

CS-A1111 Ohjelmoinnin peruskurssi Y1

Tentti 1.12.2017

Kirjoita jokaisen vastauspaperisi alkuun kurssin nimi, kokeen päivämäärä, nimesi, opiskelijanumerosi, vastauspaperiesi kokonaismäärä sekä allekirjoituksesi.

Tärkeitä ohjeita vastausten kirjoittamiseen: Kun kirjoitat ohjelmakoodia, käytä kahden ruudun levyisiä sisennyksiä. Jos sisennyksiä ei ole käytetty tai niistä ei saa selvää, vähennetään siitä pisteitä. Kirjoitettavaan ohjelmakoodiin ei tarvitse lisätä kommentteja. Missään tehtävässä tulostusta ei tarvitse muotoilla. Voit myös olettaa, että käyttäjän antama syöte on virheetöntä, ellei tehtävässä erikseen käsketä käsittelemään virhetilanteita.

Tentissä ei saa käyttää laskimia eikä lisämateriaalia. Opiskelijat, joiden äidinkieli ei ole suomi, saavat kuitenkin käyttää sanakirjaa, jos siinä ei ole merkintöjä (tentin valvoja tarkistaa sanakirjan). Nämä opiskelijat saavat halutessaan myös sekä suomen- että englannin- tai ruotsinkielisen kysymyspaperin.

Vastaa myös kurssin palautekyselyyn. Kyselyyn vastaamisesta saa 200 harjoitustehtäväpistettä. Linkki kyselyyn on lähetetty kurssille ilmoittautuneille sähköpostitse.

1. a) Vastaa kurssin lopputestiin. Tämän tentin pisteistä 2 tulee lopputestiin vastaamisesta. Linkki testiin on syksyn kurssin Goblinin etusivulla. Myös kesäkurssille osallistuneet voivat kirjautua Aalto-tunnuksilla syksyn kurssin Gobliniin ja vastata siihen. Linkki syksyn kurssin Gobliniin on julkaistu myös kesäkurssin MyCourses-sivulla. Testi on auki 8.12. asti. (2 p)

b) Vastaa kurssin loppukyselyyn (eri kuin palautekysely) Goblinissa. Katso ohjeet a-kohdasta. (2 p)

Kohdissa b, c ja d kerro, mitä annettu ohjelma tulostaa. Vastausta ei tarvitse perustella. Kohdissa e, f ja g kerro, mitä tehtävässä esitetty funktio tekee. Älä selitä funktion toimintaa käsky käskyltä, vaan selitä parilla lauseella, mikä on funktion tarkoitus (esimerkiksi: "funktio laskee ja palauttaa parametrina annetussa listassa olevien lukujen summan"). Funktioille annettavien parametrien luonne on selitetty kunkin kohdan yhteydessä. Huomaa, että annetuissa ohjelmissa tai funktioissa voi olla myös virheitä. Kerro siinä tapauksessa, mitä annettu virheellinen ohjelma tulostaa tai miten virheellinen funktio toimii - ei siis sitä, miten ohjelman tai funktion pitäisi toimia, jos siinä ei olisi virheitä.

c) (3 p)

```
def main():
    lista = [20, 70, 40, 40, 80, 10]
    tulos = 100
    for luku in lista:
        if tulos > 50:
            tulos = tulos - luku
    print(tulos)
```

main()

d) (5 p)

```
def muuttele(arvo, lukulista):
    lukulista[0] = lukulista[0] - arvo
    arvo = 20
    return arvo
```

```
def main():
    tulos = 8
    arvo = 10
    luvut = [50, 30, 60]
    muuttele(tulos, luvut)
    print(tulos)
    for alkio in luvut:
        print(alkio)
```

main()

e) Funktiolle annetaan parametreina kaksi listaa, jotka sisältävät kokonaislukuja ja joilla on sama pituus. (4 p)

```
def mysteeri1(lista1, lista2):
    i = 0
    tulos = 0
    while i < len(lista1):
        if lista1[i] > lista2[i]:
            tulos = tulos + lista1[i]
        i += 1
    return tulos
```

f) Funktiolle annetaan parametrina merkkijono, joka sisältää vähintään yhden merkin. (4 p)

```
def mysteeri2(merkkijono1):
    pituus = len(merkkijono1)
    if pituus < 4:
        return merkkijono1 + ".txt"
    elif merkkijono1[pituus - 4: pituus] == ".doc":
        return merkkijono1[0 : pituus - 4] + ".txt"
    else:
        return merkkijono1
```

g) Funktiolle annetaan ensimmäisenä parametrina positiivisia kokonaislukuja sisältävä lista, jossa on vähintään kaksi alkioita, ja toisena parametrina positiivinen kokonaisluku (5 p)

```
def mysteeri3(lista, luku):
    i = 1
    while i < len(lista):
        if lista[i] < lista[i - 1] + luku:
            return False
        i += 1
    return True
```

2. a) Eräällä kuntosalilla asiakkaat voivat maksaa käyntinsä joko kertamaksulla tai kuukausimaksulla. Jos asiakas on valinnut kertamaksun, hänen pitää maksaa se jokaisesta käynnistä. Jos taas asiakas on maksanut kuukausimaksun, saa hän käydä maksun voimassaoloaikana salilla niin usein kuin hän haluaa ilman ylimääräisiä maksuja. Kuukausimaksun valinneen asiakkaan pitää kuitenkin sitoutua kuukausimaksuun seuraavaksi 12 kuukaudeksi. Kertamaksulla käyvät asiakkaat voivat halutessaan maksaa kuntosalin jäsenmaksun, joka on 60 eur vuodessa (sitä ei siis peritä kuukausimaksun maksaneilta asiakkailta). Jäsenmaksun maksaneet asiakkaat saavat kertamaksusta 20 %:n alennuksen. Kirjoita Python-ohjelma, joka kysyy käyttäjältä, montako kertaa hän aikoo käydä kuntosalilla seuraavan 12 kuukauden aikana. Ohjelman pitää myös kysyä kertamaksun ja kuukausimaksun hinnat. Tämän jälkeen ohjelma tutkii ja tulostaa, onko käyttäjän edullisinta käydä kuntosalilla maksamalla kertamaksu ilman jäsenkorttia, maksamalla kertamaksu jäsenkortin kanssa vai maksamalla kuukausimaksu. (10 p.)

b) Opiskelija voi suorittaa erään kurssin tekemällä tarpeeksi harjoitustehtäviä. Kukin harjoitustehtävä pisteytetään, ja jos opiskelijan tehtävästä saama pistemäärä ovat vähintään kurssilla määrätty raja, tehtävä on hyväksytty. Jos jonkin harjoitustehtävän pistemäärä jää alle vaaditun rajan, tämä tehtävä on hylätty. Raja on sama kaikille harjoitustehtäville. Eri opiskelijoiden palauttamien harjoitustehtävien määrä vaihtelee. Hyväksytyyn kurssisuoritukseen vaaditaan se, että opiskelijalla on vähintään 5 hyväksyttyä harjoitustehtävää ja että hyväksytyjä harjoitustehtäviä on vähintään kaksi kertaa niin paljon kuin hylättyjä harjoitustehtäviä (molempien ehtojen pitää siis täytyä). Kirjoita Python-funktio `onko_hyvaksyttu(tehtavapisteet, raja)`. Funktio saa ensimmäisenä parametrina listan, joka sisältää opiskelijan eri harjoitustehtävistä saamat pisteet (kukin alkio on yhden harjoitustehtävän pistemäärä). Funktion toinen parametri on edellä mainittu harjoitustehtävän hyväksymisraja. Funktion pitää palauttaa arvo True, jos opiskelijan kurssisuoritus on hyväksytty, ja False, jos opiskelijan kurssisuoritus on hylätty. Tehtävässä ei tarvitse kirjoittaa muuta kuin pyydetty funktio. (20 p)

3. Eräs verkkokauppa on tallentanut tekstitiedostoon tiedot sen myymien tuotteiden tilauksista niin, että kullakin rivillä on tieto yhden tuotteen yhdestä tilauksesta, joka voi kuitenkin sisältää useita kappaleita samaa tuotetta. Rivillä on ensin tilauksen päivämäärä, sitten tilatun tuotteen tuotenumero (merkkijono), tilattu kappalemäärä ja tuotteen hinta tilauksessa. Eri tiedot on erotettu toisistaan pilkulla. Samaa tuotetta on voitu tilata useita eri kertoja ja sen kappalehinta voi olla eri tilauksissa eri. Tiedoston kolme peräkkäistä riviä voisivat näyttää esim. seuraavalta:

10.11.2017,T-112233,5,20.00

11.11.2017,S-45500,1,250.00

12.11.2017,T-112233,3,15.00

Kirjoita Python-ohjelma, joka kysyy käyttäjältä, mistä tiedostosta tilausten tiedot luetaan ja minkä tuotteen tietoja hän haluaa etsiä. Ohjelman pitää laskea halutun tuotteen tilausten kokonaisarvo tiedostossa (yhteensä kaikista ko. tuotteen tilauksista tuotteen kappalemäärä kertaa tuotteen hinta). Jos esimerkiksi yllä annetussa esimerkissä etsittäisiin tuotteen T-112233 tilausten kokonaisarvoa, olisi se 145.00, koska $5 * 20.00 + 3 * 15.00 = 145.00$. Jos tiedostossa ei ole lainkaan halutun tuotteen tilauksia, ohjelma tulostaa tilausten kokonaisarvoksi 0.0. (Desimaalien määrän tulostuksissa ei tarvitse olla sama kuin esimerkeissä.)

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä
- Tiedoston jollain rivillä kappalemäärän paikalla olevaa tekstiä ei voi tulkita kokonaisluvuksi tai kappalehinna paikalla olevaa tekstiä desimaaliluvuksi.

Näissä tapauksissa ohjelma ilmoittaa käyttäjälle, millainen virhe on sattunut (siis joko että tiedostoa ei pystytä lukemaan tai että rivillä on virheellinen luku), ja lopettaa toimintansa. Ohjelman ei tarvitse jatkaa rivien lukemista virheellisen rivin jälkeen eikä tulostaa sellaisessa tapauksessa mitään arvoa. Voit olettaa, että tiedoston jokaisella rivillä on täsmälleen neljä toisistaan pilkulla erotettua osaa. Ohjelman ei tarvitse osata käsitellä esimerkiksi sellaisia virhetilanteita, joissa rivi on tyhjä tai ei sisällä nimen lisäksi muuta tekstiä. (20 p)

VIIMEINEN TEHTÄVÄ ON SEURAAVALLA SIVULLA

4. Kirjoita Python-kielillä luokka `Liittyma` yhden prepaid-tyyppisen matkapuhelinliittymän tietojen käsittelyyn. (Prepaid-tyyppisissä liittymissä liittymälle ladataan rahaa etukäteen, ja puhelimella voi soittaa ja lähettää tekstiviestejä vain sen verran kuin mihin liittymälle ladatut rahat riittävät.)

`Liittyma`-oliolla on oltava seuraavat kentät:

- `__numero` matkapuhelinliittymän puhelinnumero (merkkijono).
- `__saldo` liittymän saldo tällä hetkellä eli liittymällä olevien rahojen määrä sentteinä (kokonaisluku). Saldo pienenee, kun soitetaan tai lähetetään tekstiviestejä. Sitä voi kasvattaa lataamalla liittymälle lisää rahaa.
- `__onko_suljettu` kentän arvo on `True`, jos liittymä on suljettu (siitä ei voi soittaa tai lähettää tekstiviestejä) ja muuten `False` (liittymää voi käyttää normaalisti).

Määrittele luokkaan seuraavat metodit. (Jos metodin kuvauksessa ei ole kerrottu mitään metodin palauttamasta arvosta, metodin ei tarvitse palauttaa mitään.)

- `__init__(self, puhelinnumero, alkusaldo)` luo uuden `Liittyma`-olion. Luotavan liittymän puhelinnumero ja liittymälle ladattava alkusaldo sentteinä annetaan parametreina. Jos viimeinen parametri on negatiivinen, alkusaldoksi asetetaan 0. Uusi liittymä ei ole suljettu.
- `kerro_saldo(self)` palauttaa liittymällä tällä hetkellä olevan saldon.
- `onko_suljettu(self)` palauttaa arvon `True`, jos liittymä on suljettu ja muuten arvon `False`.
- `avaa_liittyma(self)` avaa liittymän.
- `sulje_liittyma(self)` sulkee liittymän.
- `lataa_rahaa(self, lisays)` Lataa liittymälle parametrina annetun määrän rahaa eli kasvattaa liittymän saldoa parametrina annetulla määrällä, jos parametri on positiivinen. Jos parametri ei ole positiivinen, metodi ei tee mitään. Parametri on kokonaisluku ja sisältää lisättävän määrän sentteinä.
- `soita_puhelu(self, kesto, onko_lisamaksullinen)` "soittaa" liittymästä parametrina annetun keston (sekunneissa, kokonaisluku) mittaisen puhelun, jos liittymä ei ole suljettu ja liittymän saldo riittää puhelun maksamiseen. Jos liittymän saldo ei riitä koko puhelun maksamiseen mutta liittymä ei ole suljettu, metodi "soittaa" kuitenkin niin pitkän puhelun kuin mihin saldo riittää. Jokainen alkava tavallinen puhelu maksaa 5 senttiä sekä lisäksi puhelu maksaa yhden sentin jokaiselta täydeltä 10 sekunnilta. (Esimerkiksi 29 sekunnin pituinen puhelu maksaa 7 senttiä, mutta 30 sekuntia kestävä puhelu 8 senttiä). Metodin viimeisen parametrin arvo on `False`, jos kysymyksessä on tavallinen puhelu, ja `True`, jos kysymyksessä on puhelu lisämaksulliseen palvelunumeroon. Puhelusta lisämaksulliseen palvelunumeroon veloitetaan edellä kerrotun hinnan lisäksi aloituksesta 20 senttiä ja jokaisesta täydestä 10 sekunnista 30 senttiä. Metodi palauttaa puhelun todellisen keston (joka on siis parametrina annettua kestoa pienempi, jos saldo ei riitä haluttuun kestoan). Käytännössä "soittaminen" näkyy siten, että metodi pienentää liittymän saldoa puhelun hinnalla. Jos liittymä on suljettu tai saldo ei riitä edes lyhyen puhelun soittamiseen, metodi palauttaa arvon 0.
- `laheta_tekstiviesti(self)` "lähettää" liittymästä yhden tekstiviestin, jos liittymä ei ole suljettu ja liittymän saldo riittää tekstiviestin lähettämiseen. Yhden tekstiviestin hinta on 6 snt. Metodi palauttaa arvon `True`, jos tekstiviestin lähettäminen onnistuu ja muuten arvon `False`. Jos lähettäminen onnistuu, metodi myös vähentää liittymän saldoa tekstiviestin hinnalla.
- `__str__(self)` palauttaa merkkijonon, joka sisältää liittymän puhelinnumeron, saldon ja joko tekstin "liittyma on suljettu" tai "liittyma on käytössä" sen mukaan, onko liittymä suljettu vai ei.

Kirjoita lisäksi pääohjelma, joka luo kaksi `Liittyma`-oliota, lataa niistä ensiksi luodulle rahaa, soittaa sen jälkeen tällä liittymällä puhelun ja kertoo sen keston. Tämän jälkeen pääohjelman pitää lähettää samalla liittymällä tekstiviesti ja kertoa, onnistuiko viestin lähetys. Lopuksi ohjelman pitää sulkea toinen liittymä ja tulostaa molemmista liittymistä puhelinnumero, saldo ja tieto siitä, onko liittymä suljettu vai käytössä. Voit päättää liittymien alkutiedot sekä latauksessa ja puhelussa tarvittavat tiedot itse. Pääohjelman ei siis tarvitse kysyä mitään käyttäjältä. Voit kirjoittaa pääohjelman valintasi mukaan joko niin, että se on samassa moduulissa luokan kanssa tai sitten niin, että se on eri moduulissa. (25 p)