

T-106.1208 Ohjelmoinnin perusteet Y (Python). Tentti 24.10.2009

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():
    lampotila = 28.0
    if lampotila >= 0.0:
        print "lammin"
    elif lampotila >= 25.0:
        print "helle"
    else:
        print "pakkanen"

main()
```

- b) Seuraavassa ohjelmassa muuttujille **kuukausi** 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 **Alennus on 20 %**? Yhdet sopivat arvot riittävät. Sinun ei siis tarvitse luetella kaikkia mahdollisia vaihtoehtoja. (2 p)

```
def main():
    kuukausi = ... # Kolmen pisteen paikalla jokin merkkijono
    paiva = ... # Kolmen pisteen paikalla jokin luku
    if kuukausi == "kesakuu" or kuukausi == "heinakuu":
        print "Ei alennusta"
    else:
        if paiva > 10 and paiva < 16:
            print "Alennus on 20 %"
        else:
            print "Alennus on 40 %"

main()
```

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

```
def main():
    luvut = [4, -16, 21, 18, -3]
    tulos = 0
    for luku in luvut:
        if luku > 0:
            tulos = tulos + luku
    print tulos

main()
```

4 + 21 + 18

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):
    i = 0
    tulos = 0
    while i < len(luvut1):
        if luvut1[i] > luvut2[i]:
            tulos += luvut1[i]
        else:
            tulos += luvut2[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 sekä toisena ja kolmantena parametrina yksi desimaaliluku. (4 p)

```
def mysteeri2(luvut, luku1, luku2):
    tulos = luku2
    for alkio in luvut:
        if alkio < luku1:
            tulos = tulos - 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 = 1
    while i < len(luvut):
        if luvut[i] != 2 * luvut[i-1]:
            return False
        i += 1
    return True
```

g) Mitä 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 parametrina kaksi merkkijonoa. Mihin funktiota voisi käyttää? (5 p)

```
def mysteeri4(mjonol, mjono2):
    if len(mjonol) == 0 or len(mjono2) == 0:
        return "virhe"
    else:
        uusi = mjonol[0] + "." + mjono2[0] + "."
        return uusi
```


2. a) Eräs pankki perii välittämistään osakekaupoista välityspalkkiona yleensä 0,25 % ostetun tai myydyin osake-erän kokonaishinnasta. Jos kuitenkin näin laskettu palkkio olisi pienempi kuin 10 euroa, pankki perii 10 euron välityspalkkion. Kirjoita Python-ohjelma, joka kysyy käyttäjältä tämän ostamien osakkeiden lukumäärän ja yhden osakkeen kappalehinnan (kaikki ostetut osakkeet ovat saman hintaisia). Sen jälkeen ohjelma laskee ja tulostaa pankin perimän välityspalkkion. (10 p.)

b) Eräs kauppa antaa kanta-asiakkailleen bonuspisteitä seuraavasti: Jos yhden ostokerran summa on pienempi kuin kaupan määräämä `raja1`, ei siitä tule bonuspisteitä lainkaan. Jos ostokerran summa on vähintään yhtäsuuri kuin `raja1`, mutta pienempi kuin kaupan määräämä `raja2`, tulee siitä 5 bonuspistettä. Jos ostokerran summa on vähintään yhtäsuuri kuin `raja2`, tulee siitä 10 bonuspistettä. Kirjoita kaupan tietokoneohjelmaa varten Python-funktio `laske_bonuspisteet(ostokset, raja1, raja2)`. Funktio saa ensimmäisenä parametrina listan, joka sisältää asiakkaan eri ostokertojen summat, ja toisena ja kolmantena parametrina edellä kuvatut rajat `raja1` ja `raja2`. Funktio laskee ja palauttaa asiakkaalle kuuluvien bonuspisteiden määrän. Jos esimerkiksi lista sisältää luvut 12.0, 30.0, 20.0 ja 12.0, `raja1` on 15.0 ja `raja2` on 25.0, niin funktion tulee palauttaa arvo 15. Tehtävässä ei tarvitse kirjoittaa pääohjelmaa tai muuta ohjelman osaa, vaan pyydetyn funktion kirjoittaminen riittää. (20 p)

3. Kirjoita Python-ohjelma, joka pyytää käyttäjältä tiedoston nimen ja yhden opiskelijanumeron. Ohjelma lukee tästä tiedostosta rivejä. Jokaisella rivillä on yhden opiskelijan tiedot niin, että rivin alussa on opiskelijanumero ja sen jälkeen opiskelijan muut tiedot (esim. nimi ja tulokset jollain kurssilla). Ohjelma lukee tiedoston rivit ja jos jonkin rivin alussa on käyttäjän antama opiskelijanumero, ohjelma tulostaa koko tämän rivin sellaisenaan. Jos tiedoston minkään rivin alusta ei löyty haettua opiskelijanumeroa, ohjelma kertoo lopuksi käyttäjälle, että haluttua opiskelijaa ei löytynyt. Voit olettaa, että tiedosto on virheetön eli jokaisella rivillä on tosiaan aluksi kuuden merkin mittainen opiskelijanumero. Voit myös olettaa, että käyttäjä antaa opiskelijanumeroa pyydettyä tasan kuuden merkin mittaisen merkkijonon.

Ohjelman on kuitenkin käsiteltävä oikein (annettava käyttäjälle järkevä virheilmoitus) sellainen virhetilanne, jossa annetun nimistä tiedostoa ei ole tai tiedoston lukeminen ei onnistu jostain muusta syystä.

Vinkki: Kun olet lukenut tiedostosta yhden rivin, voit verrata sen kuutta ensimmäistä merkkiä käyttäjän antamaan opiskelijanumeroon. (20 p)

Viimeinen tehtävä seuraavalla sivulla!

4. Kirjoita Python-kielellä luokka `Jalkapallojoukkue` yhden jalkapallojoukkueen tietojen kuvaamiseen.

Luokalla `Jalkapallojoukkue` on oltava seuraavat kentät:

- `__nimi` joukkueen nimi
- `__pisteet` joukkueen pisteet jossain jalkapallosarjassa.
- `__tehdyt_maalit` joukkueen sarjan otteluissa itse tekemät maalit (yhteensä kaikissa otteluissa).
- `__paastetyt_maalit` joukkueen otteluissa vastustajien tekemät maalit (yhteensä kaikissa otteluissa).

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

- `__init__(self, joukkueen_nimi)` luo uuden `Jalkapallojoukkue`-olion. Luotavan joukkueen nimi annetaan metodin parametrina. Uuden joukkueen pisteet sekä tehdyt ja päästetyt maalit ovat 0.
- `kerro_nimi(self)` palauttaa joukkueen nimen.
- `kerro_pisteet(self)` palauttaa joukkueen pisteet.
- `laske_maaliero(self)` laskee ja palauttaa joukkueen maalieron eli tehtyjen maalien ja päästettyjen maalien erotuksen.
- `lisaa_ottelun_tiedot(self, omat_maalit, vastustajan_maalit)` lisää joukkueelle yhden ottelun tiedot eli päivittää joukkueen piste- ja maalitietoja ottelun tuloksen mukaisesti. Joukkueen tässä ottelussa tekemät maalit ja vastustajajoukkueen tekemät maalit on annettu metodin parametreina. Voitosta saa 3 pistettä, tasapelistä 1 pisteen ja tappiosta 0 pistettä. (Maalitiedoista voi päätellä sen, onko ottelu päätynyt voittoon, tasapeliin vai tappioon.) Metodi ei muuta vastustajajoukkueen piste- ja maalitietoja.
- `onko_parempi(self, toinen_joukkue)` palauttaa arvon `True`, jos tämä joukkue on parempi kuin parametrina annettu toinen joukkue, ja muuten arvon `False`. Kahdesta joukkueesta on ensisijaisesti parempi se, jolla on enemmän pisteitä. Jos joukkueiden pisteet ovat samat, parempi joukkue on se, jolla on suurempi maaliero. Jos sekä pisteet että maaliero ovat samat, joukkueet ovat yhtä hyviä. Tällöin metodi palauttaa myös arvon `False`.
- `__str__(self)` palauttaa merkkijonon, joka sisältää joukkueen nimen, pisteet sekä tehdyt ja päästetyt maalit.

Kirjoita lisäksi pääohjelma, joka luo kaksi `Jalkapallojoukkue`-oliota ja kutsuu niille molemmille kerran `lisaa_ottelun_tiedot`-metodia. Sen jälkeen ohjelman on kutsuttava toiselle joukkueelle `laske_maaliero`-metodia ja tulostettava joukkueen maaliero. Sitten ohjelman on selvitettävä, kumpi joukkue on parempi (käyttämällä `onko_parempi`-metodia) ja tulostettava paremman joukkueen nimi. Jos molemmat joukkueet ovat yhtä hyviä, ohjelma saa tulostaa kumman tahansa nimen. Lopuksi ohjelman on tulostettava kummankin joukkueen nimi, pisteet sekä tehdyt ja päästetyt maalit. Voit päättää joukkueiden nimet ja kirjattavissa otteluissa tehdyt maalit 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)