

# Ajopäiväkirja 2015

Toivo Koskinen

Kerttulin lukio

28.3.2015

## 1. Tehtävän kuvaus:

Tehtäväksi annettiin valmistaa Python ohjelma IT3-kurssin lopputyöksi. Ohjelman täytyy pitää sisällään asiat, joita kurssilla on käsitelty. Ohjelman tarkoitus on olla selkeä ja yksiselitteinen.

## 2. Ratkaisuperiaate

Kirjoitin Ajopäiväkirjaohjelman, jonne voidaan kirjoittaa ylös autolla tai muulla polttoainekäyttöisellä ajoneuvolla ajetut kilometrit, ostetun polttoaineen määrä ja litrahinta.

Ohjelma käyttää kaikkia kurssin osa-alueita ja hyödyntää niitä niin järkevästi kuin oli taidoillani mahdollista. Ohjelman pää.py tiedosto on ehkä hieman sekava alkuperäisen valmiin suunnitelman puuttumisen vuoksi. Tästä syystä tiedoston alkuun on kirjoitettu suurin osa ohjelman funktioista ja vasta niiden jälkeen alkaa varsinainen pää-ohjelma riviltä 180.

## 3. Ohjelman ja sen osien kuvaaminen

Ohjelma alkaa ilmoittamalla pikaisesti ohjelman tekijän ja toivottaa käyttäjän tervetulleeksi käyttämään ohjelmaa. Saman tien ilmoituksen jälkeen ohjelma tarkistaa, onko data.txt tiedostoa olemassa tai onko sinne kirjoitettu mitään. Jos tiedostoa ei löydy, se luodaan. Jos tiedosto löytyy, mutta sinne ei ole vielä kirjoitettu mitään, ottaa ohjelma ylös käyttäjän nimen, kulkupelin ja valmistusvuoden. Nämä tiedot ohjelma lisää *data.txt* tiedoston alkuun *nimi()*- funktiolla. Ohjelman eri osista pääsee pois painamalla (Q/q).

Funktioita:

-nimi() Kerää ylös nimitiedot ensimmäisellä käynnistyskerralla ja lisää ne data.txt-tiedostoon.

-add()-funktioita käytetään erinäisten merkkijonojen lisäämiseen data.txt tiedoston viimeisen merkkijonon perään. Funktio ottaa yhden argumentin.

-addp() taasen tekee saman kuin add, mutta lisää rivinvaihdon merkkijonon jälkeen. Funktio ottaa yhden argumentin.

-historia() tulostaa koko data.txt tiedoston näytölle ilman muokkauksia

-keski2() palauttaa data.txt tiedostosta luetut tiedot (L, km, L/€) liukulukuina omiin listoihinsa, joista dataa on helppo käsitellä.

-laskin() voisi olla oma moduulinsa. Sisältää valikon, josta voi valita haluaako laskea edellisen tankkauskerran keskikulutuksen  $L/(km_2 - km_1)$ -kaavalla, laskea kahden edellisen tankkauksen keskikulutuksen, vai historian ajan keskikulutuksen. Funktio käyttää while True rakennetta, jolloin valikkoon pääsee suoraan takaisin, kun laskutoimitus on suoritettu. Loopista pääsee pois painamalla Q (iso tai pieni). Funktiossa on jokseenkin rankka virheidenhallinta, jota tarvitaan siihen, että jotkin funktion laskutoimitukset eivät toimi ilman tarpeeksi montaa polttoainetietolisäystä. Ottaa kaksi argumenttia.

-polttoaine.py (polaskin()) käyttää while-rakennetta laskin-funktion tavoin. Polaskin() tulostaa ruudulle polttoaineen keskimääräisen listahinnan laskemalla kaikki hinta-listan alkiot yhteen ja jakamalla summan alkioden lukumäärällä. Ottaa yhden argumentin.

-raha.py (rahalaskin()) laskee bensa-listan alkiot yhteen ja kertoo summan hintojen keskiarvolla. Ottaa kaksi argumenttia.

-inikilsa() lisää tiedostoon data.txt päivämäärän, jolloin tieto on lisätty päiväkirjaan, kilometrilukeman, ostetut polttoainelitrat ja litrahinnan, jotka se kysyy käyttäjältä. (Käytetään vain ensimmäisellä avauskerralla)

-*tankkaus()* on toiminnaltaan muuten inikilsan kanssa täysin samanlainen, mutta kysymysten virkkeet ovat muotoiltu eri tavalla. (Käytetään jatkossa, kun ensimmäiset tiedot on jo lisätty)

## 4. Testauksessa havaittuja ongelmia

4.1. Ohjelmaan on pakko lisätä jokin järkevä litramäärä jo ensimmäisellä avauskerralla, vaikka ei olisi polttoainetta silloin vielä ostanutkaan. Tämä siksi, että tiedostosta data.txt lukeminen tietyltä riviltä ja siirtäminen listaan oli vaikeaa ja pelkän kilometrimäärän (ilman polttoainetietoja) lisääminen niin, että jatkossa tulevat laskutoimitukset onnistuisivat, oli liian vaikea tehtävä.

4.2. Merkkijonot numeroille tarkoitetuissa paikoissa aiheuttaa virheitä laskutoimituksissa.

4.3. Osa toiminnoista ilmestyy tulkin yläosaan, ja saattaa olla vaikeampi huomioida.

4.4. Polttoaine-funktiossa tulee ylimääräisiä numeroita ruudulle

## 5. Testimenetelmät

5.1. Itse räpläämällä kaikkia nappeja, näin ei löytynyt virheitä.

5.2. Lisäämällä numeroiksi tarkoitettuihin kohtiin data.txt tiedostoon merkkijonoja, aiheuttanee ongelmia laskuissa.

5.3. Ohjelmaa on testattu sille tarkoitetuilla syötteillä (kerrotaan ohjelmassa) ja toimii ainakin omien testieni mukaan.

Toivo Koskinen 28.3.2015