

# Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2014

**Abstraktní třída**  
BI-DNP



```
// Definice a pouziti abstraktni tridy
```

```
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 AbstractMath {  
    public partial class Form1 : Form {  
        public Form1() {  
            InitializeComponent();  
        }  
  
        MyMath ptrMyMath; // pointer na "instance" abstraktni tridy  
        Vektor ptrVektor; // pointer na instance "realne" tridy (dedice)  
        Cislo ptrCislo;   // pointer na instance "realne" tridy (dedice)  
  
        private void Form1_Load(object sender, EventArgs e) {  
            // ptrMyMath = new MyMath(); // NELZE instancovat abstraktni tridu  
            ptrCislo = new Cislo(-4);  
            ptrVektor = new Vektor(-3, 4);  
            // do pointeru na "instance" abstraktni tridy  
            // LZE priradit pointery na jeji dedice  
            ptrMyMath = ptrCislo;  
        }  
  
        private void radCislo_CheckedChanged(object sender, EventArgs e) {  
            if (radCislo.Checked)  
                ptrMyMath = ptrCislo;  
            else  
                ptrMyMath = ptrVektor;  
        }  
        // odtud program NEZAVISI na konkretnich tridach (Cislo, Vektor)  
        // ale pouziva POUZE abstraktni tridu MyMath  
        private void btnInvert_Click(object sender, EventArgs e) {  
            ptrMyMath.Invert();  
            txtVystup.Text = ptrMyMath.ToString();  
        }  
  
        private void btnAbsVal_Click(object sender, EventArgs e) {  
            txtVystup.Text = ptrMyMath.AbsVal().ToString();  
        }  
    } // konec class Form1
```

```

// definice abstraktni tridy
public abstract class MyMath {
    public abstract void Invert();        // abstraktni metoda NEMA telo!!
    public abstract double AbsVal();      // abstraktni metody jsou
                                           // automaticky virtual!!
    // abstraktni trida MUZE obsahovat "konkretni" metody
} // konec class MyMath

// trida, dedena z abstraktni tridy
// (pokud neprekryje vsechny abstraktni metody, musi byt take abstract!!)
public class Cislo : MyMath {
    private double Hodnota;
    public Cislo(double Hodnota) {
        this.Hodnota = Hodnota;
    }

    public override void Invert() {
        Hodnota = -Hodnota;
    }

    public override double AbsVal() {
        return Math.Abs(Hodnota);
    }

    public override string ToString() {
        return Hodnota.ToString();
    }
} // konec class Cislo

// dalsi trida, dedena z abstraktni tridy
public class Vektor : MyMath {
    double X, Y;

    public Vektor(double X, double Y) {
        this.X = X;
        this.Y = Y;
    }

    public override void Invert() {
        X = -X;
        Y = -Y;
    }

    public override double AbsVal() {
        return Math.Sqrt(X * X + Y * Y);
    }

    public override string ToString() {
        return "(" + X + ";" + Y + ")";
    }
}
}

```