Jazyk C# a platforma .NET

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

Definice tridy Zam
BI-DNP

```
// Deklarace tridy, pouzivane k vykladu zakladni prace s objekty

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

namespace Objekty {
    class Zam {
        // staticke promenne, metody a konstruktor
        // public static int LastID = 0;   // definice staticke promenne
        private static int LastID = 0;

        public static int GetLastID() {     // definice staticke metody
            return LastID;
        }

        static Zam() { // definice statickeho konstruktoru (volan automaticky)
            LastID = 24;    // simulace nacteni z databaze, ze souboru, ...
        }

        // staticky konstruktor a staticke metody smeji pouzivat
        // POUZE staticke promenne!!

        // instancni metody, vlastnosti a konstruktory
        // vlastnosti 1. varianta (public promenne)
        // public int ID;
        // public string Jmeno;

        // vlastnosti 2. varianta (property procedury)
        private int priID;
        private string priJmeno;

        public string Jmeno {
            get {
                // ... // zde mozno dekomprimovat, formatovat, ...
                return priJmeno;
            }

            set {
                // ... // zde mozno OTESTOVAT, komprimovat, upravit, ...
                priJmeno = value;   // value - prirazovana hodnota
            }
        }

        // read only property (chybi set); lze i write only property - bez get
        public int ID {
            get {
                // ...
                return priID;
            }
        }
```

```
// C# umoznuje i "automatickou" implementace properties:
// Mozno pouzit napr. v situaci, kdy je nutno realizovat vlastnosti
// pomoci properties, ale zatim nejsou znamy detaily jejich chovani.

// public int ID { private get; set; } // private - read only property
// public string Jmeno { get; set; }


// trida bez vlastni definice konstruktoru dostane automaticky
// pridelen implicitni bezparametricky konstruktor

// definice konstruktoru - 1. varianta
/*
public Zam(int ID, string Jmeno) {
    this.ID = ID;   // this - pointer na tu instanci teto tridy,
                    // ve ktere se prave tento kod vykonava
    this.Jmeno = Jmeno;
}
*/

// pretezovani konstruktoru
/*
public Zam() {
    this.ID = 99;
    this.Jmeno = "Franta";
}
*/

// pretezovani konstruktoru s volanim druheho konstruktoru
/*
public Zam() : this(99, "Franta") {
    // ...
}
*/

readonly int Cislo = 2; // read only promenna - lze nastavit POUZE
                        // v definici a v konstruktoru

// konstruktor, ktery vyuziva statickou promennou
public Zam(string Jmeno) {
    this.priID = ++LastID;
    this.priJmeno = Jmeno;

    Cislo = 5;    // zde MOHU priradit do readonly promenne
}
```

```
        // definice metod
        public void SetCase(bool Upper) {
            if (Upper)
                priJmeno = priJmeno.ToUpper();
            else
                priJmeno = priJmeno.ToLower();

            // Cislo = 7; // nelze - read only promenne lze nastavovat POUZE
                          // v definici a konstruktoru
        }

        // pretizeni metody (overloading) - stejne jmeno, lisi se POCET
        // nebo TYP parametru
        public void SetCase() {
            // Jmeno = Jmeno.ToUpper();
            SetCase(true);  // realizace metody volanim druhe pretizene
                            // verze této metody s konkretnimi parametry
                            // (casto pouzivano)
        }

        public void Dispose() {
            // zde mozno uklidit: zavrit connection k databazi, zavrit
            // soubory, ...

            GC.SuppressFinalize(this);  // nevolat finalizer
        }

        // finalizer ("destruktor") - volan (automaticky) pred
        // zrusenim objektu
        ~Zam() {
            // zde mozno uklidit: zavrit connection k databazi, zavrit
            // soubory, ...
            // pokud existuje Dispose, mozno pro uklidit jeho volanim

            // NIKDY nepouzivat MessageBox ve finalizeru!!
            // (zde jen pro testovani)
            System.Windows.Forms.MessageBox.Show("Koncim - " + priJmeno);
        }
    }
}
```