

Dojo 2 – konstruera algoritmer

Förberedelse

Läs kapitel 7 och kapitel 9 (speciellt om filhantering) i Learning Python.

Uppgift

I den mån följande beskrivning saknar information är det upp till dojon att gemensamt hitta bra lösningar.

Dojon ska lösa följande uppgift: vi ska göra ett program `z_fileformat.py` med funktioner som arbetar på texten i en textfil på diverse olika sätt. Textformatet Z är tänkt för att hantera t ex README-filer på ett ”intelligent” sätt. Vi ska här göra funktioner som manipulerar och extraherar information från en sådan fil. Samtliga gjorda funktioner ska testas genom att testkod skrivs som använder en lämplig testfil. Gör följande funktioner (hitta på bra namn!) till en modul `z_textformat.py`:

1. Rader som börjar med `#` är metakommentarer som inte kunna filtreras bort ur filen. T ex `# code: UTF-8` kan vara en metakommentar om vilken typ av tecken som finns i filen. Gör en funktion som returnerar en lista där alla rader som börjar med `#` har filtrerats bort. Börja med att göra pseudokod för lösningen.
2. Segment som börjar med `TODO` och slutar med `END-TODO` innehåller listor av `TODO`-enheter. Varje enhet börjar med tecknet `"""` i början av raden och är en rad lång. `TODO` och `END-TODO` kan stå var som helst på en rad. Gör en funktion som returnerar en lista av alla `TODO`-enheter som en lista av strängar. Börja med att göra pseudokod för lösningen.
3. Nyckelord med speciell betydelse märks upp med `**` före och efter ordet. T ex `**java**`. Gör en funktion som returnerar en dictionary med alla nyckelord som nycklar och antal förekomster som värden. T ex `{'Emacs': 4, 'Python': 2, 'Java': 1}` ska returneras om antalet förekomster av nyckelordet Emacs förekommer 4 gånger, Python 2 gånger och Java 1 gång.
4. Baserat på funktionerna i 1-3 ska vi nu göra en funktion som skapar ett s k Z-metadokument som innehåller all information som kan extraheras med hjälp av Z-formatets övriga funktioner. Z-metadokumentet ska även innehålla filnamnet på originaldokumentet. Z-metadokument ska alltså innehålla information om vilka `TODO`-segment som förekommer i originaldokumentet samt nyckelord och antalet förekomster. Hitta på ett lämpligt format som a) är läsbart och b) kan läsas in igen av programmet.
5. Gör en funktion som kan läsa in ett Z-metadokument och översätta dessa metadata till lämpliga datastrukturer.
6. Gör en funktion som läser in *en samling* Z-metadokument och översätter till lämplig datastruktur.
7. Gör en sökfunktion som givet en samling inlästa Z-metadokument tillåter sökning baserat på kommentarer, `TODO`-enhet och nyckelord. Funktionen ska returnera objektstrukturerna för de Z-metadokument som matchar sökvillkoret.

Om tid finns kvar och ni är klara med ovanstående. Fundera själva ut en lämplig utvidgning eller omdesign av er programmodul.