

## L07 Tehtävät

- Rakenteiset tietotyypit lista ja luokka/olio
- Listan operaatiot ja niiden käyttäminen
- Muotoiltu tulostus ja merkkijonon jako osiin, format ja split-jäsenfunktiot

Lue oppaan tämän viikon asioita käsittelevä luku 7. Lisäksi tehtävien suorittamiseen tarvitset aiempien lukujen tietoja. Ohjelmointitehtävät palautetaan Moodlen kautta CodeGradeen.

**Huom. Lisää jokaiseen CodeGradeen palauttamaasi tehtävään L07T1 -tehtävästä alkaen alla oleva alkukommentti ja täytä siihen tehtäväkohtaiset tiedot.** Ao. `otsikkotiedot.py` – tiedosto löytyy Moodlesta L07-viikkotehtävien tiedostojen yhteydestä.

```
#####
# CT60A0203 Ohjelmoinnin perusteet
# Tekijä:
# Opiskelijanumero:
# Päivämäärä:
# Kurssin oppimateriaalien lisäksi työhön ovat vaikuttaneet seuraavat
# lähteet ja henkilöt, ja se näkyy tehtävässä seuraavalla tavalla:
#
# Mahdollisen vilppiselvityksen varalta vakuutan, että olen tehnyt itse
# tämän tehtävän ja vain yllä mainitut henkilöt sekä lähteet ovat
# vaikuttaneet siihen yllä mainituilla tavoilla.
#####
```

L07T1: Valikkopohjainen ohjelma ostoslistan käsittelyyn .....	1
L07T2: Luokan määrittely ja olion käyttö .....	2
L07T3: Tiedoston rivillä olevien tietojen käsittely .....	3
L07T4: Valikkopohjainen ohjelma, aliohjelmat, luokka, olio ja lista .....	3
L07T5: Valikkopohjainen ohjelma / laskin listoilla, jatkoa L06T5:lle .....	4

### L07T1: Valikkopohjainen ohjelma ostoslistan käsittelyyn

Tee ohjelma, joka ylläpitää ostoslistaa. Ostoslistaan voi lisätä tuotteita sekä siitä voi poistaa tuotteita. Ohjelman tulee toimia siten, että käyttäjä syöttää valikkoon tiedon haluaako hän lisätä tai poistaa tuotteen vai lopettaa. Mikäli käyttäjä haluaa lisätä jotain, ohjelma pyytää tuotteen nimen ja lisää sen listalle. Mikäli taas käyttäjä poistaa jotain, pyydetään poistettavan tuotteen järjestysnumeroa. "Lopeta" valinta kertoo, mitä käyttäjältä jäi ostamatta ja päättää ohjelman. Ostoslistan toteutus tehdään listana ja ohjelma tulostaa jokaisen toimenpiteen jälkeen listan uudelleen.

Ohjelmassa tulee olla paaohjelma() ja valikon käsittelyyn kannattaa käyttää omaa aliohjelmaa. Mikäli valintaa ei tunnisteta, kertoo ohjelma siitä käyttäjälle ilmoituksella ”Tunnistamaton valinta.”

#### Ohjelman esimerkkiajo:

Ostoslistasi sisältää seuraavat tuotteet:

```
[ ]
Voit valita alla olevista toiminnoista:
1) Lisää
2) Poista
0) Lopeta
Valintasi: 1
Anna lisättävä tuote: Maito

Ostoslistasi sisältää seuraavat tuotteet:
['Maito']
Voit valita alla olevista toiminnoista:
1) Lisää
2) Poista
0) Lopeta
Valintasi: 1
Anna lisättävä tuote: Piimä

Ostoslistasi sisältää seuraavat tuotteet:
['Maito', 'Piimä']
Voit valita alla olevista toiminnoista:
1) Lisää
2) Poista
0) Lopeta
Valintasi: 2
Ostoslistassasi on 2 tuotetta.
Anna poistettavan tuotteen järjestysnumero: 1

Ostoslistasi sisältää seuraavat tuotteet:
['Piimä']
Voit valita alla olevista toiminnoista:
1) Lisää
2) Poista
0) Lopeta
Valintasi: 0
Sinulta jäi ostamatta seuraavat tuotteet:
['Piimä']
Kiitos ohjelman käytöstä.
```

## L07T2: Luokan määrittely ja olion käyttö

Tee ohjelma, jossa on luokka ja demonstroi luokan käyttöä kysymällä tietoja sekä tulostamalla ne käyttäen hyväksi aliohjelmia.

1. Määrittele luokka, jossa on *auton merkki* sekä *autojen lukumäärä* –jäsenmuuttujat.
2. Luo paaohjelma():ssa luokasta olio, kutsu tiedot kysyvää aliohjelmaa ja välitä sille parametrina em. olio. Palauta olio paluuarvona paaohjelma():aan.
3. Välitä olio tulostus-aliohjelmaan, joka tulostaa sen sisältämien jäsenmuuttujien arvot.

Noudata tulosteissa esimerkkiajon muotoa.

### Ohjelman esimerkkiajo:

```
Anna automerkki: Fiat
Anna automerkin lukumäärä varastossa: 3
Varastossa on nyt Fiat-merkkisiä autoja 3 kpl.
Kiitos ohjelman käytöstä.
```

### L07T3: Tiedoston rivillä olevien tietojen käsittely

Data-tiedostoissa on useita tietoalkioita samalla rivillä erotettuna erotin-merkillä. Tee ohjelma, joka

1. Lukee tiedoston rivi kerrallaan, tiedoston formaatti näkyy alla.
2. Jakaa rivin erotinmerkeillä (;) tietoalkioihin ohjelmointioppaan luvun 6 mukaisesti **split**-käskyllä.
3. Tulostaa käyttäjälle kellonajan ja punnitun marjan nimen alla olevan esimerkkiajon mukaisesti.

Huomaa, että tiedostossa on otsikkorivi, se kannattaa lukea ensin pois ja käydä sen jälkeen kaikki data-rivit läpi samalla tavalla. Testaa ohjelmasi tiedostoilla **L07T3D1.txt** ja **L07T3D2.txt**. Toteuta ohjelma paaohjelma():na ja kaikki tulosteet voivat olla merkkijonoja.

#### Ohjelman lukeman tiedoston muutama ensimmäistä riviä:

```
Marja;Paino(g);Kellonaika  
vadelma;234.0;00:00  
sudenmarja;40.5;08:05
```

#### Ohjelman esimerkkiajo:

```
Anna tiedoston nimi: L07T3D2.txt  
Kello oli 00:00, kun punnittiin marjoja vadelma.  
Kello oli 08:05, kun punnittiin marjoja sudenmarja.  
Kello oli 11:43, kun punnittiin marjoja lakka.  
Kello oli 13:15, kun punnittiin marjoja pensasmustikka.  
Kello oli 23:59, kun punnittiin marjoja karpalo.  
Kello oli 15:03, kun punnittiin marjoja karviainen.  
Kello oli 11:11, kun punnittiin marjoja viherherukka.  
Kello oli 12:19, kun punnittiin marjoja banaani.  
Kello oli 5:15, kun punnittiin marjoja papaija.  
Kello oli 20:12, kun punnittiin marjoja tomaatti.  
Kiitos ohjelman käytöstä.
```

### L07T4: Valikkopohjainen ohjelma, aliohjelmat, luokka, olio ja lista

Tee valikkopohjainen ohjelma, joka pystyy kysymään käyttäjältä automerkkejä ja niiden myyntihintoja, tulostamaan kaikki annetut tiedot näytölle sekä lopettamaan ohjelman suorittamisen. Määrittele ohjelmaan luokka, jossa on kaksi jäsenmuuttujaa yhden auton merkkiä (merkkijono) ja hintaa (kokonaisluku) varten. Käyttäjän antaessa auton tietoja, välitä lista parametrinä aliohjelmaan, luo uusi olio ja laita käyttäjän antamat tiedot siihen ja lisää olio listaan sekä palauta lista paluuarvona kutsuvaan ohjelmaan. Käyttäjän valitessa tietojen tulostuksen, välitä lista aliohjelmaan ja tulosta tiedot siellä. Kun käyttäjä valitsee lopetoiminnon, ohjelma tyhjentää listan ennen suorituksen päättymistä. Mikäli ohjelma ei tunnista käyttäjän antamaa valintaa, ilmoittaa se käyttäjälle tästä ilmoituksella "Tuntematon valinta, yritä uudestaan."

Tietojen tulostuksessa kannattaa harjoitella format-funktion käyttöä, sillä se tarjoaa hyviä työkaluja tulosteiden muotoiluun niin kuin luennoilla oli puhetta. Tämän tehtävän tulosteet pystyy kuitenkin tekemään ihan hyvin aiemmin läpikäydyillä tulostustavoilla.

#### Ohjelman esimerkkiajo:

```
Tämä ohjelma lisää autojen tietoja listaan ja tulostaa ne.  
Käytettävissä olevat toiminnot:
```

```
1) Anna auton tiedot
2) Tulosta autojen tiedot
0) Lopeta
Valintasi: 1
Anna auton merkki: Audi
Anna auton hinta: 12345

Käytettävissä olevat toiminnot:
1) Anna auton tiedot
2) Tulosta autojen tiedot
0) Lopeta
Valintasi: 2
Listalta löytyy seuraavat automerkit ja hinnat:
Audi 12345

Käytettävissä olevat toiminnot:
1) Anna auton tiedot
2) Tulosta autojen tiedot
0) Lopeta
Valintasi: 0
Kiitos ohjelman käytöstä.
```

### **L07T5: Valikkopohjainen ohjelma / laskin listoilla, jatkoa L06T5:lle**

Tämä tehtävä laajentaa aiemmin tehtyä laskinta, jossa valikkopohjaisen laskimen valintarakenne tehtiin tehtävässä L03T3, toistorakenne L04T5, aliohjelmarakenne L05T5 ja tiedostonkäsittely L06T5. Lisää L06T5 ohjelmaan kaksi listaa, joista yhteen luetaan kaikki tiedostosta olevat luvut ja toiseen listaan lisätään kaikki tulokset ohjelman aikana. Tämän tehtävän lähtökohtana kannattaa käyttää itse tekemääsi ohjelmaa L06T5.

Tällä kertaa pitää muuttaa pääohjelmaa ja luvut tiedostosta lukevaa aliohjelmaa sekä lisätä tulokset tiedostoon tallentava uusi aliohjelma. Pääohjelmassa kysytään nyt luettavan ja kirjoittavan tiedoston nimet sekä määritellään kaksi listaa. Kun käyttäjä antaa ohjelmalle luvut laskentaa varten ensimmäistä kertaa, ohjelma lukee kaikki tiedostossa olevat luvut, laittaa ne listaan ja sijoittaa kahdella ensimmäisellä rivillä olleet luvut laskettaviksi luvuiksi. Jatkossa käyttäjän pyytäessä uusia lukuja tiedostoa ei enää lueta vaan seuraavat luvut ovat listan kaksi seuraavaa alkioita niin kauan kuin listassa on lukuja. Voimme olettaa, että tiedostossa on parillinen määrä lukuja, jolloin listasta saadaan luettua aina kaksi lukua. Ohjelman pitää tietää listalla olevien lukujen määrä sekä seuraavan luvun indeksi listalla, jotta ohjelma pystyy käymään läpi kaikki listan alkiot ja osaa lopettaa sopivalla hetkellä.

Laskutoimituksen jälkeen tulos laitetaan tulos-listaan merkkijonona. Kun käyttäjä lopettaa ohjelman, talleta tämä lista omassa aliohjelmassa tiedostoon, jonka jälkeen molemmat listat voidaan tyhjentää ja ohjelma lopettaa normaalisti. Huomaa, että edellisen viikon tehtävästä poiketen voimme nyt listojen avulla toteuttaa kaikki tiedostoihin liittyvät toimenpiteet aina yhdessä luku- tai kirjoitusaliohjelmassa tiedostonimen ja listan avulla.

Moodlessa on tiedostot **L06T5D1.txt** ja **L06T5D2.txt**, joissa on erilaiset syötteet ohjelmalle ja ohjelman esimerkkitalenne **L06T5T1.txt**. Molempien tiedostojen muoto on sama kuin L06.

#### **Luettavan tiedoston muoto, **L06T5D2.txt**:**

1  
2

#### **Kirjoitettavan tiedoston muoto, **L06T5T1.txt**:**

Summa 2 + 8 = 10

Osamäärä  $1 / 3 = 0.33$

Summa  $2 + 2 = 4$

### Ohjelman esimerkkiajo:

```
Anna luettavan tiedoston nimi: L06T5D2.txt
Anna kirjoitettavan tiedoston nimi: L06T5T1.txt
Tämä laskin osaa seuraavat toiminnot:
1) Anna luvut
2) Summa
3) Osamäärä
0) Lopeta
Valitse toiminto (0-3): 1
Luettu tiedosto 'L06T5D2.txt'.
Luettiin luvut 1 ja 2
Tämä laskin osaa seuraavat toiminnot:
1) Anna luvut
2) Summa
3) Osamäärä
0) Lopeta
Valitse toiminto (0-3): 2
Tulos lisätty listaan.
Tämä laskin osaa seuraavat toiminnot:
1) Anna luvut
2) Summa
3) Osamäärä
0) Lopeta
Valitse toiminto (0-3): 1
Luvut loppuivat, lopeta ohjelma.
Tämä laskin osaa seuraavat toiminnot:
1) Anna luvut
2) Summa
3) Osamäärä
0) Lopeta
Valitse toiminto (0-3): 0
Tallennettu tiedosto 'L06T5T1.txt'.
Kiitos ohjelman käytöstä.
```