

Jazyk C# a platforma .NET

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

© Pavel Štěpán, Helena Wallenfelsová 2015

Anonymní a lambda funkce (výrazy)
BI-DNP



```

using System;
using System.Windows.Forms;

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

        // definice delegata (pro ukazku prirazeni anonymnich a
        // lambda funkci (vyrazu))
        delegate void ZobrazDelegate(string Msg);

        private void btnProved_Click(object sender, EventArgs e) {
            ZobrazDelegate zStd, zAnonym, zLambda;

            // standardni prirazeni delegata
            zStd = new ZobrazDelegate(Zobraz);
            zStd = Zobraz; // druha syntaxe
            zStd("Standardni\r\n"); // volani

            // anonymni funkce (metoda); vraceny typ se nedeklaruje
            // pripadna vracena hodnota standardne pomoci return
            // definice anonymni funkce:
            zAnonym = delegate(string s) { txtOutput.Text += s; };
            zAnonym("Anonymni\r\n"); // volani
            // anonymni funkce lze pouzit pro definovani eventu nebo
            // jako parametru funkce atd. tam, kde se ocekava delegat

            // lambda vyraz (lambda funkce)
            // (c) - parametry funkce (anonymni)
            // operator => na pravo od nej je telo funkce
            zLambda = (c) => { txtOutput.Text += c; }; // definice
            zLambda("Lambda\r\n"); // volani

            // dalsi informace k lambda vyrazum:
            // x => x*x // u jednoho parametru zavorky nepovinne
            // // jde-li o vyraz pro return (vracena
            // // hodnota), nemusime pouzivat { }
            // (x,y) => x == y // 2 a vice parametru - zavorky NUTNE
            // () => DateTime.Now // NUTNE i pro funkci bez parametru

            // vystupni typ se snazi kompilator odvodit
            // (Type-inference)
            // lze definovat i typy parametru:
            // (int x, string popis) => popis + x

            // lambda fce lze opet pouzit tam, kde se ocekava delegat
        }
    }
}

```

```
        // explicite definovana funkce pro "standardni" prirazeni
        void Zobraz(string Zprava) {
            txtOutput.Text += Zprava;
        }
    }
}
```