

CSE/Ako-A1111 Ohjelmoinnin peruskurssi Y1 T-106.1208 Ohjelmoinnin perusteet Y (Python). Tentti 18.8.2014

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 ja käytä sisennyksiä selvästi. 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 sekä suomen- että englanninkielisen kysymyspaperin.

1. Kohdissa a, 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.

a) (2 p)

```
def main():
    palkka = 4500
    if palkka > 4000:
        print("Loistava palkka!")
    if palkka < 1000:
        print("Kehno palkka.")
    else:
        print("Menetteleehan tuo.")
```

main()

b) (2 p)

```
def main():
    brutto = 5000
    verot = 2000
    netto = brutto - verot
    brutto = 4500
    print(netto, brutto)
```

main()

c) (3 p)

```
def pienenna(suuri_luku):
    print(suuri_luku - 200)
```

```
def main():
    suuri_luku = 4000
    luku = 1000
    pienenna(luku)
```

main()

d) (5 p)

```
def vaihda(luku):
    luku = 5

def muuta(lista):
    lista[2] = 20

def main():
    arvo = 10
    vaihda(arvo)
    arvot = [100, 200, 300, 400]
    muuta(arvot)
    i = 0
    print(arvo)
    while i < 4:
        print(arvot[i])
        i += 1

main()
```

e) Funktiolle annetaan parametrina desimaalilukuja sisältävä lista. (4 p)

```
def mysteeril(luvut):
    i = 0
    while i < len(luvut):
        if luvut[i] < 0:
            return i
        i += 1
    return -1
```

f) Funktiolle annetaan parametrina merkkijono. (4 p)

```
def mysteeri2(mjono):
    uusi_jono = "?"
    i = 0
    while i < len(mjono):
        uusi_jono = uusi_jono + mjono[i]
        uusi_jono = uusi_jono + "?"
        i = i + 1
    return uusi_jono
```

g) Funktiolle annetaan parametrina kaksi kokonaislukuja sisältävää listaa, joilla on sama pituus. (5 p)

```
def mysteeri3(luvut1, luvut2):
    i = 0
    luvut3 = []
    while i < len(luvut1):
        if luvut1[i] == luvut2[i]:
            luvut3.append(luvut1[i])
        i += 1
    return luvut3
```

2. a) Tiina Teekkari on järjestämässä juhlia ja on pyytänyt tarjouksia ruokatarjoilun järjestämisestä kolmesta eri yrityksestä. Yritysten veloittamat summat riippuvat sekä vieraiden lukumäärästä että juhlien kestosta. Yritys A laskuttaa 20 eur / vieras ja lisäksi 100 eur / tunti. Yritys B laskuttaa 25 eur / vieras ja lisäksi 80 eur / tunti. Yritys C veloittaa 35 eur / vieras, mutta se ei laskuta lainkaan juhliin kuluneesta ajasta, vaan hinta riippuu vain vieraiden määrästä. Kirjoita Python-ohjelma, joka kysyy käyttäjältä vieraiden lukumäärän ja juhlien keston. Tämän jälkeen ohjelman pitää tulostaa, minkä yrityksen tarjous on halvin ja kuinka paljon juhlien tarjoilu maksaa tämän yrityksen järjestämänä. Jos useammalla kuin yhdellä yrityksellä on sama hinta, ohjelman tarvitsee tulostaa näistä halvimmista yrityksistä vain yksi (ei ole väliä mikä). (10 p.)

b) Eräs kurssi suoritetaan tekemällä harjoitustöitä. Jokainen harjoitustyö pisteytetään. Harjoitustyö on hyväksytty, jos siitä saatu pistemäärä on vähintään määrätty raja. Raja on sama kaikille harjoitustöille. Opiskelija saa hyväksytyin kurssisuorituksen, jos hänellä on vähintään 6 hyväksyttyä harjoitustöitä. Hyväksytyjen harjoitustöiden lisäksi opiskelijalla saa olla hylättyjä töitä, kunhan hyväksytyjä on tarpeeksi. Kirjoita Python-funktio `onko_hyvaksyty(pisteet, raja)`. Funktion ensimmäinen parametri on lista, joka sisältää opiskelijan pisteet eri harjoitustöistä (töiden määrä voi vaihdella eri opiskelijoilla), ja toinen parametri edellä kuvattu pisteraja yhden harjoitustyön hyväksymiseen. Funktion pitää palauttaa `True`, jos opiskelijan kurssisuoritus on hyväksytty, ja `False`, jos kurssisuoritusta ei ole hyväksytty. (20 p)

3. Internetissä toimiva kauppa tallentaa kunkin asiakkaan tilauksen yhteen tekstitiedostoon. Tiedostossa on yhdellä rivillä tilatun tuotteen tuotenumero, tilattu kappalemäärä (kokonaisluku) ja tuotteen yhden kappaleen hinta (desimaaliluku) tässä järjestyksessä. Eri tiedot on erotettu toisistaan kaksoispisteellä. Yksi tilaus voi sisältää useita tuotteita. Tiedoston rivit voisivat näyttää esim. seuraavilta:

```
A-12456:1:14.50
B-22556:2:20.00
A-65234:5:10.00
```

Yhden tuotteen tietoja on tiedostossa vain yhdellä rivillä. Jos asiakas tilaa kerralla vähintään 5 kappaletta samaa tuotetta, saa hän tästä tuotteesta 20 % alennuksen. Alennusta ei ole otettu tiedostoon kirjoitetuissa yksikköhinnoissa huomioon, vaan se lasketaan erikseen. Kirjoita Python-ohjelma, joka pyytää käyttäjältä tilauksen sisältävän tiedoston nimen. Ohjelma lukee tiedot tästä tiedostosta ja tulostaa tilauksen kokonaishinnan, kun kappalemäärät ja alennukset on otettu huomioon. Yllä olevassa esimerkkitapauksessa kokonaishinta olisi $14.50 + 2 * 20.00 + 5 * 8.00 = 94.50$. Ohjelman tarvitsisi tulostaa vain kokonaishinta 94.50 eikä annettua laskutoimitusta.

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä.
- Tiedoston jollain rivillä kappalemäärän paikalla ei ole kokonaisluku tai hinnan paikalla ei ole desimaaliluku.

Näissä tapauksissa ohjelma ilmoittaa käyttäjälle, millainen virhe on sattunut, ja lopettaa toimintansa. Ohjelman ei siis tarvitse jatkaa rivien lukemista virheellisen rivin jälkeen. Voit myös olettaa, että tiedoston jokaisella rivillä on täsmälleen kolme toisistaan kaksoispisteellä erotettua osaa. Ohjelman ei tarvitse osata käsitellä esimerkiksi sellaisia virhetilanteita, joissa rivi on tyhjä tai ei sisällä tuotenumeron lisäksi muuta tekstiä. (20 p)

TENTTI JATKUU SEURAAVALLA SIVULLA

4. Ystäväsi aikoo perustaa käytettyjen tavaroiden nettihuutokaupan. Kirjoita Python-kielellä luokka Huutokohde yhden myytävän tavarahan (kohteen) tietojen käsittelyyn.

Huutokohde-oliolla on oltava seuraavat kentät:

- `__nimi` myytävän kohteen nimi
- `__hintavaraus` myyjän asettama vähimmäishinta, jota pienemmällä hinnalla myyjä ei myy kohdetta. Ostajat voivat kuitenkin tehdä tätä pienempiä tarjouksia.
- `__onko_auki` kentän arvo on `True`, jos kohde on tällä hetkellä huudettavana (siitä voidaan tehdä tarjouksia) ja `False`, jos kohteesta ei voi tehdä tarjouksia, koska sen myyntiaika ei ole vielä alkanut tai myyntiaika on jo päättynyt.
- `__korkein_tarjous` kohteesta tähän mennessä tehty korkein tarjous (summa euroina)
- `__tarjoaja` tähän mennessä korkeimman tarjouksen tekijä (nimi)

Huomaa, että kohteen yhteydessä säilytetään tietoa vain korkeimmasta tarjouksesta ja aikaisemmat tarjoukset tavallaan unohdetaan.

Määrittele luokkaan seuraavat metodit. (Halutessasi saat määrittellä muitakin metodeita. Jos metodin kuvauksessa ei ole kerrottu mitään metodin palauttamasta arvosta, metodin ei tarvitse palauttaa mitään.)

- `__init__(self, nimi, hintavaraus)` luo uuden Huutokohde-olion. Luotavan kohteen nimi ja hintavaraus annetaan parametreina. Jos viimeinen parametri on negatiivinen, hintavaraukseksi asetetaan `0.0`. Uusi kohde ei ole auki, korkein tarjous on `0.0` ja tarjoaja tyhjä merkkijono (`""`).
- `kerro_nimi(self)` palauttaa kohteen nimen.
- `kerro_hintavaraus(self)` palauttaa kohteen hintavarauksen.
- `avaa_kohde(self)` avaa kohteen huudettavaksi.
- `sulje_kohde(self)` sulkee kohteen.
- `tee_tarjous(self, summa, tekija)` Tätä metodia käytetään, kun joku tekee kohteesta uuden tarjouksen. Metodin parametreina annetaan uuden tarjouksen määrä (euroina) ja tarjouksen tekijän nimi. Metodi tarkistaa, että kohde on auki ja että uusi tarjous on vähintään 2 euroa suurempi kuin kohteesta aikaisemmin tehty korkein tarjous. Jos molemmat ehdot pitävät paikkansa, metodi muuttaa kenttien `__korkein_tarjous` ja `__tarjoaja` arvoja ja palauttaa arvon `True`. Muussa tapauksessa metodi ei muuta minkään kentän arvoa ja palauttaa arvon `False`.
- `maaraa_myyntihinta(self)` palauttaa kohteen lopullisen myyntihinnan (korkeimman tarjouksen arvon), jos kohde on jo suljettu ja siitä tehty korkein tarjous on vähintään yhtäsuuri kuin kohteen hintavaraus. Muussa tapauksessa metodi palauttaa arvon `-1.0` kertomaan, että kohteen myynti ei (ainakaan vielä) onnistu.
- `maaraa_ostaja(self)` palauttaa kohteen ostajan (korkeimman tarjouksen tekijän nimen), jos kohde on jo suljettu ja siitä tehty korkein tarjous on vähintään yhtäsuuri kuin kohteen hintavaraus. Muussa tapauksessa metodi palauttaa tyhjän merkkijonon (`""`) kertomaan, että kohteen myynti ei (ainakaan vielä) onnistu.
- `__str__(self)` palauttaa merkkijonon, joka sisältää kohteen nimen, hintavarauksen, korkeimman tarjouksen ja sen tekijän nimen ja joko tekstin "kohde on auki" tai "kohde on suljettu" sen mukaan, onko kohde auki vai ei.

Kirjoita lisäksi pääohjelma, joka luo kaksi Huutokohde-oliota, avaa ne molemmat ja tekee niistä molemmista kaksi tarjousta. Yhden tarjouksen kohdalla ohjelman pitää tulostaa, onnistuiko tarjous (oliko sen summa tarpeeksi suuri niin, että tarjous hyväksyttiin uudeksi korkeimmaksi tarjoukseksi). Tämän jälkeen ohjelman pitää sulkea toinen kohteista ja tulostaa sen ostajan nimi ja myyntihinta tai tieto siitä, että kohteen myynti ei onnistunut. Lopuksi ohjelman on tulostettava molemmista kohteista nimi, hintavaraus, korkein tarjous ja sen tekijä sekä tieto siitä, onko kohde auki. Voit päättää kohteiden alkutiedot ja tarjousten 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)