

Dokumentace úlohy: JMP: Jednoduchý makroprocesor

Jméno a příjmení: Filip Kalous

Login: xkalou03

1) Zadání

Úkolem bylo vytvořit jednoduchý makroprocesor, který je zjednodušenou verzí použitého makroprocesoru v TEXu. Program vyžaduje ke své práci vstup zadaný souborem v parametrech programu nebo na standardním vstupu. Skript také využívá vestavěné knihovny *re* jazyka Python, která obsahuje funkce pracující s regulárními výrazy.

2) Implementace

Tento projekt už mnohem více nežli první vyhovoval objektovému návrhu. Skript obsahuje celkově 3 třídy pro oddělené pracování s čtením vstupu, tabulkou maker a samotnými makry.

Třída *Reader* byla vytvořena pro zjednodušení práce při čtení vstupu. Při použití třídní metody *getc()* programátora nemusí zajímat, zda má číst vstup ze souboru či řetězce apod. Obsahuje i metodu pro přečtení bloku, který vypadá takto {xxx}, xxx představuje libovolný řetězec.

Třída *Table* představuje samotnou tabulku maker, a i přesto, že v celém programu je jen jedna její instance, rozhodl jsem se pro ni vytvořit speciální třídu. Tato třída obsahuje třídní metody na přidání, odstranění maker a expandování makra. Samotná tabulka maker je implementována pomocí *dictionary*.

Třída *Macro* obsahuje všechny informace, které potřebujeme k práci s daným makrem. Například, zda je chráněn proti redefinování, počet argumentů makra, atd.

Celý program je řízen funkcí *main()*.

3) Zpracování parametrů

Při spuštění skriptu se jako první provádí vyhodnocení parametrů programu. V tomto skriptu jsem přemýšlel, zda použít opět *getopt*, ale nakonec jsem použil knihovnu *argparse*. Práce s touto knihovnou je velmi jednoduchá, osobně bych řekl, že se mi s ní zatím pracovalo nejlépe. Pomocí nastavení správných parametrů dokáže knihovna i vypisovat nápovědu, čehož jsem také nakonec využil.

4) Možný rozpor se zadáním

Skript je napsán tak, že při čtení jména makra čte, dokud nenajde znak, který ukončuje název. Až pozdě mi došlo, že v některých případech by makroprocesor nemusel rozpoznat jméno makra nebo přečíst zbytečně dlouhý název.

5) Závěr

Skript byl po dokončení řádně otestován přiloženou sadou testů. V průběhu psaní byl také testován a pro jistotu spouštěn na školním serveru Merlin z důvodu kompatibility. Práce na tomto projektu mě velmi zaujala a posunula ve studiu, protože jsem byl konečně donucen se naučit alespoň základy jazyka Python.