

Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2014

Definice děděné třídy BI-DNP



```
// Definice tridy a jejího dedice
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Dedicnost {
    class Zam {
        private static int LastID = 0;

        private int locID;
        protected string locJmeno;
        // modifikatory pristupu (access modifiers)
        // - public - pristup maji vsichni
        // - private - jen ja
        // - protected - ja a dedici
        // - internal - ja a vsichni ve stejne assembly
        // - protected internal - ja a vsichni ve stejne assembly
        // + dedicove kdekoli

        public string Jmeno {
            get {
                // ...
                return locJmeno;
            }

            set {
                // ...
                locJmeno = value;
            }
        }

        public int ID {
            get {
                // ...
                return locID;
            }
        }

        public Zam(string Jmeno) {
            this.locID = ++LastID;
            this.locJmeno = Jmeno;
        }

        public void SetCase(bool Upper) {
            if (Upper)
                locJmeno = locJmeno.ToUpper();
            else
                locJmeno = locJmeno.ToLower();
        }
    }
}
```

```

    public void SetCase() {
        SetCase(true);
    }

    // virtual - povolují POLYMORFNI redefinici této metody v dedici
    public virtual void ToProper() {
        locJmeno = locJmeno.Substring(0, 1).ToUpper() +
            locJmeno.Substring(1).ToLower();
    }
} // konec class Zam

class Sef : Zam { // ... : Zam - dedi ze tridy Zam;
    // dedeni pouze z JEDNE tridy

    // nema-li rodicovska trida bezparametricky konstruktor, MUSI
    // definovat svůj konstruktor dedena trida

    public int Lidi;

    // ... : base(...) volani konstrukturu rodice - NUTNE!!
    public Sef(string Jmeno, int Lidi) : base(Jmeno) {
        this.Lidi = Lidi;
    }

    // override - POLYMORFNE redefinují stejnojmennou metodu rodice
    public override void ToProper() {
        locJmeno = locJmeno.ToUpper();
    }

    // new (nebo NIC!!) - NEpolymorfne redefinují stejnojmennou metodu
    // rodice
    /*
    public new void ToProper() { // vola se metoda dle typu promenne
        // v dobe kompilace, NIKOLI dle
        // obsahu pri behu programu!!

        locJmeno = locJmeno.ToUpper();
    }
    */
}
}

```