

Harjoitustyön perustason tehtävänanto

1. Tehtävä (sama teksti kuin Moodlessa).....	1
2. Ohjelman toiminnot	1
3. Luettavan tiedoston rakenne	1
4. Kirjoitettavan tiedoston rakenne	2
5. Ohjelman toteutuksen rakenteelliset vaatimukset.....	2
6. Toteutusta tarkentavat toiminnalliset vaatimukset.....	3
7. Esimerkkiajo	3
7.1. Virheilmoitukset	4
8. Tulostiedostosta Excelillä tehty esimerkkikuvaaja.....	4

1. Tehtävä (sama teksti kuin Moodlessa)

Tee Python-ohjelma, joka lukee yhden vuoden auringonpaistetiedot sisältävän tekstitiedoston, analysoi tiedot ja tallentaa tulokset tulostiedostoon. Paisteaikatiedot ovat Moodlen tiedostoissa tunnin välein ja ohjelma laskee näistä tallennettavat vuorokausisummat. Paisteaikatiedostoja on useilta eri vuosilta ja useilta Ilmatieteen laitoksen havaintoasemilta. Havaintoaseman nimi ja käsiteltävä vuosi kysytään käyttäjältä, jonka jälkeen paisteaikatiedot löytyvät <havaintoasema><vuosi>.txt –nimisistä tiedostoista. Luettavasta tiedostosta otetaan analysoitavaksi vain rivit, joiden päivämäärä on käyttäjän antamalta vuodelta. Tulokset kirjoitetaan tiedostoon, jonka nimi kysytään käyttäjältä. Ohjelma kirjoittaa tulostiedostoon ensimmäiselle riville otsikon ”Pvm” ja sen perään kaikki vuoden päivämäärät puolipisteillä (;) eroteltuina. Seuraavalle riville otsikoksi tulee havaintoaseman nimi ja kutakin päivämäärää vastaavalle kohdalle ko. päivän kumulatiivinen paisteikasumma minuutteina puolipisteillä eroteltuina.

Harjoitustyö palautetaan Moodlen Harjoitustyö-lehdelle, missä on selitetty myös harjoitustyön arvostelu. Viope tarkistaa ohjelman kulun ja sen kirjoittamien tekstitiedostojen sisällöt, jonka jälkeen assistentti tarkistaa ohjelman rakenteen. Harjoitustyö on henkilökohtainen tehtävä, joten jokainen kirjoittaa itse oman ohjelmansa.

2. Ohjelman toiminnot

Ohjelman valikko näkyy esimerkkiajossa, mutta idea on antaa ensin ohjelmalle käyttäjää kiinnostavan havaintoaseman nimi ja vuosi. Sen jälkeen käyttäjä voi valita tietojen lukemisen ja analysoida ne sekä tallentaa tulokset tiedostoon ohjelman lopettamisen lisäksi.

3. Luettavan tiedoston rakenne

Luettavan tiedoston rakenne näkyy alla. Päivämäärä ja kellonaika –kentät noudattavat L08 läpikäytyjä periaatteita, mutta paiste aika on muokattu Ilmatieteen laitoksen datasta. Alkuperäisessä datassa paiste aika ilmoitetaan sekunteina minuutissa, mutta harjoitustyössä datamäärän rajoittamiseksi paiste aika on sekunteja tunnissa.

Vuosi-Kk-Pv;Tunti:Minuutti;Paisteaika (s)

2018-01-01;00:00;0

2018-01-01;01:00;0

. . .

2018-07-14;12:00;3600

2018-07-14;13:00;1010

2018-07-14;14:00;0

2018-07-14;15:00;2255

2018-07-14;16:00;820

. . .

4. Kirjoitettavan tiedoston rakenne

Kirjoitettavan tiedoston rakenne näkyy alla. Ensimmäinen rivi sisältää sanan ”Pvm” ja vuoden jokaisen päivämäärän formaatissa pp.kk.vvvv. Arvot on eroteltu puolipistein ja viimeisen päivämäärän lopussa EI ole puolipistettä. Toisella rivillä on havaintoaseman nimi ja vuoden jokaista päivää vastaava kumulatiivinen paisteaikasumma minuutteina puolipisteillä eroteltuina eikä viimeisen paisteaajan jälkeen tule puolipistettä. Muuta paisteaikat kokonaisluvuiksi yksikkömuunnoksen jälkeen int()-käskyllä.

Pvm;01.01.2018;02.01.2018;03.01.2018;04.01.2018;05.01.2018;...

Kumpula;0;0;0;0;0;0;0;264;...

5. Ohjelman toteutuksen rakenteelliset vaatimukset

1. Ohjelmassa on oltava pääohjelma (paaohjelma()) ja aliohjelmiä. paaohjelma() –kutsu on ainoa päätasolla oleva ohjelmakäske. Kaikkien aliohjelmien tulee päättyä return-käskeyn. Mikäli aliohjelma ei palauta mitään, käytä None-avainsanaa.
2. Tiedonvälitys aliohjelmien välillä on toteutettava parametreilla ja paluuarvoilla. Globaalit muuttujat ovat kiellettyjä.
3. Tiedostonkäsittelyn yhteydessä tulee olla poikkeustenkäsittely aina tiedostoa avattaessa ja poikkeustilanteessa ohjelman suoritus lopetetaan sys.exit(0) komennolla kuten L09 tullaan puhumaan. Poikkeustenkäsittelyn on oltava tiedoston käsittelyn suorittavassa aliohjelmassa ja sen sisällä saa olla vain poikkeuksen kannalta oleelliset koodirivit.
4. Ohjelman toteutuksessa on oltava lista, luokka ja olio. Dynaamisia tietorakenteita eli listaa, tuplea tai sanakirjaa on käytettävä, jos samanlaisia muuttujia on yli 5 kpl. Luokassa tulee olla vähintään kolme jäsenmuuttujaa.
5. Ohjelmassa on hyödynnettävä itse tehtyä aliohjelmakirjastoa HT_kirjasto.py, jossa on oltava kaikki valikosta kutsuttavat aliohjelmat. Pääohjelmätiedostossa on ohjelman päävalikkorakenne eli paaohjelma(). Sekä pääohjelma- että kirjastotiedostoissa on oltava asiallisesti täytetyt alkukommentit.
6. Ohjelman toteutuksen tulee olla selkeä ja ohjelmakoodin on oltava ymmärrettävää, ylläpidettävää ja laajennettavaa.

6. Toteutusta tarkentavat toiminnalliset vaatimukset

1. Analysointi-aikaväliä tulostettaessa ensimmäinen ja viimeinen päivämäärä on selvitettävä tiedostossa olevasta datasta.
2. Jokainen käsiteltävä tiedosto tulee sulkea tiedostonkäsittelyn päätyttyä.
3. Tiedostoa kirjoittaessa mahdollinen aiempi tiedosto tuhotaan avaamisen yhteydessä. Kirjoitettavassa tiedostossa tietoalkiot erotetaan toisistaan puolipisteellä (;).
4. Kaikki laskenta tulee tehdä alkuperäisissä yksiköissä ja yksikkömuunnos tehdään viimeisenä vaiheena. Yksikkömuunnoksen jälkeen tulos muutetaan kokonaisluvuksi int()-käsityllä.
5. Kun tiedostossa oleva data luetaan listaan, on lista tyhjennettävä ennen datan lisäystä.
6. Datan analysointi tai tallennus ei ole mahdollista, jos lista on tyhjä. Tarkista aliohjelmissa, että tietoa tuovassa listassa on dataa ja jos näin ei ole, suorita paluu aliohjelmasta välittömästi, paluuarvona aliohjelman normaalisti palauttama lista tyhjänä virheen merkiksi.

7. Esimerkkiajo

Mitä haluat tehdä:

- 1) Anna havaintoasema ja vuosi
- 2) Lue säätilatiedosto
- 3) Analysoi päivittäiset säätilatiedot
- 4) Tallenna päivittäiset säätilatiedot
- 0) Lopeta

Valintasi: 1

Anna havaintoaseman nimi: Kumpula

Anna analysoitava vuosi: 2018

Mitä haluat tehdä:

- 1) Anna havaintoasema ja vuosi
- 2) Lue säätilatiedosto
- 3) Analysoi päivittäiset säätilatiedot
- 4) Tallenna päivittäiset säätilatiedot
- 0) Lopeta

Valintasi: 2

Tiedosto 'Kumpula2018.txt' luettu. Tiedostossa oli 8762 riviä.

Mitä haluat tehdä:

- 1) Anna havaintoasema ja vuosi
- 2) Lue säätilatiedosto
- 3) Analysoi päivittäiset säätilatiedot
- 4) Tallenna päivittäiset säätilatiedot
- 0) Lopeta

Valintasi: 3

Data analysoitu ajalta 01.01.2018 - 31.12.2018.

Mitä haluat tehdä:

- 1) Anna havaintoasema ja vuosi
- 2) Lue säätilatiedosto
- 3) Analysoi päivittäiset säätilatiedot
- 4) Tallenna päivittäiset säätilatiedot
- 0) Lopeta

Valintasi: 4

Anna tulostiedoston nimi: Kumpula2018_tulos.txt

Paisteaika tallennettu tiedostoon 'Kumpula2018_tulos.txt'.

Mitä haluat tehdä:

- 1) Anna havaintoasema ja vuosi
- 2) Lue säätilatiedosto
- 3) Analysoi päivittäiset säätilatiedot
- 4) Tallenna päivittäiset säätilatiedot
- 0) Lopeta

Valintasi: 0

Kiitos ohjelman käytöstä.

7.1. Virheilmoitukset

Esimerkkiajossa ei näy ohjelman virheilmoitukset, joten ne on koottu alle:

Anna valinta kokonaislukuna.

Anna vuosiluku kokonaislukuna.

Valintaa ei tunnistettu, yritä uudestaan.

Valitse havaintoasema ja vuosi ennen tiedostonlukua.

Lista on tyhjä. Lue ensin tiedosto.

Lista on tyhjä. Analysoi data ennen tallennusta.

Tiedoston '<tiedostonimi>' avaaminen epäonnistui.

Tiedoston '<tiedostonimi>' lukeminen epäonnistui.

Tiedoston '<tiedostonimi>' kirjoitus epäonnistui.

8. Tulostiedostosta Excelillä tehty esimerkkikuvaaja

Alla on ohjelman tulostiedostosta Kumpula2018_tulos.txt Excelillä piirretty kuvaaja. Käytetty datatiedosto on Kumpula2018.txt. Viope tarkistaa vain tehdyn tiedoston sisällön, joten tämän kuvaajan tekeminen jää oman mielenkiinnon ja harrastuksen varaan. Tuloksen hahmottamiseksi viikossa on 10080 minuuttia ($60 \cdot 24 \cdot 7 = 10080$) eli aurinko paistaa Kumpulassa yhteensä noin 12 viikkoa vuodessa.

