

T-106.1208 Ohjelmoinnin perusteet Y (Python). Tentti 20.10.2012

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 merkitse sisennykset selvästi. Jos sisennyksiä ei ole käytetty tai ne on merkitty epäselvästi, 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).**

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

a) (2 p) ✓ 2p

```
def main():
    pisteet = 1500
    if pisteet > 1400:
        print "Huippupisteet"
    if pisteet > 1000:
        print "Pisteet OK"
    else:
        print "Huonot pisteet"
```

main()

b) (2 p) ✓ 2p

```
def main():
    paiva = "lauantai"
    kello = 6
    if paiva == "lauantai" and paiva == "sunnuntai":
        print "Lippu maksaa 2 euroa."
    else:
        if kello < 7 or kello >= 17:
            print "Lippu maksaa 3 euroa."
        else:
            print "Lippu maksaa 4 euroa."
```

main()

c) (3 p) ✓ 3p

```
def main():
    luvut = [-4, 10, 12, -3, 4]
    tulos = 0
    for luku in luvut:
        if luku > 0:
            tulos = tulos + luku
    print tulos
```

main()

d) (5 p)

```

def muuta1(paral):
    paral = paral - 2

def muuta2(paralista):
    paralista[1] = paralista[2] + 1

def main():
    luku = 10
    muuta1(luku)
    luvut = [2, 4, 6]
    muuta2(luvut)
    print luku
    i = 0
    while i < 3:
        print luvut[i]
        i += 1

main()

```

paral = 8

0 1 2
paral, bc = 2, 4 6
2 7 6

e) Funktiolle annetaan parametrina kaksi kokonaislukuja sisältävää listaa, joilla on sama pituus (4 p)

```

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

```

f) Funktiolle annetaan parametrina merkkijono (5 p)

```

def mysteeri2(nimi):
    pit = len(nimi)
    if pit >= 4 and nimi[pit-4:pit] == ".doc":
        return nimi[0:pit-4] + ".txt"
    else:
        return nimi

```

g) Seuraavan funktion tarkoituksena on tarkistaa, onko sille parametrina annettujen listojen sisältö sama eli onko niissä samat luvut vastaavilla paikoilla. Voit olettaa, että funktiolle annetaan parametrina kaksi kokonaislukuja sisältävää listaa, joiden pituus on sama. Mikä virhe funktiossa on ja miten se vaikuttaa funktion toimintaan? (Etsitty virhe ei siis ole se, että listojen pituus voi olla eri, jos tehtävänannon oletukset eivät pidä paikkaansa.) (4 p)

```

def onko_samat(lista1, lista2):
    i = 0
    while i < len(lista1):
        if lista1[i] != lista2[i]:
            return False
        else:
            return True
        i += 1

```

2. a) Eräs yritys maksaa myyntiedustajalleen peruspalkkaa 650 euroa kuukaudessa sekä lisäksi myyntiprovisio seuraavasti: Jos edustajan kuukauden kokonaismyynti on vähintään 9000 mutta alle 14000 euroa, provisio on 10 % kokonaismyynnistä. Jos kuukauden kokonaismyynti on vähintään 14000 mutta alle 21000 euroa, provisio on 15 % kokonaismyynnistä. Jos kokonaismyynti on vähintään 21000 euroa, provisio on 20 % kokonaismyynnistä. Kirjoita Python-ohjelma, joka kysyy käyttäjältä myyntiedustajan kuukausimyynnin ja laskee ja tulostaa hänen kokonaispalkkansa (peruspalkka + myyntiprovisio). (10 p)

b) Eräs kauppa järjestää alennusmyyntikampanjan, jossa se antaa 10 %:n alennuksen jokaisesta tuotteesta, jonka hinta ylittää kaupan määräämän rajan. Kirjoita kampanjan käyttämää tietokoneohjelmaa varten Python-funktio `laske_kokonaissumma(ostokset, alennusraja)`. Funktio saa ensimmäisenä parametrina listan, joka sisältää asiakkaan ostamien tuotteiden normaalihinnat ja toisena parametrina kaupan asettaman hintarajan alennukselle. Funktio laskee ja palauttaa sen summan, jonka asiakkaan pitää tuotteista maksaa (eli tuotteiden hinnat laskettuna yhteen niin, että mahdolliset alennukset on otettu huomioon). Jos esimerkiksi lista sisältää luvut 12.0, 30.0, 20.0 ja 12.0 ja alennusraja on 15.0, niin funktion tulee palauttaa arvo 69.0. Tehtävässä ei tarvitse kirjoittaa pääohjelmaa tai muuta ohjelman osaa, vaan pyydetyn funktion kirjoittaminen riittää. (20 p)

3. Eräessä kilpailussa on ensin kaksi alkukierrosta, joista kukin kilpailija voi kerätä pisteitä. Alkukierrosten jälkeen valitaan jatkoon ne kilpailijat, joiden yhteispistemäärä (alkukierrosten pisteet yhteensä) on tarpeeksi suuri. Kilpailijoiden pisteet on tallennettu tekstitiedostoon siten, että jokaisella rivillä on yhden kilpailijan nimi ja sitten tämän kilpailijan saamat pisteet kahdelta eri kierrokselta (pisteet ovat kokonaislukuja). Eri tiedot on erotettu toisistaan kaksoispisteellä. Tiedoston rivit voisivat näyttää esimerkiksi seuraavilta:

```
Raija Raksalainen:18:16
Kalle Kemisti:21:11
```

Kirjoita Python-ohjelma, joka pyytää käyttäjältä pisteet sisältävän tiedoston nimen sekä sen pistemäärän, joka on vähintään oltava, jotta pääsisi jatkoon. Ohjelma lukee tämän tiedoston ja tulostaa jatkoon valittujen kilpailijoiden nimet ja yhteispisteet.

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä.
- Käyttäjän antamaa alinta pistemäärää ei voi tulkita kokonaisluvuksi.
- Tiedoston jollain rivillä pisteiden paikalla olevaa tekstiä ei voi tulkita kokonaisluvuksi.

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 nimen lisäksi muuta tekstiä. (20 p)

Viimeinen tehtävä seuraavalla sivulla

4. Kirjoita Python-kielessä 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)` "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. Puhelun jokainen alkava 9 sekuntia maksaa yhden sentin. Metodi palauttaa puhelun todellisen keston (joka on siis parametrina annettua kestoja pienempi, jos saldo ei riitä haluttuun kesto). Käytännössä "soittaminen" näkyy siten, että metodi pienentää liittymän saldoa puhelun hinnalla. Jos liittymä on suljettu, 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 toiselle rahaa, soittaa sen jälkeen toisella liittymällä puhelun ja kertoo sen keston. Tämän jälkeen pääohjelman pitää lähettää toisella 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)