

## Cvičení

### Cíl:

- Předávání informace o chybě mezi vrstvami aplikace.
- Nezávislost datové a logické vrstvy na uživatelském rozhraní.
- Využití v aplikaci s grafickým uživatelským rozhraním.

### Výstup:

- Plnění úkolu 2 - 5

### Zadání:

1. Jaké jsou typové praktické příklady pro využití mechanismu výjimek a indikování „chybového“ stavu?
2. Využijte projekt z přednášky k vytvoření aplikace s grafickým uživatelským rozhraním. Tato aplikace bude umožňovat počítat hodnotu faktoriálu čísla a zobrazovat chybové upozornění v případě chybného čísla na vstupu. Využijte k tomu implementované metody výpočtu faktoriálů `faktoriál2` a definované výjimky.  
Pozn: Prezentační vrstva aplikace bude ošetřovat více chybových stavů: konverze textu na celé číslo a výpočet faktoriálu čísla. Jaké výjimky budou ošetřeny?
3. Jaké třídy výjimek (Exception) známe? Jaké chybové stavy obvykle indikují?
4. Jaký je význam jednotlivých větví konstrukce *try-catch-finally*? Která větev bývá označována jako „kritická sekce“? Může jedna větev *catch* ošetřovat více typů výjimek? Jaký je význam větve *finally*?
5. Jak se označí, že metoda generuje typ výjimky? Jak se označí více typů výjimek generovaných metodou? Jaké klíčové slovo se použije v deklaraci metody (v hlavičce metody)?
6. Zmapujte systém tříd mechanismu výjimek. Jaká je dědičná hierarchie? Jaké typy výjimek jsme detekovali?