

## Dokumentace úlohy JMP: Jednoduchý makroprocesor v Python 3 do IPP 2014/2015

Jméno a příjmení: Petr Polanský

Login: xpolan07

### Popis zadání:

Cílem je vytvoření makroprocesoru, který vyhledává makra na vstupu a nahrazuje je metodou makra, do které dosadí načtené parametry.

### Popis implementace:

Skript pracuje jako konečný automat, který načítá vstup znak po znaku a dochází k rozlišení, zda se jedná o název makra, blok nebo jiný znak. Tabulka maker je vytvořena jako pole struktur. Každá struktura kromě jména a metody obsahuje jména a umístění jednotlivých parametrů a identifikátor, zda se jedná o vestavěné makro. Ty jsou do tabulky vektorů vloženy před samotným zpracováním vstupu.

Při názvu makra se nejprve kontroluje, zda se nejedná o některé z vestavěných maker. Tyto makra jsou zpracovány každé zvlášť. V opačném případě se ihned kontroluje, zda makro se stejným názvem již existuje. Pokud ano, dochází k redefinici makra, která je provedena smazáním původního makra z tabulky maker a vložením makra nového. Parametry jsou kontrolovány pomocí regulárních výrazů. Blok je rozpoznán první „{“ na vstupu, kde ostatní jsou přeskakovány a ihned vypisovány. Každá závorka „{“ a „}“ má svůj čítač. Pokud se hodnoty rovnají, dojde k vyskočení z bloku. U bloků se rozlišuje jejich vlastnost pomocí identifikátorů, které určují, zda se jedná o obyčejný blok, seznam parametrů makra, metodu makra nebo tvoří parametr při volání.

Dále je kontrolováno při posledním znaku ze vstupu, jestli, byly všechny procedury úspěšně dokončeny např. ukončení bloku a načtení všech parametrů.

Samotná expanze probíhá ve funkci „nalezení“, kde jsou porovnány struktury z tabulky maker se strukturou naplněnou při volání makra. Dochází také k ověření, zda již byly načteny všechny parametry. Poté jsou porovnány parametry podle pořadí jejich načtení. Pokud dojde ke shodě, je v metodě vyhledán parametr definovaného makra v metodě. Je-li nalezen, je nahrazen parametrem načteným při volání makra. Metoda je takto procházena tolikrát, kolik má makro parametrů. Skript tudíž nenačte všechny parametry najednou, ale ukládá si je postupně, dokud čítač načtení parametrů není roven počtu očekávaných parametrů. Po dosazení všech načtených parametrů je k metodě připojen zbytek vstupu a je vypočtena nová délka vstupního řetězce.

Při oddefinování je jméno makra, které má být odstraněno, načteno jako parametr, kde se jménem se načte i „@“. To je dodatečně odstraněno pomocí funkce replace.

Samostatný blok je zpracován funkcí zprac\_blok, která slouží hlavně pro odlišné zpracování escape sekvencí v bloku a mimo blok. Ostatní znaky v bloku nejsou nijak zpracovány. Prohledání metody a dosazení parametrů probíhá vyhledáním znaku „\$“, který značí začátek parametru nebo „}“, která značí konec. Pro co nejdelší možný parametr je řetězec po každém znaku kontrolován, zda odpovídá, syntaxi parametru. Pokud další znak již neodpovídá syntaxi, byl nalezen nejdelší možný parametr v metodě. Pokud nalezený název parametru se neshoduje definovaným, nedochází k nahrazení. Jméno makra je taktéž kontrolováno regulárními výrazy.

Výstupní řetězec je těsně před výpisem zbaven přebytečných bílých znaků.

S pythonem jsem neměl moc dobré zkušenosti, zvláště s oddělovači.