

Todo-tietokanta

Ohjelma toteuttaa Todo-tietokannan tekstitiedostona ja tekstikäyttöliittymän, jonka kautta käyttäjä voi nähdä Todo-listauksen sekä etsiä, tallentaa, muokata ja poistaa Todoja. Käyttäjä voi siis pitää kirjaa tekemättömistä tehtävistään ohjelman avulla.

Tässä toteutuksessa Todoon liittyy nimi (name), kuvaus (description), määräaika (deadline), prioriteetti (priority) ja status (done_status). Todot tallennetaan tekstitiedostoon (.txt) muodossa, jossa jokaisen kentän jälkeen on välilyönti, | -merkki ja toinen välilyönti siten että yhtä riviä kohden on yksi Todo.

Todo luokka

Todo-luokka toteutettiin ohjelmoinnin helpottamiseksi ja myös ohjelman laajennettavuuden helpottamiseksi, vaikka tosin sitä ei välttämättä tarvitsisi. Kuitenkin tämä luokka kapseloi Todo-käsitteen.

Tekstitiedoston valinta

Aivan ensimmäiseksi käyttäjä valitsee tekstitiedoston johon Todot tallennetaan ja tämä tiedosto luodaan, jos sitä ei vielä ole. Samalla myös ohjelma tarkistaa, että tiedosto on oikeaa muotoa (.txt päätteellä) tarkistamalla neljä viimeistä kirjainta. Kun käyttäjä on valinnut tallennustiedoston ei sitä enää kysytä uudelleen, ellei käyttäjä halua itse vaihtaa tallennustiedostoa, vaan tämän jälkeen käyttäjä pääsee Todojen pariin.

Todojen listaus: `todo_list/read`

Listauksessa kaikki Todot kerätään listalle ja tulostetaan käyttäjän nähtäväksi.

Todon etsiminen: `find`

Etsiminen tapahtuu Todon nimen perusteella, siten että jos nimeen sisältyy hakusana niin tämä Todo sisältyy tuloksiin. Hakusana on case-sensitive ja jos hakusanan jättää tyhjäksi niin kaikki Todot palautetaan käyttäjälle.

Todon Tallennus: `create/save`

Tallennuksessa käyttäjältä kysytään viiden eri kentän tietoja yksi kerrallaan ja lopulta käyttäjä saa päättää tallennetaanko tieto tiedostoon. Lisäksi jos käyttäjä teki näppäilyvirheen niin hän voi syöttää tiedot uudelleen.

Todon muokkaus: update/todo_modify

Tiedon muokkauksessa käyttäjä ensin etsii Todon nimen perusteella (tyhjä hakusana palauttaa kaikki), ja saa nähtäväkseen listauksen jossa näkyy jokaiseen TODOon liitetty indeksi, jonka perusteella käyttäjä voi valita mitä TODOa muokataan. Muokkauksessa käyttäjän ei tarvitse syöttää kaikkia tietoja uudelleen, jos hän haluaa vain muokata yhtä alitietoa sillä jättämällä kentät tyhjäksi ja muokkaamalla vain haluttua kohtaa, ohjelma täyttää vanhat tiedot automaattisesti ja sitten vielä kysyy varmistuksen tallennetaanko.

Itse tallennus tapahtuu luomalla uusi tiedosto (.tmp) johon kopioidaan kaikki vanhat rivit ja muutettavan rivin paikalla on uusi muokattu tieto. Lopuksi uusi tiedosto uudelleennimetään `os.rename()` funktion avulla alkuperäisen tiedoston nimellä (mikä näyttäisi tuottavan halutun lopputuloksen). Näin kaikki tiedot säilyvät samoilla paikoilla.

Todon poisto: delete

Tiedon poisto toimii samalla tavalla kuin tiedon muokkaus valittavan Todon osalta, sekä myös tallennuksessa käytetään uutta tilapäistiedostoa. Poistettua riviä ennen olevat rivit säilyvät ennallaan ja poistetun rivin jälkeen olevat tiedot siirtyvät yhden rivin ylemmäs.

Importit

Ainoa import on `os.path` koska etsin tietoa miten tarkistaa onko tiedosto jo olemassa, ja löysin tällaisen ratkaisun.

Toiminnallisuus yleisesti

Ohjelmassa on pyritty ensisijaisesti siihen, että se ei kaadu missään normaaleissa olosuhteissa varautumalla virhesyötteisiin ja muihin virhetilanteisiin sekä käyttäjäliittymästä on pyritty tehdä mahdollisimman helppokäyttöinen. Esimerkiksi tämän vuoksi työskentelytiedosto valitaan ensin, jotta sitä ei koko ajan tarvitsisi uudelleen näppäillä.

Lisäksi tiedon muokkaus ja poisto yrittää noudattaa samaa ajatusta kuin jos käyttäjä muokkaisi itse tietoa tekstieditorissa eli muokattu tieto säilyy paikallaan ja poistettu tieto aiheuttaa sen alapuolella olevien rivien siirron ylös. Toisaalta turhaa monimutkaisuutta tiedon muokkaamisesta tiedoston keskellä on vältetty luomalla uusi tiedosto muokkauksen ja poiston yhteydessä.

Rajoitukset

Tiedoston koko ei voi olla mielivaltaisen suuri koska esimerkiksi kaikki Todot voidaan listata jolloin on olemassa raja sille kuinka iso lista tietokoneen muistissa voi olla kerrallaan. Toisaalta tällaisen tekstitiedoston kasvu normaalikäytössä on hyvin epätodennäköistä.

Tietokanta ei ole mitenkään erityisen tehokas, sillä tietoa haetaan aina levyltä (eli tekstitiedostosta) eikä esimerkiksi tiedon hakua nopeuteta esimerkiksi muistissa olevilla hajautusrakenteilla/hakupuilla.

Juuso Silvennoinen

Toisaalta tällä tuskin on mitään merkitystä, koska käyttäjällä tuskin on biljoonia Todoja työn alla ja tarvittaessa vanhoja voi poistaa.

Ohjelmaa on käytetty vain Linux-ympäristössä ja testattu pelkästään manuaalisesti (eli testattu meneekö rikki).