Návrhová dokumentace – ConfigRW

Při tvorbě API pro knihovnu ConfigRW byl kladen důraz na co nejjednoduší použití na straně uživatele knihovny. Proto jsme se rozhodli pro minimalistické rozhraní, které vypadá následovně:

* Třída Configuration – slouží k vytváření konfiguračních objektů
* Interface IConfiguration – používaný pro zpřístupnění služeb na konfiguračních objektech
* Třída QualifiedName – slouží pro jednoznačné pojmenování elementů konfiguračního souboru
* Specifikační Attributy – umožňující rozšířené možnosti specifikace struktury konfiguračního souboru
* Výjimky – používané pro upozornění na chybové situace a špatná použití knihovny

# Návrh třídy Configuration

Abychom mohli pracovat s konfiguračním souborem, musíme nejprve specifikovat jeho strukturu. Způsobů jak specifikaci provést je několik. Například zavést formální gramatiku pro popis optionů a sekcí, zadefinovat XML popis souboru, případně bychom mohli vytvořit builder jímž strukturu specifikujeme. My jsme se však rozhodli pro umožnění popisu konfigurační struktury pomocí .NET interface. Tím získáme možnost typové kontroly již za překladu. Samotné využití ve zdrojovém kódu navíc vypadá velmi intuitivně.

**Motivační příklad:**

*Určení struktury přístupem „Builder“*   *stejná struktura přes interface*



*Použití konfigurace, která není silně typovaná typ je jednoznačně určen pomocí interface*



Jak je vidět z motivačního příkladu, naše knihovna po uživateli nevyžaduje žádné explicitní získávání nebo měnění hodnot konfigurace, ani jejich přetypování. Použití načtených hodnot je tedy velmi jednoduché a nepřidává zbytečnou zavlečenou složitost.

**Způsoby vytváření konfiguračních objektů:**

Naše knihovna nabízí uživateli možnost vytvořit konfigurační objekt kromě standardních způsobů jako je ze souboru nebo streamu také z defaultních hodnot. Pro tuto možnost jsme se rozhodli, neboť uživatel může do struktury popisující konfigurační soubor jednoduše specifikovat defaultní hodnoty a komentáře k jednotlivým optionům. Jejich vypsáním do souboru pak získá předpis konfigurace aniž by ji musel například ručně vytvářet.

# Návrh interface IConfiguration

Interface IConfiguration musí být předkem popisu struktury konfiguračního souboru, se kterým pracuje knihovna ConfigRW. Pro tuto povinnost jsme se rozhodli kvůli snadnému zpřístupnění služeb na konfiguračním objektu. Kdybychom totiž chtěli jiným způsobem například zpřístupnit ukládání konfiguračních objektů, vypadalo by to nějak takto:

Configuration.Save(configObj, "output.ini");

Problémem je, že pokud by configObj nebyl odděděn od nějaké společné třídy nebo interface, musela by metoda Save přijímat objekty typu System.Object. Tím bychom však ztratily možnost ubránit uživatele špatnému použití knihovny typovou kontrolou. Proto jsme se rozhodli požadovat po struktuře konfiguračního souboru rozhraní typu IConfiguration. Použití konfiguračního objektu v kódu se tím také zpřehlední:

configObj.SaveTo("output.ini");

# Návrh třídy QualifiedName

# Návrh specifikačních atributů

# Návrh výjimek