

# Zadání úlohy do projektu z předmětu IPP 2011/2012

Zbyněk Křivka a Dušan Kolář

E-mail: {krivka, kolar}@fit.vutbr.cz, {54 114 1313, 54 114 1238}

## JSN: JSON2XML

**Zodpovědný cvičící:** Zbyněk Křivka (krivka@fit.vutbr.cz)

### 1 Detailní zadání úlohy

Vytvořte skript pro konverzi JSON formátu (viz RFC 4627) do XML. Každému prvku z JSON formátu (objekt, pole, dvojice jméno-hodnota) bude odpovídat jeden párový element se jménem podle jména dvojice a obsahem podle hodnoty dvojice. Každé pole bude obaleno párovým elementem `<array>` a každý prvek pole bude obalen párovým elementem `<item>`. Ve jméně elementu všechny nepovolené<sup>1</sup> znaky nahraďte pomlčkou (vznikne-li i tak invalidní jméno XML elementu, skončete s chybou a návratovým kódem 51).

JSON hodnoty typu `string` a `number` a JSON literály `true`, `false` a `null` budou transformovány v závislosti na parametrech skriptu na atribut `value` daného elementu s odpovídající hodnotou (stejného tvaru jako v JSON vstupu; nezapomeňte na požadavek uzavírání atributů v XML) nebo na textový element v případě hodnoty typu `string` a `number` či na párový element (ve zkráceném zápise) `<true/>`, `<false/>` a `<null/>`. Je-li hodnotou desetinné číslo, bude na výstupu zaokrouhleno na odpovídající celé číslo.

**Tento skript bude pracovat s těmito parametry:**

- `--help` viz společné zadání všech úloh
- `--input=filename` zadaný vstupní JSON soubor v UTF-8
- `--output=filename` textový výstupní XML soubor s obsahem převedeným ze vstupního souboru
- `-n` negenerovat XML hlavičku<sup>2</sup> na výstup skriptu (vhodné například v případě kombinování více výsledků)
- `-r=root-element` jméno párového kořenového elementu obalující výsledek. Pokud nebude zadán, tak se výsledek neobaluje kořenovým elementem, ač to potencionálně porušuje validitu XML (skript neskončí s chybou). Zadání řetězce `root-element` vedoucího na nevalidní XML značku ukončí skript s chybou a návratovým kódem 50.
- `--array-name=array-element` tento parametr umožní přejmenovat element obalující pole z implicitní hodnoty `array` na `array-element`. Zadání řetězce `array-element` vedoucího na nevalidní XML značku ukončí skript s chybou a návratovým kódem 50.

---

<sup>1</sup>Požadavky na validitu elementů XML v UTF-8 je třeba kontrolovat dle standardu minimálně v rozsahu 0x00 až 0x7F.

<sup>2</sup>Tradiční XML hlavička je `<?xml version="1.0" encoding="UTF-8"?>`

- `--item-name=item-element` analogicky, tímto parametrem lze změnit jméno elementu pro prvky pole (implicitní hodnota je `item`). Zadání řetězce `item-element` vedoucího na nevalidní XML značku ukončí skript s chybou a návratovým kódem 50.
- `-s` hodnoty dvojic typu `string` budou transformovány na textové elementy místo atributů.
- `-i` hodnoty dvojic typu `number` budou transformovány na textové elementy místo atributů.
- `-l` hodnoty literálů (`true`, `false`, `null`) budou transformovány na elementy `<true/>`, `<false/>` a `<null/>` místo na atributy.
- `-c` tento přepínač oproti implicitnímu chování aktivuje překlad problematických znaků. Pro XML problematické znaky s UTF-8 kódem menším jak 128 ve vstupních řetězcových hodnotách konvertujte na odpovídající zápisy v XML pomocí metaznaků `&` (např. `&amp;`, `&lt;`, `&gt;` atd.). Ostatní problematické znaky nemusíte uvažovat.
- `-a`, `--array-size` u pole bude doplněn atribut `size` s uvedením počtu prvků v tomto poli.
- `-t`, `--index-items` ke každému prvku pole bude přidán atribut `index` s určením indexu prvku v tomto poli (číslování začíná od 1, pokud není parametrem `--start` určeno jinak).
- `--start=n` inicializace inkrementálního čítače pro indexaci prvků pole (nutno kombinovat s parametrem `--index-items`) na zadané kladné celé číslo `n` včetně nuly (implicitně `n = 1`).

Uvažujte definici formátu JSON prostřednictvím RFC 4627. *Stručně*: Soubor obsahuje jeden globální objekt nebo pole. *Objekty* jsou zapsány ve složených závorkách a obsahují čárkami oddělené dvojice *jméno : hodnota*. *Jméno* je řetězec jazyka JavaScript. *Pole* je zapsáno do hranatých závorek a obsahuje hodnoty oddělené čárkami. *Hodnota* může být další objekt či pole nebo řetězec či číslo jazyka JavaScript. Hodnotou může být také literál `true`, `false` nebo `null`. Bílé znaky mimo řetězce jsou ignorovány. Jména v rámci objektu jsou unikátní.

## Reference:

- D. Crockford: RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON), 2006. [cit. 2012-02-09]. Dostupné z: <http://tools.ietf.org/html/rfc4627>

## 2 Bonusová rozšíření

- **JPD** (0,5 bodu): Podpora parametru `--padding`, který u použitých čítačů (parametr `--index-items`) provede doplnění takového počtu nul zleva, aby všechny čísla dané sekvence měla stejný počet číslic, který bude ale minimální možný.
- **FLA** (1,0 bodu): Podpora parametru `--flattening`, který provede zploštění vstupního hierarchického formátu následujícím způsobem. Na výstup budou převedeny pouze listové dvojice jméno-hodnota (kde hodnota je skalárního typu, tj. žádný další objekt). Pokud je hodnota dvojice pole, tak z něj bude vzata pouze lexikograficky nejmenší skalární hodnota (ne objekt ani další pole) a zbytek prvků bude zahozen. Pokud takové pole žádnou skalární hodnotu neobsahovalo, tak bude z výstupu zcela vypuštěno. Při přidávání dvojic jméno-hodnota na nejvyšší úroveň mohou nastat tři případy: (1) dvojice je dle jména nová, pak provedeme obyčejné přidání; (2) jméno dvojice již ve výsledku existuje a jeho hodnota je skalární, pak je hodnota

výstupní dvojice nahrazena polem obsahujícím původní i novou hodnotu pro dané jméno; (3) jméno dvojice již ve výsledku existuje a jeho hodnota je již pole, pak je do tohoto pole přidána nová hodnota. Pokud jsou v takto vzniklém poli výhradně číselné hodnoty, dojde k jejich vzestupnému seřazení dle velikosti. Pokud je některá hodnota vzniklého pole nečíselná, provede se řazení lexikograficky vzestupně, ale výsledné hodnoty budou ve výstupu v původním typu (tj. konverze na řetězce se provede pouze za účelem seřazení a nikoli výstupu). V poli se mohou vyskytovat vícenásobné výskyty stejných hodnot.

### 3 Poznámky k hodnocení

Výsledný XML soubor bude porovnáván s referenčními XML soubory nástrojem na porovnání XML souborů, který se umí správně vypořádat například s různým odsazením elementů nebo s prohozeným pořadím podelementů v rámci jednoho elementu.