



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Výpočetní technika programování

## Vlastnosti OOP programů

Tento materiál vznikl v rámci projektu CZ.1.07/1.2.15/01.0022  
„Záchytná síť– široké spektrum podpůrných aktivit“

Střední škola spojů a informatiky, Tábor, Bydlišského 2474, 390 11 Tábor  
[www.sous.cz](http://www.sous.cz)

© 2014 Ing. Almásiová Dana

# VLASTNOSTI OBJEKTIVĚ ORIENTOVANÝCH PROGRAMŮ

## Vlastnosti OOP

Objektový přístup umožňuje v programech modelovat skutečný svět. Návrh programu je vlastně návrhem tříd pro řešení úlohy (objekty jsou instance třídy). Jednou z vlastností OOP je **zapouzdření**, což je seskupení logicky souvisejících proměnných do jednoho celku - tedy objektu (kde místo např. 3 proměnných – typ auta, obsah motoru, najeto km, vytvoříme třídu Auto s těmito vlastnostmi).

Další vlastností OOP je **dědičnost**, když vytvoříme třídu potomka ze třídy rodičovské, bude mít potomek vlastnosti rodičovské třídy (zdědí je) a navíc svoje další. Takže když například vytvoříme třídu Nákladní auto ze třídy Auto, bude mít její vlastnosti a třeba jednu navíc – nosnost). Pokud program pracuje s větším počtem objektů dané třídy, je jejich úložištěm obvykle indexovaný seznam(<List>).

## Knihovny DLL

Knihovna je soubor s podprogramy nebo třídami. Používají se pro uložení opakovaně použitelných podprogramů a tříd. Vytvářejí se ze šablony „Class Library“ s modifikátorem public.

Hotová knihovna se k projektu připojuje přes Project-> Add reference a ve zdrojovém kódu řádkem „using“. Dnes se nejčastěji připojuje dynamicky, tedy kód knihovny se nezapisuje do .exe souborů, ale aktivuje se buď při spuštění aplikace nebo při aktivaci požadavku na její podprogram.

## PROGRAMOVÁNÍ VÍCE OKEN

Modularizace programu znamená rozdělení programu na jednotlivé podúlohy - samostatné moduly. Další okna můžeme potřebovat, když je aplikace rozsáhlá, ale lze ji rozdělit na podúlohy a pro ně připravit vlastní formuláře, které spustíme z úvodního formuláře aplikace. Druhé okno přidáme přes volbu Project-> Add Windows Form a zadáme mu jméno (např. *Druhy*).

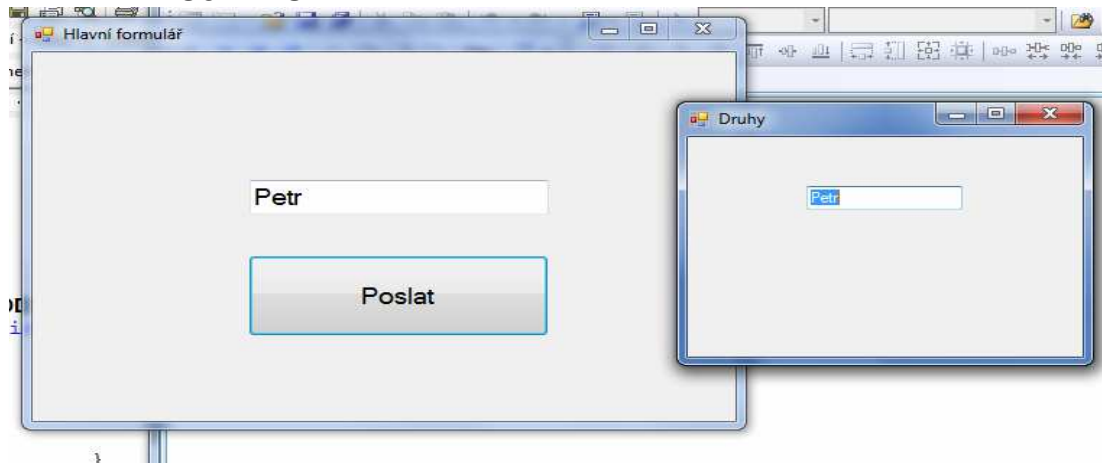
Chceme-li vytvořit proměnnou, s kterou můžeme pracovat ve více oknech, nastavíme ji vlastnost Modifiers na Public (proměnná je otevřená do světa a příchod k ní z venku). Nejjednodušší je textbox nebo proměnnou, s kterou pracujeme v prvním formuláři, přiřadit přímo do druhého formuláře, ve kterém chceme s proměnnou dále pracovat a můžeme si její změny, které jsme provedli v prvním formuláři, zobrazit v textboxu druhého formuláře.

## ŘEŠENÝ PŘÍKLAD

### ZADÁNÍ:

Zadejte vaše jméno v úvodním formuláři. Po stisku tlačítka předejte zadané jméno do druhého formuláře.

## NÁVRH PROJEKTU:



## KÓD ŘEŠENÍ:

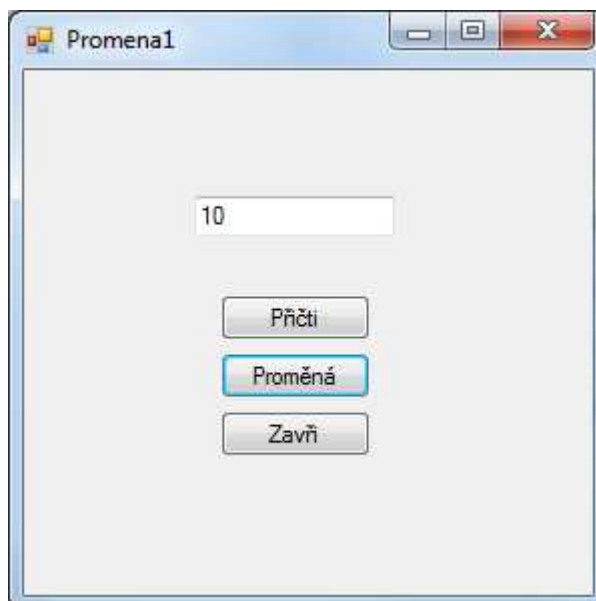
```
private void button1_Click(object sender, EventArgs e)
{
    //aktivace druhého formuláře
    Druhy form = new Druhy();
    //předání hodnoty z prvního formuláře do textboxu na druhém
    form.poleDruhy.Text = polePrvni.Text;
    //zobrazení druhého formuláře
    form.Show();
}
```

## ŘEŠENÝ PŘÍKLAD

### ZADÁNÍ:

Vytvořte aplikaci s dvěma formuláři. Spočítejte součet zadaných hodnot v prvním formuláři a výsledek zobrazte ve druhém. Třetí tlačítko bude ukončovat aplikaci.

## NÁVRH PROJEKTU:



## KÓD ŘEŠENÍ:

```
private void Promena_Click(object sender, EventArgs e)
{
    Promena2 form = new Promena2();
    form.textBox1.Text = pocet.ToString(textBox1.Text);
    form.Show();
}
private void Pricti_Click(object sender, EventArgs e)
{
    pocet += Convert.ToInt32(textBox1.Text);
    textBox1.Text = null;
}
private void Zavri_Click(object sender, EventArgs e)
{
    Close();
}
```



Pokud po zapsání hodnoty do textboxu nestiskneme tlačítko Přičti, kterým bychom hodnotu (10) připočetli do proměnné *pocet*, zobrazí se nám to, co je napsáno v textboxu prvního formuláře bez použití proměnných.



Když zadáme jakékoliv číselné hodnoty do prvního formuláře a každou z nich potvrdíme tlačítkem Přičti, které zapisuje hodnoty do proměnné *pocet*, tak v druhém formuláři se nám zobrazí změněná proměnná *pocet*, se kterou můžeme dále pracovat ve druhém formuláři (tento proces může být i zpětný).

## TYPOVÉ PŘÍKLADY:

1. Zadejte v úvodním formuláři vaše jméno, v druhém vytvořte uvítací okno s vaším jménem.
2. Rozšiřte první úkol o zadání hesla, podle hesla upravte viditelnost (vlastnost *visible*) nabídek v druhém formuláři – pro běžného uživatele a pro správce.