

## T-106.1208 Ohjelmoinnin perusteet Y (Python). Tenti 8.5.2009

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. a) Mitä seuraava ohjelma tulostaa? Vastausta ei tarvitse perustella. (2 p)

```
def main():
    ylinopeutta = 25.0
    if ylinopeutta > 10.0:
        print "Rikesakko"
    elif ylinopeutta > 20.0:
        print "Paivasakkoja"
    else:
        print "Ei sakkoja"
```

```
main()
```

b) Seuraavassa ohjelmassa muuttujille ika ja paiva annetaan jotkin arvot, joita ohjelmaan ei ole merkitty (niiden tilalle esitettyssä ohjelmakoodissa on kirjoitettu kolme pistettä). Mitkä arvot näille muuttujille pitää antaa, jotta ohjelma tulostaisi Taysi maksu? Yhdet sopivat arvot riittävät. Sinun ei siis tarvitse luetella kaikkia mahdollisia vaihtoehtoja. (2 p)

```
def main():
    ika = ... # Kolmen pisteen paikalla jokin luku
    paiva = ... # Kolmen pisteen paikalla jokin merkkijono
    if ika <= 6:
        print "Vapaa paasy"
    else:
        if paiva == "lauantai" or paiva == "sunnuntai":
            print "Vapaa paasy"
        else:
            print "Taysi maksu"
```

```
main()
```

c) Mitä seuraava ohjelma tulostaa? Vastausta ei tarvitse perustella. (3 p)

```
def main():
    luvut = [1, 25, 4, 9, 16]
    tulos = 100
    for luku in luvut:
        tulos = tulos - luku
    print tulos
```

main()

d) Mitä alla esitetty funktio `mysteeri1` tekee? Älä selitä funktion toimintaa käsky käskyltä, vaan selitä 1-2 lauseella, mikä on funktion tarkoitus. Voit olettaa, että funktiolle annetaan parametrina kaksi listaa, jotka molemmat sisältävät kokonaislukuja. Molemmilla listoilla on sama koko. (4 p)

```
def mysteeri1(luvut1, luvut2):
    tulos = 0
    i = 0
    while i < len(luvut1):
        if luvut1[i] != luvut2[i]:
            tulos = tulos + 1
        i = i + 1
    return tulos
```

e) Mitä alla esitetty funktio `mysteeri2` tekee? Älä selitä funktion toimintaa käsky käskyltä, vaan selitä 1-2 lauseella, mikä on funktion tarkoitus. Voit olettaa, että funktiolle annetaan ensimmäisenä parametrina desimaalilukuja sisältävä lista ja toisena parametrina yksi desimaaliluku. (4 p)

```
def mysteeri2(lista, luku):
    tulos = 0.0
    for alkio in lista:
        tulos = tulos + luku * alkio
    return tulos
```

f) Mitä alla esitetty funktio `mysteeri3` tekee? Älä selitä funktion toimintaa käsky käskyltä, vaan selitä 1-2 lauseella, mikä on funktion tarkoitus. Voit olettaa, että funktiolle annetaan parametrina kokonaislukuja sisältävä lista. (5 p)

```
def mysteeri3(luvut):
    i = 0
    while i < len(luvut) - 1:
        if luvut[i] <= luvut[i + 1]:
            return False
        i = i + 1
    return True
```

g) Mikä alla esitetty funktio `mysteeri4` tekee? Älä selitä funktion toimintaa käsky käskyltä, vaan selitä 1-2 lauseella, mikä on funktion tarkoitus. Voit olettaa, että funktiolle annetaan ensimmäisenä parametrina merkkijono ja toisena parametrina yksi kokonaisluku. (5 p)

```
def mysteeri4(merkkijono, luku):
    if len(merkkijono) >= luku:
        return merkkijono[0:luku]
    else:
        return merkkijono + (" " * (luku - len(merkkijono)))
```

2. a) Eräs kauppa järjestää alennusmyyntikampanjan, jossa se antaa alennusta tuotteiden hinnoista seuraavasti: Jos tuotteen alkuperäinen hinta on vähintään 200 euroa, siitä saa alennusta 10 %. Jos tuotteen alkuperäinen hinta on vähintään 100 euroa, mutta alle 200 euroa, siitä saa alennusta 5 %. Jos tuotteen alkuperäinen hinta on alle 100 euroa, siitä ei saa alennusta. Kirjoita ohjelma, joka pyytää käyttäjältä tuotteen alkuperäisen hinnan ja tulostaa alennettun hinnan. (10 p.)

b) Eräällä kurssilla on seuraava käytäntö: Kurssilla on useita harjoitustehtäväkierroksia, joilla kaikilla on sama minimipistemäärä. Jos opiskelijan pisteet ylittävät yhdelläkin kierroksella minimipistemäärän, hänen harjoitustehtävänsä on hylätty. Jos opiskelijalla on jokaiselta kierrokselta vähintään minimipistemäärä, hänen harjoitustehtävänsä on hyväksytty. Kirjoita kurssin tietojärjestelmien käyttöön funktio `onko_hyvaksyetty(pisteet, minimiraja)`. Funktio saa ensimmäisenä parametrina listan, joka sisältää opiskelijan harjoitustehtäväpisteet eri kierroksilta (pisteet ovat kokonaislukuja). Funktion toisena parametrina on kokonaisluku, joka on kierrosten minimipistemäärä. Funktio palauttaa arvon `True`, jos opiskelijan harjoitustehtävät on hyväksytty ja arvon `False`, jos opiskelijan harjoitustehtävät on hylätty. Kirjoita vain tehtävässä pyydetty funktio. Sinun ei siis tarvitse kirjoittaa pääohjelmaa tai muuta ohjelman osaa. (20 p)

3. Kirjoita ohjelma, joka pyytää käyttäjältä tiedoston nimen. Ohjelma lukee tästä tiedostosta lämpötiloja. Kukin lämpötila (desimaaliluku) on annettu omalla rivillään, mutta lämpötilojen lukumäärä ei ole tiedossa etukäteen. Kun tiedosto on luettu loppuun, ohjelma tulostaa, kuinka moni luetuista lämpötiloista on alle 0 astetta, kuinka moni on 0 - 25 astetta ja kuinka moni on yli 25 astetta. Alla on esimerkki halutusta tulostuksesta ohjelman lopussa (luvut riippuvat siitä, mitä lämpötiloja tiedosto sisältää):

```
Lämpötila alle 0 astetta 4 kertaa.
Lämpötila 0 - 25 astetta 10 kertaa.
Lämpötila yli 25 astetta 2 kertaa.
```

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä
- Tiedoston jokin rivi on virheellinen (esimerkiksi rivillä on yhden desimaaliluvun lisäksi jotain muuta tekstiä tai rivillä ei ole lainkaan lukua)

Näissä tapauksissa ohjelma ilmoittaa käyttäjälle virheestä ja lopettaa toimintansa. Ohjelman ei siis tarvitse jatkaa rivien lukemista virheellisen rivin jälkeen tai ilmoittaa virhetilanteesta jo luettujen lämpötilojen jakaumaa. (20 p.) **JATKUU SEURAAVALLA SIVULLA**

Jos tehtävä tuntuu liian vaikealta, voit tehdä ohjelmasta yksinkertaisemman version, joka ei toteuta kaikkia annettuja vaatimuksia (esim. ei käsittele virhetilanteita tai lukee lämpötilat käyttäjältä tiedoston sijasta.) Tällaisesta ohjelmasta saa kuitenkin selvästi vähemmän pisteitä (ohjelmasta, joka ei käsittele vaadittuja virhetilanteita, saa korkeintaan 15 pistettä ja ohjelmasta, joka lukee lämpötilat käyttäjältä tiedoston sijasta, saa korkeintaan 7 pistettä).

4. Eräs yritys haluaa panna www-sivulleen pelin, jota halukkaat käyttäjät voivat pelata. Sivulla pidetään myös kirjaa pelaajista ja heidän ennätyksistään. Kirjoita sivua käyttämää ohjelmistoa varten luokka `Pelaaja` seuraavan kuvauksen mukaisesti (muuta ohjelman osaa ei tarvitse toteuttaa, paitsi luokan lisäksi tehtävänannon lopussa selitetty pääohjelma).

Luokalla `Pelaaja` on seuraavat kentät:

- `__nimi` pelaajan nimi
- `__pelikm` pelaajan pelaamien pelien lukumäärän
- `__ennatys` pelaajan ennätys (suurin pistemäärä pelaajan pelaamista kaikista peleistä).
- `__yhteispisteet` pelaajan kaikista peleistä yhteensä saamat pisteet.

Määrittele luokkaan seuraavat metodit. (Tehtävän ratkaisun lyhentämiseksi luokasta on jätetty pois metodeita, joita siihen olisi järkevä määrittellä. Jos metodin kuvauksessa ei ole kerrottu mitään metodin palauttamasta arvosta, metodin ei tarvitse palauttaa mitään.)

- `__init__(self, pelaajan_nimi)` luo uuden `Pelaaja`-olion. Luotavan pelaajan nimi annetaan metodin parametrina. Uuden pelaajan ennätys, pelien lukumäärä ja yhteispisteet ovat 0.
- `kerro_ennatys(self)` palauttaa pelaajan ennätyksen.
- `lisaa_peli(self, pisteet)` lisää pelaajalle tiedon uudesta pelistä. Pelistä saadut pisteet annetaan metodin parametrina. metodi siis muuttaa pelaajan pelaamien pelien määrää sekä yhteispisteitä. Tarvittaessa metodi myös muuttaa pelaajan ennätystä.
- `laske_keskiarvo(self)` laskee ja palauttaa pelaajan kaikista peleistä saamien pisteiden keskiarvon.
- `onko_mestari(self)` palauttaa arvon `True`, jos pelaaja on mestaripelaaja ja muussa tapauksessa arvon `False`. Pelaaja on mestaripelaaja, jos hänen ennätyksensä on vähintään 4500 ja lisäksi hän on pelannut vähintään 50 peliä. Muussa tapauksessa pelaaja ei ole mestaripelaaja.
- `__str__(self)` palauttaa merkkijonon, joka sisältää pelaajan nimen, ennätyksen ja pelaajan pelaamien pelien lukumäärän.

Kirjoita lisäksi pääohjelma, joka luo kaksi `Pelaaja`-oliota, kutsuu molemmille kaksi kertaa `lisaa_peli`-metodia ja sen jälkeen toiselle pelaajista myös `kerro_ennatys`- ja `laske_keskiarvo`-metodeita. Pääohjelman pitää myös tulostaa metodien palauttama ennätys ja keskiarvo. Tämän jälkeen pääohjelman pitää selvittää toisesta pelaajasta, onko hän mestaripelaaja ja tulostaa joko "Pelaaja on mestaripelaaja" tai "Pelaaja ei ole mestaripelaaja" selvityksen tuloksen mukaisesti. Lopuksi pääohjelman on tulostettava molemmista pelaajista perustiedot (nimi, ennätys ja pelattujen pelien lukumäärä.) Voit päättää luotavien pelaajien nimet ja pelien pisteet itse. Pääohjelman ei siis tarvitse kysyä mitään käyttäjältä. (25 p)