

Standardní zadání semestrální práce pro KIV/PT 2016/2017

Zadání je určeno pro **dva** studenty. Práce zahrnuje dvě dílčí části - vytvoření funkčního programu diskrétní simulace a napsání strukturované dokumentace.

Zadání:

Vytvořte program, který načte text ze souboru a ze všech unikátních slov vyskytujících se v textu vytvoří slovník. Slovník bude možné vyexportovat ve vhodně zvoleném formátu do souboru a znovu použít při dalším spuštění programu. Program dále umožní napsat nebo načíst libovolný text a v něm vyhledat zadané slovo. Pokud se slovo bude vyskytovat v prohledávaném textu, tak program vypíše počet výskytů a uvede u všech výskytů počáteční a koncový index, kde se v textu slovo nachází. Pro prohledávání textu využijte algoritmu komprimované trie. Pokud se zadané slovo v textu nenachází, tak vypište maximálně 10 nejbližších slov ze slovníku. K porovnání vzdáleností mezi hledaným slovem a slovy ze slovníku použijte Levensteinovu metriku probíranou na přednáškách. Uživatel by měl mít také možnost přidat hledané slovo do slovníku, pokud se v něm nenachází.

Uživatelské rozhraní programu může být grafické i konzolové. Program bude mít ošetřeny všechny vstupy a zdrojový kód projde validací nástrojem PMD. Kód programu bude okomentovaný javadoc komentáři. Struktura dokumentace je uvedena níže v tomto dokumentu.

První kontrola:

- Připravte rozumná vstupní data a uložte je ve vhodném formátu. **(10b.)**
- Zvolte a implementujte vhodné datové struktury pro reprezentaci vstupních dat, důsledně zvažujte paměťovou náročnost zvolených struktur a časovou náročnost algoritmů pro následovné výpočty. **(10b.)**
- Provedte základní vyhledávání (nemusí být implementovaná trie ani vzdálenost mezi řetězci) a výsledek vyhledávání (počet nalezených slov a pozice v textu) vypište na obrazovku a do souboru. **(10b.)**

Výše popsaná část bude váš minimální výstup při kontrolním cvičení cca v polovině semestru.

- Vytvořte prostředí pro snadnou obsluhu programu (menu, ošetření vstupů) - nemusí být grafické. **(5b.)**
- Implementujte vyhledávání zadaného slova algoritmem komprimované trie. **(5b.)**
- Pro hledaná slova, která nejsou ve slovníku, použijte Levensteinovu metriku pro nalezení až deseti nejbližších slov ve slovníku, seřadte je podle vzdálenosti a umožněte uživateli přidat hledané slovo do slovníku. **(10b.)**
- Vytvořte dokumentační komentáře ve zdrojovém textu programu a vygenerujte programovou dokumentaci (Javadoc). **(10b.)**
- Vytvořte kvalitní dále rozšiřitelný kód – pro kontrolu použijte softwarový nástroj PMD (více na <http://www.kiv.zcu.cz/~herout/pruzkumy/pmd/pmd.html>), soubor s pravidly pdmrules.xml najdete na portálu v podmenu *Samostatná práce*. **(10b.)**
 - mínus 1 bod za vážnější chybu, při 5 a více chybách nutno opravit
 - mínus 2 body za 10 a více drobných chyb

V rámci dokumentace: **(20b.)**

- připojte zadání **(1b.)**,
- popište analýzu problému **(6b.)**,
- popište návrh programu (např. UML diagram) **(6b.)**,
- vytvořte uživatelskou dokumentaci **(5b.)**,
- zhodnoťte celou práci, vytvořte závěr **(2b.)**.