

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. Perustason toteutuksen rakenteelliset vaatimukset	2
6. Toteutusta tarkentavat toiminnalliset vaatimukset.....	3
7. Esimerkkiajo	3
8. Virheilmoitukset	4
9. Tuloksista Excelillä tehtyjä esimerkkikuvaajia	5

1. Tehtävä (sama teksti kuin Moodlessa)

Tee Python-ohjelma, joka lukee vuoden 2020 Ohjelmoinnin perusteet kurssin viikoittaisten ohjelmointitehtävien palautusten tiedot sisältävän tekstitiedoston, analysoi tiedot ja tallentaa tulokset tulostiedostoon. Palautuksista tiedetään aikaleima, palauttaja ja palautettu tehtävä, ja ohjelma selvittää näiden perusteella kunkin tehtävän palautusten määrän sekä seuraavat tilastotiedot: (1) Palautusten kokonaismäärä, (2) moneenko eri tehtävään tuli palautuksia, (3) keskimääräinen palautusmäärä tehtävää kohden, (4) mihin tehtävään tuli eniten ja (5) mihin vähiten palautuksia. Käyttäjä antaa tietoja sisältävän tiedoston nimen ja tulokset tulostetaan näytölle sekä kirjoitetaan tiedostoon, jonka nimi kysytään käyttäjältä. Tiedostojen rakenne on selitetty tarkemmissa ohjeissa.

Harjoitustyö palautetaan Moodlen Harjoitustyö-lehdelle, missä on selitetty myös harjoitustyön arvostelu. Viope tarkistaa ohjelman toiminnan ja sen kirjoittamien tekstitiedostojen sisällöt, jonka jälkeen assistentti tarkistaa ohjelman rakenteen. Ohjelman rakenteesta saa palautetta myös ASPA-ohjelmalla, ks. Moodlen ASPA-välilehti. Harjoitustyö on henkilökohtainen tehtävä, joten jokainen kirjoittaa itse oman ohjelmansa ja kaikki ohjelman vaikuttaneet henkilöt ja lähteet tulee mainita ohjelman alkukommenteissa.

2. Ohjelman toiminnot

Ohjelman valikko näkyy esimerkkiajossa, mutta käyttäjä voi valita siitä tietojen lukemisen, tietojen analysoimisen ja tulosten tallentamisen sekä ohjelman lopettamisen. Tiedoston lukemisen ja kirjoittamisen yhteydessä ko. tiedoston nimi kysytään käyttäjältä. Tiedoston lukeminen on käsitelty Kohdassa 3 ja kirjoittaminen Kohdassa 4. Lukemisen jälkeen tehtävässä analyysissä kannattaa ensin laskea tehtäväkohtaiset palautusmäärät käymällä kaikki palautukset läpi. Analyysin toisessa osassa lasketaan tilastotiedot käyttäen ensimmäisen vaiheen tuloksia.

3. Luettavan tiedoston rakenne

Luettavan tiedoston rakenne näkyy alla. Yksi rivi vastaa yhden opiskelijan palautusta yhteen viikkotehtävän. Aikaleimasarake, joka sisältää sekä päivämäärä että kellonaika -tiedot, noudat-

taa L08 läpikäytyjä periaatteita, ”Opiskelija”-sarake sisältää opiskelijan anonymisoidun tunnisteiden merkkijonona ja ”Tehtävä”-sarake sisältää tehtävän tunnisteiden kurssilla käytetyssä muodossa, esim. L05-T2 eli luento 5 ja tehtävä 2. Näistä tehtävätunniste on tärkein tieto analyysissä. Alla on esitetty luettavan tiedoston rakenne käyttäen Moodlesta ja Viopesta nimellä `palautukset15.txt` löytyvän tiedoston alkua ja laajempi kaikki tiedot sisältävä tiedosto on `palautukset.txt`. Tiedosto on järjestetty tehtävätunnisteiden mukaan siten, että alussa on järjestyksessä ensimmäinen tehtävä ja lopussa viimeinen, mutta välissä voi olla tehtäviä, joihin ei ole palautuksia, esim. *L01-T1, L01-T2, L03-T3* jne.

Aikaleima (dd-mm-YYYY HH:MM:SS);Opiskelija;Tehtävä

04-09-2020 12:24:12;Opiskelija_001;L01-T1

03-09-2020 10:42:10;Opiskelija_002;L01-T1

03-09-2020 15:05:10;Opiskelija_175;L01-T2

04-09-2020 15:08:00;Opiskelija_176;L01-T2

04-09-2020 13:52:25;Opiskelija_177;L01-T2

16-09-2020 05:21:15;Opiskelija_001;L03-T3

4. Kirjoitettavan tiedoston rakenne

Kirjoitettavan tiedoston rakenne näkyy alla. Muuta keskimääräinen palautusmäärä kokonaisluvuksi laskemisen jälkeen `int()`-käskyllä pyöristysongelmien välttämiseksi. Kirjoitettavan tiedoston alussa on tilastotiedot ja näiden alle listataan yhden tyhjän rivin jälkeen tehtäväkohtaiset palautusmäärät.

Palautuksia tuli yhteensä 15, 4 eri tehtävään.

Viikkotehtäviin tuli keskimäärin 3 palautusta.

Eniten palautuksia, 9, tuli viikkotehtävään L03-T3.

Vähiten palautuksia, 1, tuli viikkotehtävään L14-T3.

Tehtävä;Lukumäärä

L01-T1;2

L01-T2;3

L03-T3;9

L14-T3;1

5. Perustason toteutuksen rakenteelliset vaatimukset

1. Ohjelmassa on oltava pääohjelma (`paaohjelma()`) ja aliohjelmia. Pääohjelman kutsu on ainoa päätasolla oleva ohjelmakäskey. Kaikkien aliohjelmien tulee päättyä `return`-käskeyyn. 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 aina poikkeusten käsittely ja poikkeustilanteessa ohjelman suoritus lopetetaan `sys.exit(0)` komennolla L09 mukaisesti. Poikkeusten käsittelyn 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. Listan sijasta voi käyttää muita dynaamisia tietorakenteita, mutta niitä eli listaa tai sanakirjaa on käytettävä, jos samanlaisia muuttujia on yli 5 kpl. Luokassa tulee olla aina vähintään kaksi jäsenmuuttujaa ja luokkia voi olla useampia.
5. Ohjelmassa on hyödynnettävä itse tehtyä aliohjelmakirjastoja `HTPerusKirjasto.py`, jossa on oltava kaikki valikosta kutsuttavat aliohjelmat. Pääohjelmätiedostossa on ohjelman päävalikkorakenne eli `paaohjelma()` sekä valikko-aliohjelma. 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. Analysoituja palautuksia ja tehtäviä tulostettaessa lukumäärät on selvitettävä tiedostossa olevasta datasta. Kaikki tiedostossa olevat tiedot ovat samalta vuodelta.
2. Jokainen käsiteltävä tiedosto tulee sulkea tiedostonkäsittelyn päätyttyä.
3. Tiedostoa kirjoittaessa mahdollisesti olemassa olevan saman nimisen tiedoston tiedot häviävät, kun uusi tiedosto avataan.
4. Kaikki laskenta tulee tehdä alkuperäisissä yksiköissä ja muunnos kokonaisluvuksi tehdään `int()`-käskyllä.
5. Tiedoston lukemisen yhteydessä lista on tyhjennettävä ennen datan lisäystä siihen. Lukemisen päätteeksi ilmoitettava rivimäärä on luettujen datarivien määrä, ts. otsikkoriviä ei oteta mukaan.
6. Datan analysointi tai tulosten tallennus ei ole mahdollista, jos dataa ei ole. Tarkista pääohjelmassa, että luettu tai analysoitu lista sisältää dataa ja jos näin ei ole näytä käyttäjälle Kohdan 8 mukainen virheilmoitus.
7. Jos usealla tehtävällä on sama määrä eniten tai vähiten palautuksia, tulee ohjelman käyttää pienimmän ja suurimman palautusmäärän tulosteessa järjestyksessä aiemmin olevaa tehtävää.

7. Esimerkkiajo

Mitä haluat tehdä:

- 1) Lue tiedosto
- 2) Analysoi palautukset
- 3) Tallenna tulokset
- 0) Lopeta

Valintasi: 1

Anna luettavan tiedoston nimi: palautukset15.txt

Tiedostosta 'palautukset15.txt' luettiin listaan 15 datarivin tiedot.

Anna uusi valinta.

Mitä haluat tehdä:

- 1) Lue tiedosto
- 2) Analysoi palautukset
- 3) Tallenna tulokset
- 0) Lopeta

Valintasi: 2

Analysoitu 15 palautusta 4 eri tehtävään.

Tilastotiedot analysoitu.

Anna uusi valinta.

Mitä haluat tehdä:

- 1) Lue tiedosto
- 2) Analysoi palautukset
- 3) Tallenna tulokset
- 0) Lopeta

Valintasi: 3

Anna kirjoitettavan tiedoston nimi: tulos15.txt

Palautuksia tuli yhteensä 15, 4 eri tehtävään.

Viikkotehtäviin tuli keskimäärin 3 palautusta.

Eniten palautuksia, 9, tuli viikkotehtävään L03-T3.

Vähiten palautuksia, 1, tuli viikkotehtävään L14-T3.

Tehtävä;Lukumäärä

L01-T1;2

L01-T2;3

L03-T3;9

L14-T3;1

Tulokset tallennettu tiedostoon 'tulos15.txt'.

Anna uusi valinta.

Mitä haluat tehdä:

- 1) Lue tiedosto
- 2) Analysoi palautukset
- 3) Tallenna tulokset
- 0) Lopeta

Valintasi: 0

Kiitos ohjelman käytöstä.

8. Virheilmoitukset

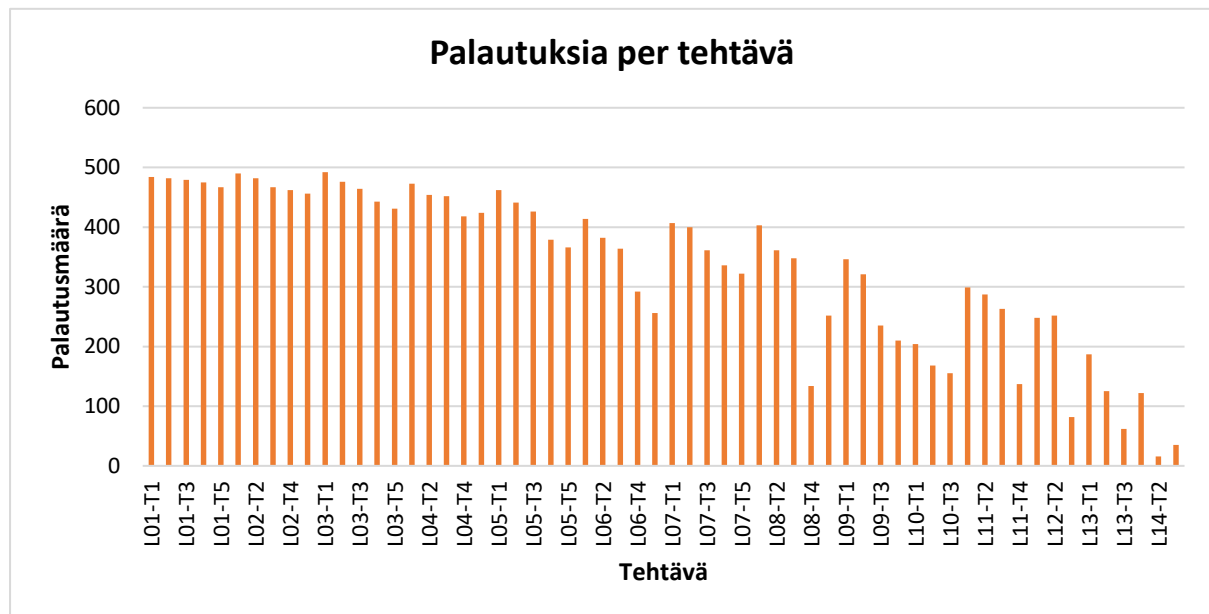
Esimerkkiajossa ei näy ohjelman virheilmoituksia, joten ne on koottu alle. Laita ohjelmaasi normaalit poikkeustenkäsittelijät tiedostonkäsittelyn yhteyteen ja tarkista muut asiat valintarakenteiden avulla.

- Valikon valinta: Tuntematon valinta, yritä uudestaan.
- Analyysin tarkistus: Ei tietoja analysoitavaksi, lue tiedot ennen analyysiä.

- Kirjoituksen tarkistus: Ei tietoja tallennettavaksi, analysoi tiedot ennen tallennusta.
- Tiedostonkäsitely: Tiedoston '<tiedostonimi>' käsittelyssä virhe, lopetetaan.

9. Tuloksista Excelillä tehtyjä esimerkkikuvaajia

Alla Kuvassa 1 on palautus.txt tiedostolle tehdyn analyysin tuloksista piirretty Excel-kaavio. Viope tarkistaa vain tehdyn tiedoston sisällön, joten kuvaajan tekeminen jää oman mielenkiinnon ja harrastuksen varaan.



Kuva 1. Tehtäväkohtaisen analyysin tulos palautukset.txt tiedostolle.