L08 Tehtävät

- Valmiiden aliohjelmakirjastojen käyttö, erityisesti math, random ja datetime
- Omien aliohjelmakirjastojen teko ja käyttö
- Tehtävä 4 on laajeneva esimerkki aiemmilta viikoilta
- Tehtävän 5 rakenne vastaa harjoitustyön rakennetta ja Viopen hyväksymät palautukset tarkastetaan rakenteen osalta assistenttien toimesta, ks. tarkemmin alla
- **Kaikissa tehtävissä pitää olla mukana otsikkotiedot, ks. L07 tehtäväksianto. Otsikkotietojen puuttuminen voi johtaa tehtävän hylkäämiseen.**

Lue oppaan tämän viikon asioita käsittelevä luku 8. Lisäksi tehtävien suorittamiseen tarvitset aiempien lukujen tietoja. Ohjelmointitehtävät palautetaan Moodlen kautta CodeGradeen. Jokaiselle tehtävälle on oma linkki Moodlessa.

L08T1: Python-kirjastojen käyttö, math ja random 1

L08T2: Oman kirjaston tekeminen ja käyttö 2

L08T3: Python-kirjastojen käyttö, datetime 4

L08T4: Valikkopohjainen ohjelma / laskin kirjastolla, jatkoa 5

L08T5: Useasta tiedostosta muodostuva perusohjelma 7

L08T1: Python-kirjastojen käyttö, math ja random

Tee ohjelma, joka hyödyntää Pythonin eri kirjastoja eri aliohjelmissa:

1. **Ympyrän pinta-alan laskenta `math`-kirjastolla**. Ohjelma kysyy käyttäjältä ympyrän säteen ja laskee sen jälkeen ympyrän pinta-alan käyttäen `math`-kirjastosta löytyvää piin arvoa (`pi`) sekä potenssilasku-funktiota `pow`. Tulosta lopuksi pinta-ala pyöristettynä 2 desimaalin tarkkuudella.
2. **Arvotun luvun arvaaminen ja `random`-kirjasto**. Ohjelma arpoo luvun väliltä 0-1000 `random`-kirjaston `randint`-funktiolla ko. arvot mukaan lukien. Sen jälkeen ohjelma kysyy käyttäjältä arvausta ja kertoo, onko haettu luku suurempi vai pienempi niin kauan kunnes käyttäjä arvaa luvun oikein. Onnistumisesta tiedottamisen lisäksi ohjelma kertoo, montako arvausta käyttäjä tarvitsi oikean luvun löytämiseen. Muista, että nyt käytetään pseudosatunnaislukuja eli ne lasketaan lähtien liikkeelle siemenarvosta. Jotta ohjelman testaaminen olisi mahdollista, aseta ohjelmassasi tämä siemenluku pääohjelmassa `random.seed(1)` –käskyllä.

Katso alla olevasta esimerkkiajosta ohjelman tarkempi toiminta. Toteuta molemmat toiminnot omina aliohjelminaan, noin 5-15 riviä per aliohjelma.

Ohjelman esimerkkiajo:

```
Mitä haluat tehdä:
1) Laskea ympyrän pinta-alan
2) Arvata luvun
0) Lopeta
Valintasi: 1
Anna ympyrän säde kokonaislukuna: 7
Säteellä 7 ympyrän pinta-ala on 153.94.

Mitä haluat tehdä:
1) Laskea ympyrän pinta-alan
2) Arvata luvun
0) Lopeta
Valintasi: 2
Arvaa ohjelman arpoma kokonaisluku.
Anna kokonaisluku välillä 0-1000: 500
Haettu luku on pienempi.
Anna kokonaisluku välillä 0-1000: 250
Haettu luku on pienempi.
Anna kokonaisluku välillä 0-1000: 125
Haettu luku on suurempi.
Anna kokonaisluku välillä 0-1000: 150
Haettu luku on pienempi.
Anna kokonaisluku välillä 0-1000: 137
Oikein! Käytit arvaamiseen 5 kierrosta.

Mitä haluat tehdä:
1) Laskea ympyrän pinta-alan
2) Arvata luvun
0) Lopeta
Valintasi: 0
Kiitos ohjelman käytöstä.
```

L08T2: Oman kirjaston tekeminen ja käyttö

Tee Python-ohjelma, joka muuntaa lämpötiloja Fahrenheit, Kelvin ja Celsius lämpötila-asteikkojen välillä. Esimerkiksi 0 Celcius astetta on Kelvineissä 273.15 astetta ja näille kaikille muunnoksille on olemassa yksinkertaiset kaavat (ks. esim. Wikipedia). Ohjelmoi kukin lämpötilamuunnos omaksi aliohjelmaksi, joka saa parametrina muunnettavan lämpötilan ja palauttaa paluuarvona liukulukutuloksen. Laita nämä muunnosfunktiot omaan tiedostoon aliohjelmakirjastoksi ja lisää kirjastoon kiintoarvona kirjaston versionumero, nyt 1.0.

Tee toinen tiedosto, jossa on pääohjelma eli alla olevan esimerkkiohjelman mukainen valikko sekä siihen liittyvä valintarakenne. Käyttäjä aloittaa valitsemalla haluamansa muunnoksen ja antaa muunnettavan lämpötilan kokonaislukuna, jonka jälkeen pääohjelma kutsuu kirjastofunktiota suorittamaan tämän muunnoksen sekä tulostaa tuloksen näytölle pyöristettynä kahden desimaalin tarkkuuteen.

Kirjastojen kanssa on oleellista, että rajanpinta ja toiminnallisuus on ymmärretty oikein. Tässä ohjelmassa on muistettava, että jokainen funktio saa parametrina kokonaisluvun ja palauttaa muunnoksen jälkeen liukuluvun pyöristämättä sitä. Koska kyseessä on lämpötilamuunnoskirjasto, on kirjastossa vain ko. muunnosfunktiot sekä kirjaston version kertova numero (kiintoarvo). Pääohjelma ja valikko muodostavat tässä tehtävässä ohjelman käyttöliittymän ja siksi ne ovat toisessa tiedostossa.

Palauta CodeGradeen kaksi tiedostoa, pääohjelma tiedostossa L08T2.py ja kirjasto tiedostossa L08T2Kirjasto.py.

Ohjelman esimerkkiajo:

```
Käytetään lämpötilamuunnoskirjaston versiota 1.0
Minkä lämpötilamuunnoksen haluat tehdä?
1) Celsius->Fahrenheit
2) Celsius->Kelvin
3) Fahrenheit->Kelvin
4) Fahrenheit->Celsius
5) Kelvin->Celsius
6) Kelvin->Fahrenheit
0) Lopeta
Valintasi: 1
Anna lähtölämpötila: 0
Lämpötila Fahrenheit asteina: 32.0

Minkä lämpötilamuunnoksen haluat tehdä?
1) Celsius->Fahrenheit
2) Celsius->Kelvin
3) Fahrenheit->Kelvin
4) Fahrenheit->Celsius
5) Kelvin->Celsius
6) Kelvin->Fahrenheit
0) Lopeta
Valintasi: 4
Anna lähtölämpötila: 80
Lämpötila Celsius asteina: 26.67

Minkä lämpötilamuunnoksen haluat tehdä?
1) Celsius->Fahrenheit
2) Celsius->Kelvin
3) Fahrenheit->Kelvin
4) Fahrenheit->Celsius
5) Kelvin->Celsius
6) Kelvin->Fahrenheit
0) Lopeta
Valintasi: 6
Anna lähtölämpötila: 293
Lämpötila Fahrenheit asteina: 67.73

Minkä lämpötilamuunnoksen haluat tehdä?
1) Celsius->Fahrenheit
2) Celsius->Kelvin
3) Fahrenheit->Kelvin
4) Fahrenheit->Celsius
5) Kelvin->Celsius
6) Kelvin->Fahrenheit
0) Lopeta
Valintasi: 0
Kiitos ohjelman käytöstä.
```

L08T3: Python-kirjastojen käyttö, datetime

Harjoittele Pythonin `datetime` kirjaston käyttöä toteuttamalla seuraavat toiminnot omissa aliohjelmissaan ja tee näitä kutsuva pääohjelma. Pääohjelmassa on sama rakenne kuin edellisissä tehtävissä eli niitä kannattaa käyttää hyväksi.

1. **`datetime`-olion jäsenmuuttujat**. Ohjelma pyytää käyttäjältä päivämäärän ja kellonajan merkkijonona formaatissa ”pp.kk.vvvv hh:mm” ja muuttaa merkkijonon `datetime`-olioksi. Tästä oliosta on helppo käyttää jäsenmuuttujia – nyt päivämäärä, kuukausi, vuosi, tunnit ja minuutit – ja tulostaa ne näytölle esimerkkiajon mukaisesti.
2. **Ajanjakson pituuden laskenta**. Ohjelma kysyy käyttäjältä tämän syntymäajan muodossa pp.kk.vvvv ja laskee kuinka vanha tämä oli 1.1.2000 sekä tulostaa vastauksen päivinä. Käyttämällä `datetime`-olioita voit laskea iän erotuksena.
3. **Viikonpäivien nimien tulostaminen**. `strftime`-funktiolla voi tulostaa `datetime`-olion sisältämiä tietoja monipuolisesti. Käytä tätä funktiota hyväksi ja tulosta viikonpäivien nimet näytölle. Aloita luomalla `datetime`-olio käyttäen maanantain päivämäärää, mikä tahansa maanantain päivämäärä käy. Käy sen jälkeen yhden viikon kaikki päivät läpi toistorakenteella ja siirtymällä tulostuksessa seuraavaan päivään `timedelta`-jäsenfunktiolla. Huomaa, että tietokoneen asetuksissa olevat kieliasetukset saattavat vaikuttaa tulostuksen kieleen, mutta Viopessa viikonpäivät tulostuvat Englanniksi `strftime`-funktiolla.
4. **Yhden vuoden kuukausien nimien tulostaminen lyhenteinä**. Tämä tehtävä onnistuu edellisen kohdan ideoilla, mutta mieti tähän tehtävään sopivat lähtökohta ja siirtymä. Tehtävä on tulostaa 12 kuukauden nimet oikein eli siirtymän pitää osua aina seuraavalle kuukaudelle, muttei välttämättä samalle päivälle.

Katso alla olevasta esimerkkiajosta ohjelman tarkempi toiminta ja ohjelmointioppaassa on kerrottu tärkeimmät `datetime`-moduulin toiminnallisuudet. Toteuta kukin yllä olevista toiminnoista omana aliohjelmanaan, tyypillisesti alle 10 riviä per aliohjelma.

Ohjelman esimerkkiajo:

```
Tämä ohjelma käyttää datetime-kirjastoa tehtävien ratkaisemiseen.
Mitä haluat tehdä:
1) Tunnistaa aika-olion komponentit
2) Laskea iän päivinä
3) Tulostaa viikonpäivät
4) Tulostaa kuukaudet
0) Lopeta
Valintasi: 1
Anna päivämäärä ja kello muodossa 'pp.kk.vvvv hh:mm': 24.12.2012 20:13
Annoit vuoden 2012
Annoit kuukauden 12
Annoit päivän 24
Annoit tunnin 20
Annoit minuutin 13

Mitä haluat tehdä:
1) Tunnistaa aika-olion komponentit
2) Laskea iän päivinä
3) Tulostaa viikonpäivät
4) Tulostaa kuukaudet
0) Lopeta
Valintasi: 2
Anna syntymäpäiväsi muodossa pp.kk.vvvv: 1.1.1999
1.1.2000 sinä olit 365 päivää vanha.
```

```
Mitä haluat tehdä:
1) Tunnistaa aika-olion komponentit
2) Laskea iän päivinä
3) Tulostaa viikonpäivät
4) Tulostaa kuukaudet
0) Lopeta
Valintasi: 3
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Mitä haluat tehdä:
1) Tunnistaa aika-olion komponentit
2) Laskea iän päivinä
3) Tulostaa viikonpäivät
4) Tulostaa kuukaudet
0) Lopeta
Valintasi: 4
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Mitä haluat tehdä:
1) Tunnistaa aika-olion komponentit
2) Laskea iän päivinä
3) Tulostaa viikonpäivät
4) Tulostaa kuukaudet
0) Lopeta
Valintasi: 0
Kiitos ohjelman käytöstä.
```

L08T4: Valikkopohjainen ohjelma / laskin kirjastolla, jatkoa

Tämä tehtävä laajentaa aiemmin tehtyä laskinta, jossa valikkopohjaisen laskimen valintarakenne tehtiin tehtävässä L03T3, toistorakenne L04T5, aliohjelmarakenne L05T5, tiedostonkäsittely L06T5 ja lista-ratkaisu L07T5. Muokkaa L07T5 ohjelmaa jakamalla koodi kahteen tiedostoon siten, että valikko ja pääohjelma ovat yhdessä tiedostossa ja kaikki muut aliohjelmat kirjastossa. Tämän tehtävän lähtökohtana kannattaa käyttää itse tekemääsi ohjelmaa L07T5.

Toimivan ohjelman jako useisiin tiedostoihin ei edellytä juurikaan uutta koodia, vaikka luonnollisesti jotain on muutettava ohjelman rakenteen muutoksen yhteydessä. Jaa toimiva ohjelma ensin ohjeiden mukaan kahteen eri tiedostoon ja nimeä ne L08T4.py sekä

L08T4Kirjasto.py. Tee sen jälkeen ohjelmakoodiin tarvittavat muutokset oppimateriaalien esimerkkien mukaisesti. Idea on, että useaan tiedostoon jaettu ohjelma toimii samalla tavalla kuin aiemmin, joten esimerkkiajot ja tiedostot ovat samoja kuin edellisellä kerralla, vaikka ohjelman sisäinen toteutus on erilainen.

Tämä ohjelma lukee ja kirjoittaa samoja tiedostoja kuin ohjelmat L06T5 ja L07T5, vaikka tallennettavan tiedoston nimi muuttuukin ja on L08T4T.txt. Moodlessa on testausta varten tiedostot L06T5D1.txt ja L06T5D2.txt.

Luettavan tiedoston muoto, L06T5D2.txt:

```
1
2
```

Kirjoitettavan tiedoston muoto, L08T4T.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: L08T4T1.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 'L08T4T1.txt'.
Kiitos ohjelman käytöstä.
```

L08T5: Useasta tiedostosta muodostuva perusohjelma

Tee valikkopohjainen ohjelma, joka pystyy lukemaan tiedoston, analysoimaan luetut tiedot, tallentamaan tulokset tiedostoon sekä lopettamaan ohjelman suorituksen. Tässä tehtävässä harjoitellaan itse tehdyn kirjaston käyttämistä, joten laita pääohjelma valikon kanssa yhteen tiedostoon ja kaikki muut aliohjelmat toiseen tiedostoon eli kirjastoon.

Pääohjelma on minimaalinen ja keskittyy ikisilmukan pyörittämiseen sekä kutsuu kirjaston aliohjelmia käyttäjän toiveiden mukaisesti. Koska tieto kulkee ohjelmien välillä parametreina ja paluuarvoina, määrittele pääohjelmaan kaksi listaa, yksi tiedostosta luettuja rivejä varten ja toinen analysoitujen tietojen tallettamista varten. Kysy tiedostojen nimet pääohjelmassa käyttäjältä ja välitä ne parametreina aliohjelmiin.

Kirjastossa tulee määritellä luokka tietojen varten ja aliohjelmat toimintojen tekemiseen. Tässä tehtävässä tärkeintä on ohjelman järkevä rakenne, joten pidä aliohjelmat lyhyinä ja niiden rooli on varmistaa ohjelman järkevä eteneminen; yksittäisiä aliohjelmia voidaan myöhemmin laajentaa ja kehittää. Alla on ohjelman lukema tiedosto, L08T5D1.txt, josta näkyy siinä olevat tiedot: tuotetunniste, tuotteiden lukumäärä ja yhden tuotteen hinta. Tee ohjelmaan luokka, jossa on jäsenmuuttuja jokaista tietoa varten ja laita aina yhden rivin tiedot yhden olion jäsenmuuttujiin ja olio listaan. Tietojen analyysi tarkoittaa varaston eri tuotteiden arvojen laskemista tuloslistaan niin, että jokaisen tuotteen varaston arvo laitetaan tuloslistan alkioksi, ts. tuotteiden määrä kerrotaan niiden arvolla, ja analyysi-aliohjelman päätteeksi tulostetaan koko varaston arvo. Tietojen tallennusvaiheessa kaikki tuloslistassa olevat alkiot kirjoitetaan yksi alkio aina yhdelle riville ja rivinvaihto viimeisenä merkkinä. Tulosta kaikki desimaaliluvut aina kahden desimaalin tarkkuudella.

Katso valikon ja ilmoitusten esitysasut ohjelman esimerkkiajosta. Jaa ohjelma kahteen tiedostoon, pääohjelma ja valikko-aliohjelma tiedostoon L08T5.py ja muut tiedostoon L08T5Kirjasto.py.

Ohjelman lukeman tiedoston alku, esim. L08T5D1.txt

```
A01-128;3;12.30
A01-987;10;1.15
B12-1;1;123.45
B54-182;111;13.50
C21-321;32;321.10
```

Ohjelman kirjoittaman tiedoston alku, L08T5T1.txt

```
36.90
11.50
123.45
1498.50
10275.20
```

Ohjelman esimerkkiajo:

```
Anna luettavan tiedoston nimi: L08T5D1.txt
Anna kirjoitettavan tiedoston nimi: L08T5T1.txt
Mitä haluat tehdä:
1) Lue tiedosto
2) Analysoi tiedot
3) Tallenna Tulokset
0) Lopeta
Valintasi: 1
Tiedosto 'L08T5D1.txt' luettu, 5 riviä.

Mitä haluat tehdä:
1) Lue tiedosto
2) Analysoi tiedot
3) Tallenna Tulokset
0) Lopeta
Valintasi: 2
Tiedot analysoitu, varaston arvo on 11945.55 EUR.

Mitä haluat tehdä:
1) Lue tiedosto
2) Analysoi tiedot
3) Tallenna Tulokset
0) Lopeta
Valintasi: 3
Tulokset tallennettu tiedostoon 'L08T5T1.txt'.

Mitä haluat tehdä:
1) Lue tiedosto
2) Analysoi tiedot
3) Tallenna Tulokset
0) Lopeta
Valintasi: 0
Kiitos ohjelman käytöstä.
```