void btnProper2_Click(object sender, EventArgs e) {
        objZam2.ToProper();
        ShowZam(objZam2);
    }
}

```