

Cvičení

Příprava na první den výuky

Vytvoříme si solution CSharp-pCv a v něm tři projekty typu WindowsForms:

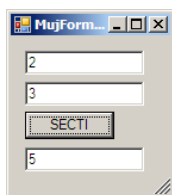
FormTextBox, VlaknaFormCas a VlaknaTrideni.

Práce na projektech bude mít několik etap, první z nich si vždy připravíte doma

Před závěrem vláken by měly být probrány delegáty

Ošetření vstupu z TextBoxu

Etapa 1: Vytvoření neošetřeného formuláře na součet čísel



V Solution CSharp-pCv založíme projekt typu Windows Form Application, nazveme jej FormTextBox.

V projektu se automaticky vytvoří formulář Form1, ve kterém pak nakreslíme tři textová pole tBox1, tBox2, tBoxVysledek a tlačítko btSecti. Poklepáme na tlačítku, tím se do viditelné části kódu vloží:

```
private void btSecti_Click(object sender, EventArgs e) { }
```

Do těla metody napíšeme kód, který vyčte do proměnných a,b čísla z tBox1 a tBox2. Poté zavolá metodu secti(). Výsledek vloží do tBoxVysledek.

Program odzkoušíme a ověříme si, že program havaruje v případě, že do prvního nebo druhého textového pole zadáme nějaký neplatný znak (takže výsledný řetězec nepůjde převést na celé číslo). Nebo když zůstane některé z těchto dvou polí prázdné. Chyba nastává na řádku, kde je metoda int.Parse().

Pozn.: typy výjimek, které mohou nastat při čtení z textového pole (konkrétně v etapě převodu z řetězce na číslo) zjistíme tak, že počkáme kurzorem na Parse. Objeví se informační obdélník, že generuje výjimky System.OverflowException, FormatException a ArgumentNullException. První výjimka nastane, když zadáme číslo větší, než se vejde do daného typu (např. float), druhá když zadáme nějaký znak, kvůli kterému nelze obsah textového pole převést na číslo (toto budeme dále řešit). Poslední výjimka v našem programu nemůže nastat, protože i když uživatel nic nezadá, tak textBox vrátí aspoň prázdný řetězec. Pokud bychom si tuto výjimku chtěli nasimulovat, museli bychom napsat natvrdo float.Parse(Null).

Pozn.: místo tBoxVysledek by bylo logičtější použít label nebo TextBox bez možnosti zapsat z klávesnice.