

**T-106.1208 Ohjelmoinnin perusteet Y (Python). Tentti 15.8.2011**

Kirjoita jokaisen vastauspaperisi alkuun kurssin nimi, kokeen päivämäärä, nimesi, opiskelijanumerosi (myös tarkistuskirjain), 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.

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. 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)

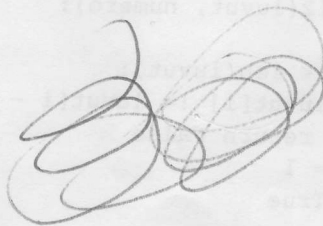
```
def main():
    ph = 3.0
    if ph < 6.5:
        print "Hapan liuos"
    if (ph >= 6.5) and (ph <= 7.5):
        print "Neutraali liuos"
    else:
        print "Emaksinen liuos"

main()
```

b) (2 p)

```
def main():
    paiva = "maanantai"
    aika = 12
    if paiva == "perjantai" or paiva == "lauantai":
        print "Hinta on 150 euroa."
    else:
        if (aika < 16) and (aika > 22):
            print "Hinta on 90 euroa."
        else:
            print "Hinta on 120 euroa."
```

main()



c) (3 p)

```
def main():
    tulos = 0
    raja = 50
    luvut = [40, 80, 25, -15, 90, 49]
    for luku in luvut:
        if luku < raja:
            tulos = tulos + luku
    print tulos
```

main()

d) (5 p)

```
def muuta1(luvut):
    luvut[0] = 10

def muuta2(para):
    para = para - 2

def main():
    tieto = 30
    muuta2(tieto)
    lukulista = [15, 20, 25]
    muuta1(lukulista)
    print tieto
    i = 0
    while i < 3:
        print lukulista[i]
        i += 1
```

main()

e) Funktiolle annetaan parametreina kaksi kokonaislukua sisältävää listaa, joilla on sama pituus (4 p)

```
def mysteeril(luvut1, luvut2):
    tulos = 0
    i = 0
    while i < len(luvut1):
        if luvut1[i] < luvut2[i]:
            tulos = tulos + 1
        i = i + 1
    return tulos
```

f) Funktiolle annetaan parametrina kokonaislukuja sisältävä lista ja kokonaisluku. (4 p)

```
def mysteeri2(luvut, numero):
    i = 1
    while i < len(luvut):
        if luvut[i] != luvut[i - 1] + numero:
            return False
        i += 1
    return True
```

g) Funktiolle annetaan parametrina merkkijono (5 p)

```
def mysteeri3(merkkijono):
    luku = len(merkkijono)
    if merkkijono[luku-4:luku] == ".txt":
        return merkkijono
    return merkkijono + ".txt"
```

2. a) Remonttimies tarjoaa palveluksiaan kodin pieniin ja suuriin korjauksiin seuraavin ehdoin: Neljältä ensimmäiseltä tunnilta veloitetaan normaali tuntihinta. Sen jälkeen seuraavilta tunneilta remonttimies veloittaa alennetun tuntihinnan. Hän ottaa kuitenkin aina maksun vähintään kolmesta tunnista, vaikka työ olisi kestänyt vähemmän aikaan. Kirjoita Python-ohjelma, joka pyytää käyttäjältä normaalin tuntihinnan, alennetun tuntihinnan ja työn keston tunteina. Ohjelman pitää laskea ja tulostaa remonttimiehen veloittama kokonaishinta. (10 p.)

b) Erään kurssin voi suorittaa tekemällä harjoitustehtäviä. Jokaisen harjoitustehtävän ratkaisu pisteytetään erikseen, ja jos tehtävän pistemäärä ylittää määrätyn rajan, tehtävä on hyväksytty (hyväksymisraja on sama kaikille tehtäville). Jos opiskelijalla on kurssin lopussa vähintään 12 hyväksyttyä harjoitustehtävää, hän saa arvosanan 5. Jos hyväksytyjen harjoitustehtävien määrä on 9-11, hän saa arvosanan 3. Jos hyväksytyjen harjoitustehtävien määrä on 6-8, hän saa arvosanan 1. Jos hyväksytyjen tehtävien määrä on alle 6, opiskelijan suoritus on hylätty eli hän saa kurssista arvosanan 0. Kirjoita kurssin tuloksia käsittelevän ohjelman osaksi Python-funktio `laske_arvosana(pisteet, raja)`. Funktion ensimmäisenä parametrina on lista, joka sisältää opiskelijan eri tehtävistä saamat pisteet (listan yksi alkio on yhden harjoitustehtävän pistemäärä). Listan alkiot ovat kokonaislukuja, eikä listan pituutta ole määrätty etukäteen. Funktion toisena parametrina on harjoitustehtävän hyväksymisraja (yksi kokonaisluku, koska raja on sama kaikille tehtäville). Funktion on laskettava ja palautettava opiskelijalle kuuluva arvosana. Kirjoita vain pyydetty funktio, älä muita ohjelman osia. (20 p.)



3. Eräs iltapäivälehteä julkaiseva lehtiyhtiö toimittaa joka päivä eri jälleenmyyjille sovitus määrän lehtiä myytäväksi (kappalemäärä vaihtelee jälleenmyyjän mukaan). Seuraavana päivänä yhtiö hakee jälleenmyyjiltä takaisin myymättä jääneet lehdet. Yhtiö kirjaa tekstitiedostoon kunkin jälleenmyyjän osalta myyntipaikan nimen, paikkaan toimitettujen lehtien määrän ja paikasta palautettujen lehtien määrän. Tiedoston rivit voisivat näyttää esimerkiksi seuraavilta:
- Otaniemen R-kioski:40:10  
 Tapiolan S-market:60:20  
 Suomalainen Kirjakauppa Tapiola:50:1  
 Otavalinta:30:15

Tiedostossa on siis yhdellä rivillä aina yhden myyntipaikan tiedot, ja eri kentät (paikan nimi, toimitettujen lehtien määrä, palautettujen lehtien määrä) on erotettu toisistaan kaksoispisteellä. Lehtiyhtiö haluaa tarkkailla palautettujen lehtien osuutta myyntiin toimitetuista lehdistä ja poimia esiin ne myyntipaikat, jotka ovat palauttaneet yli 30 % myyntiin toimitetuista lehdistä. Kirjoita Python-ohjelma, joka pyytää käyttäjältä lehtitiedot sisältävän tiedoston nimen. Ohjelma lukee tämän tiedoston ja tulostaa niiden myyntipaikkojen nimet, joissa palautettujen lehtien osuus on ollut yli 30 %. Esimerkkitapauksessa ohjelman pitäisi tulostaa

Tapiolan S-market  
 Otavalinta

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimestä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä.
- Tiedoston jollain rivillä toimitettujen tai palautettujen lehtien määrän 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)

4. Kirjoita Python-kielellä 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).

**JATKU SEURAAVALLA SIVULLA**

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)