

## Zadání

Úkolem projektu je implementování skriptu v PHP5, který načte vstup ve formátu JSON a převede na výstup ve formátu XML, který je generován na základě vstupních požadavek.

## Řešení

Vzhledem k syntaxi JavaScript objektové notace a možnosti využít PHP funkce `is_array()` a `is_object()` bylo vhodné projekt implementovat objektivě rekurzivně. Skript je podle logických celků rozdělen do třech tříd:

### 1. Třída pro načítání a zpracování argumentů

Třída obsahuje proměnné, které jsou inicializované na výchozí hodnoty. Funkce `loadArgs()` cyklem třídí hodnoty z přepínačů a tyto hodnoty přiřazuje proměnným. Taktéž ověřuje správnost argumentů a jejich správné kombinace. Návrátový kód chybných argumentů je 1. Když se pomocí přepínačů `-r`, `--array-name` nebo `--item-name` zadá nevalidní název XML elementu, skript končí s návratovým kódem 50.

### 2. Třída pro zpracování JSON vstupu

Funkce `readInput()` načítá vstupný JSON soubor. Když se tahle operace povede, zavolá se funkce `fixJson()`, která provede funkci `json_decode`. Když je vstupní JSON nevalidní (což se projeví neúspěchem `json_decode`) a je nastaven přepínač `--error-recovery` proběhne úprava vstupního JSON. Tahle úprava je rozšířením projektu JER a je uskutečněna doplněním/odebráním chybných závorek, čárek nebo uvozovek pomocí regulárních výrazů. Tahle úprava probíhá, dokud není výstup `json_decode` validní, tudíž může se zpracovat do XML.

Ke konci se vytvoří instance třídy `Xml`, což vytvoří kořenový objekt, který se dále zpracuje a vypíše na výstup/do výstupního souboru a když není nastaven přepínač `-n`, přidá se hlavička. Chyby třídy `Json` mají návratové kódy čísla 2, 3 a 4.

### 3. Třída objektů a vytvoření výstupního XML

Třída vytváří rekurzivní instance objektů podle aktuálních dat dekodovaného JSON vstupu. Když jsou data pole, zpracuje se taky požadavek přepínače `-t` a když je to kombinované s přepínačem `--padding`, data se budou indexovat s případným odsazením znakem 0. Tahle úprava je rozšířením projektu JPD.

Výslední objekt se zpracuje ve funkci `writeOutput()`, která data serializuje na výstup. Podle nastavených přepínačů se upravuje výsledný formát XML výstupu. Pro vytváření elementů jsou použity pomocné funkce. Funkce `getName()` vrací jméno elementu, provádí požadavek přepínače substituce nevalidních znaků a kontroluje validnost výsledné XML značky pomocí regulárního výrazu. Při zjištění nevalidní značky skript končí s návratovým kódem 51. Funkce `getAtribs()` vrací všechny atributy elementu. Funkce `getValue()` vrací hodnotu elementu. Když je hodnotou elementu literál, vrací jeho textovou podobu. Při desetinných číslech vrací celé číslo zaokrouhlené dolů. Když je nastaven přepínač pro konvertování problematických znaků použije se funkce `htmlspecialchars()`. Funkce `getType()` vrací atribut `type` elementu, pokud je nastaven přepínač `--types`.

**Implementována rozšíření projektu:** JER, JPD