

Návratová hodnota, běhové chyby a výjimky, desetinná čísla



Návratová hodnota metod

- `int x;`
`x = Convert.ToInt32(textBox1.Text);`
Návratová (funkční) hodnota metody je zde celé číslo.
- `string slovo = Convert.ToString(x)`
– řetězec
- `Close()` – bez návratové metody

Návratová hodnota metod

- Metody s návratovou hodnotou poskytují tuto hodnotu – mohou být např. na pravé straně přiřazovacího příkazu
- Metody bez návratové hodnoty vystupují jako samostatné příkazy jejich typ je **void** (prázdný)

Běhové chyby a jejich ošetření

- Chyby, které mohou způsobit pád programu:
 - dělení nulou
 - vstup, který není možné převést na potřebný typ (Řetězec na číslo apod.)
- Ošetření:
 - Užití vhodných komponent
 - Přímo v programu
 - Výjimky

Výjimky

- Vytvoření chráněného bloku konstrukcí try-catch

- try

- {
➤ int x = Convert.ToInt32(textBoxCislo.Text);
➤ int mocnina = x * x;
➤ textBoxMocnina.Text =
Convert.ToString(mocnina);
➤ }

- catch

- {
➤ MessageBox.Show("Musíte zadat celé číslo");
➤ }

Výjimky

- Konstrukce **try-catch** uzavírá pokusný blok.
- Když vykonání některého příkazu v tomto bloku selže, místo běhové chyby se zbytek přeskočí a vykonají se příkazy v bloku za slovem **catch**. (**zachycení výjimky**)
- Do pokusného bloku uzavíráme také všechny příkazy, které navazují na příkaz, který by mohl způsobit problém..

Výjimky

- Try

- {

pokusný blok

- }

- Catch

- {

zachycení výjimky

- }

Desetinná čísla

- V počítači se ukládají jinak než čísla celá
- Výpočty s nimi provádí jiná část procesoru
- Výpočty jsou náročnější
- Vzniká při nich zaokrouhlovací chyba
- Typ **double**
- Převodní metodu **Convert.ToDouble**.
- Třída **Math** pro matematické operace

Příklady

- π – `Math.PI`
- a^b – `Math.Pow(a,b)`
- $\sqrt{x}$ – `Math.Sqrt(x)`

Práce s výrazy

- Jen kulaté závorky
- Priorita operací jako v matematice
- V nejistotě závorkovat

