CSE-A1111 Ohjelmoinnin peruskurssi Y1 Tentti 28.5.2015

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. 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 kysymyspaperit sekä suomeksi että englanniksi tai ruotsiksi.

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():
    lampotila = 28
    if lampotila > 10:
        print("Lammin.")
    elif lampotila >= 25:
        print("Helle.")
    else:
        print("Kylma.")

main()
```

b) (2 p)

```
def main():
    hinta = 100
    alennus = 10
    uusi_hinta = hinta - alennus
    alennus = 30
    print(uusi_hinta)

main()
```

c) (3 p)

```
def main():
    lista = [10, 20, 30, 60, 40]
    tulos = 100
    i = 0
    while i < len(lista):
        if tulos > 50:
            tulos = tulos - lista[i]
        i = i + 1
    print(tulos)

main()
```

d) (5 p)

```
def muuta(luku, luvut):
    luvut[1] = 115
    luku = 46
    return luku

def main():
    laskuri = 33
    lista = [10, 20, 30]
    vaihda(laskuri, lista)
    print(laskuri)
    for alkio in lista:
        print(alkio)

main()
```

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

```
def mysteeri1(lista1, lista2):
    tulos = []
    i = 0
    while i < len(lista1):
        if lista1[i] < lista2[i]:
            tulos.append(lista1[i])
        else:
            tulos.append(lista2[i])
        i += 1
    return tulos
```

f) Funktiolle annetaan ensimmäisenä parametreina merkkijono ja toisena parametrina positiivinen kokonaisluku. (4 p)

```
def mysteeri2(merkkijono, luku):
    if luku > len(merkkijono):
        return merkkijono + "*" * (luku - len(merkkijono))
    else:
        return merkkijono[0:luku]
```

g) Funktiolle annetaan ensimmäisenä parametrina positiivisia kokonaislukuja sisältävä lista ja toisena parametrina positiivinen kokonaisluku (5 p)

```
def mysteeri3(lista, luku):
    i = 0
    while i < len(lista):
        if (lista[i] % luku) != 0:
            return False
        i += 1
    return True
```

2. a) Olet hankkimassa mobiililaajakaistaa kannettavaan tietokoneeseesi. Tarvitset myös nettitikun. Eräs operaattori antaa normaalikuukausihinnasta 5 %:n alennuksen, jos teet vähintään 12 kuukauden sopimuksen. Tämän alennuksen lisäksi operaattori antaa nettitikun ilmaiseksi, jos teet vähintään 24 kuukauden sopimuksen. Muussa tapauksessa joudut ostamaan nettitikun itse. Mietit, miten pitkä sopimus sinun kannattaisi tehdä. Kirjoita valinnan tueksi Python-ohjelma, joka pyytää käyttäjältä normaalikuukausihinnan, nettitikun hinnan ja sopimusajan kuukausina. Ohjelma laskee ja tulostaa mobiililaajakaistan hinnan kuukautta kohti, kun sekä kuukausimaksu (mahdollisine alennuksineen) että nettitikun hinta (ostohinta jaettuna sopimuskuukausien määrällä) otetaan huomioon. Jos operaattori antaa nettitikun ilmaiseksi, ei nettitikun hintaa lasketa kustannuksiin. (10 p.)

   b) Eräs yritys maksaa myyjilleen palkkaa 50 euroa työpäivältä. Jos myyjän päivän myynti (euroina) ylittää yrityksen määräämän rajan, myyjä saa tältä päivältä mainitun peruspalkan lisäksi lisäpalkkaa 15 % sen päivän myynnistä. Oletetaan, että myyjän eri päivien myyntitiedot on tallenettu listaan niin, että listan 1. alkiona on 1. päivän myynti, 2. alkiona 2. päivän myynti jne. Kirjoita Python-funktio `laske_palkkio(paivamyynnit, raja)`, joka saa parametreina erään myyjän päivittäiset myynnit sisältävän listan ja yrityksen määräämän rajan lisäpalkan antavalle myynnille. Funktio laskee ja palauttaa myyjälle eri päiviltä yhteensä kuuluvan palkkion. Vastauksena on siis yksi desimaaliluku (eri päivien palkkioiden summa). Kirjoita vain pyydetty funktio, älä muita ohjelman osia. (20 p)

3. Eräällä liikkeellä on tekstitiedostossa tieto sen myymistä tuotteista, niiden normaalihinnoista ja alennusprosenteista. Yhden tuotteen tiedot on tiedoston yhdellä rivillä niin, että tuotteen nimi, normaalihinta ja alennusprosentti on erotettu toisistaan puolipisteellä. Tiedoston rivit voisivat näyttää esimerkiksi seuraavilta:

```
Kattila Hackman;40.0;15.0
Hiusharja;8.0;5.0
Uusi Cola;2.0;20.0
```

   Kirjoita Python-ohjelma, joka pyytää käyttäjältä tiedot sisältävän tiedoston nimen. Ohjelma lukee tämän tiedoston ja tulostaa niiden tuotteiden nimet ja alennetut hinnat, joiden alennusprosentti on vähintään 10. Jos tuotteen alennusprosentti on alle 10, tuotteesta ei tulosteta mitään tietoja. Yllä olevassa esimerkkitapauksessa ohjelman tulostus olisi

```
Kattila Hackman 34.0
Uusi Cola 1.6
```

   Ohjelman on käsiteltävä seuraavat virhetilanteet:
   - Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä
   - Tiedoston jollain rivillä tuotteen hinnan tai alennusprosentin 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 puolipisteellä erotettua osaa. Ohjelman ei tarvitse osata käsitellä esimerkiksi sellaisia virhetilanteita, joissa rivi on tyhjä tai ei sisällä tuotteen nimen lisäksi muuta tekstiä. (20 p)

VIIMEINEN TEHTÄVÄ SEURAAVALLA SIVULLA

4. Eräs yritys haluaa panna www-sivulleen maksullisen pelin, jota halukkaat käyttäjät voivat pelata. Sivulla pidetään myös tilastoa pelaajista ja heidän ennätyksistään. Jotta pelaajan pelaamat pelit vaikuttaisivat näihin tilastoihin, hänen pitää kuitenkin maksaa pelioikeudesta. Kirjoita sivun käyttämää ohjelmistoa varten luokka `Pelaaja` seuraavan kuvauksen mukaisesti.

   `Pelaaja` -oliolla on oltava seuraavat kentät:
   - `__nimi` pelaajan nimi
   - `__pelioikeus` kentän arvo on `True`, jos pelaajalla on voimassa oleva pelioikeus, ja muuten `False`.
   - `__pelilkm` 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 määritellä. 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. Uuden pelaajan pelioikeus on voimassa.
   - `kerro_ennatys(self)` palauttaa pelaajan ennätyksen.
   - `onko_pelioikeus(self)` palauttaa arvon `True`, jos pelaajan pelioikeus on voimassa, ja muuten arvon `False`.
   - `lisaa_pelioikeus(self)` asettaa pelaajan pelioikeuden olevan voimassa.
   - `poista_pelioikeus(self)` muuttaa pelaajan tiedot siten, että pelioikeus ei ole voimassa.
   - `lisaa_peli(self, pisteet)` lisää pelaajalle tiedon uudesta pelistä, jos hänen pelioikeutensa on voimassa. Pelistä saadut pisteet annetaan metodin parametrina. Metodi siis muuttaa pelaajan pelaamien pelien määrää sekä yhteispisteitä. Tarvittaessa metodi muuttaa myös pelaajan ennätystä. Jos pelioikeus ei ole voimassa, metodi ei muuta mitään.
   - `laske_keskiarvo(self)` laskee ja palauttaa pelaajan kaikista peleistä saamien pisteiden keskiarvon. Jos pelaaja ei ole vielä pelannut yhtään peliä, metodi palauttaa arvon 0.0.
   - `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 30 peliä. Muussa tapauksessa pelaaja ei ole mestaripelaaja.
   - `__str__(self)` palauttaa merkkijonon, joka sisältää pelaajan nimen, ennätyksen ja pelaajan pelaamien pelien lukumäärän sekä joko tekstin "pelioikeus on voimassa" tai "pelioikeus ei ole voimassa" sen mukaan, onko pelaajan pelioikeus voimassa.

   Kirjoita lisäksi pääohjelma, joka luo kaksi `Pelaaja`-oliota, kutsuu molemmille kaksi kertaa `lisaa_peli`-metodia ja sen jälkeen toiselle pelaajista myös `kerro_ennatys`-metodia. Pääohjelman pitää myös tulostaa metodin palauttama ennätys. Tämän jälkeen pääohjelman pitää selvittää toisesta pelaajasta, onko hän mestaripelaaja ja tulostaa joko "Pelaaja on mestaripelaaja" tai "Pelaaja ei ole mestaripelaaja" selvityksen tuloksen mukaisesti. Sitten ohjelman pitää poistaa toisen pelaajan pelioikeus. Lopuksi pääohjelman on tulostettava molemmista pelaajista nimi, ennätys, pelattujen pelien lukumäärä ja tieto siitä, onko pelaajan pelioikeus voimassa. Voit päättää luotavien pelaajien nimet ja pelien pisteet itse. Pääohjelman ei siis tarvitse kysyä mitään käyttäjältä. Voit kirjoittaa pääohjelman valintasi mukaan niin, että se on joko samassa moduulissa luokan kanssa tai sitten niin, että se on eri moduulissa.(25 p)