

CS-A1111 Ohjelmoinnin peruskurssi Y1, Tentti 12.8.2019

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 tarvi lisätä kommentteja. Missään tehtävässä tulostusta ei tarvitse muotoilla. Voit myös olettaa, että käyttäjän antama syöte on virheetöntä, ellei tehtävässä erikseen käsketä käsittelemään virhetilanteita.

Opiskelijat, joiden äidinkieli ei ole suomi, saavat käyttää sanakirjaa, jos siinä ei ole merkintöjä (tentinvalvoja tarkistaa sanakirjan). Nämä opiskelijat saavat halutessaan sekä suomen- että englanninkielisen kysymyspaperin.

Tentissä ei saa käyttää laskimia eikä lisämateriaalia.

1. Funktiot ja tulostus (25 p)

Kaikki ohjelmat ja funktiot toimivat jollain tavalla eli niistä minkään suoritus ei keskeydy.

Kerro, mitä seuraavat ohjelmat (kohdat a-d) tulostavat.

a) (2p)

```
def main():
    price = 600
    discount = 200
    new_price = price - discount = 400
    discount = 300
    print(new_price)
```

main()

b) (2p)

```
def main():
    bacteria = 8
    if bacteria < 10:
        print("No cyanobacteria. Water quality is excellent for swimming.")
    if bacteria > 100:
        print("Plenty of cyanobacteria. Do not swim!")
    else:
        print("Some cyanobacteria. Go shower after swimming.")
```

main()

c) (3p)

```
def main():
    list1 = [20, 10, 40, 50, 12]
    result = 0
    i = 0
    while i < len(list1) and list1[i] > 15:
        if list1[i] < 45:
            result = result + list1[i]
        i = i + 1
    print(result)
```

main()

$list[0] = 20$

$list[1]$

1) $result = 0 + 20 = 20$
 $i = 1$

2)

$i = 1$

3. Tiedostot ja virhetilanteet (20 p)

Lapsiperhe on lähdössä lomamatkalle, ja kaikki perheenjäsenet ovat saaneet ehdottaa mieluisia kohteita. Kohdetta valitessaan perhe haluaa tehdä vertailua eri kohteisiin menemisen ympäristökuormituksesta. Maaseudun tulevaisuus -lehdestä he huomasivat oheisen taulukon, jossa on eri kulkuneuvojen 400 kilometrin matkalla (vastaa väliä Helsinki-Tukholma) tuottamat hiilidioksidipäästöt. Avoimesta datasta he ovat löytäneet etäisyyksiä eri kohteisiin eri kulkuvälineitä käyttäen (ohessa).

	nro	kulkuväline	CO2/400km	kaupunki	linnuntie	autotie	junarata
4	1	juna	1 kg	Tallinna	80	760	840
3	2	bussi	26 kg	Tukholma	400	1760	1750
	3	sähköauto	54 kg	Rovaniemi	700	810	830
	4	polttomootoriauto	88 kg	Billund	1050	2540	2660
2	5	potkurilentokone	97 kg	Rooma	2200	3550	3830
←	6	laiva (autolautta)	136 kg	Peking	6300	8830	10070
2	7	suihkulentokone	207 kg	Arusha	7150	11510	-
				Rio de Janeiro	11100	-	-

Kohteiden etäisyytiedot Helsingistä on tallennettu pilkuilla eroteltuina tiedostoon, joka näyttää jotakuinkin seuraavalta:

1,Tallinna,80,760,840
1,Tukholma,400,1760,1750
0,Arusha,7150,11510,-1

eli ensin on numero 1 tai 0, riippuen onko kohde meren rannalla vai ei, sitten kaupungin nimi, sitten etäisyys linnuntietä, autotietä ja junarataa pitkin. Meren rannalla ovat Tallinna, Tukholma ja Rio de Janeiro. Jos etäisyydeksi autotietä tai junarataa pitkin on merkitty -1, kulkureittiä ei ole. Voit olettaa, että lentokoneet kulkevat linnuntietä, ja laivalla pääsee rannalla oleviin kohteisiin linnuntietä. Kukin kohde on tiedostossa vain yhden kerran.

Kirjoita Python-ohjelma, joka pyytää käyttäjältä etäisyydet sisältävän tiedoston nimen. Ohjelma lukee etäisyydet tästä tiedostosta ja tulostaa kuhunkin kohteeseen eri kulkuvälineillä pääsyn tuottamat hiilidioksidipäästöt laatimasi otsikkorivin jälkeen. Jos kulkuvälineellä ei pääse kohteeseen, tulostetaan merkki "-". Kulkuvälineet esitetään sarakkeina samassa järjestyksessä kuin ne ovat yllä olevassa taulukossa. Kohteiden järjestystä ei tarvitse muuttaa, vaan voit ne tulostaa samassa järjestyksessä kuin ne ovat tiedostossa. (Tulostuksen muotoilua esim. tiettyyn sarakkeen reunaan tasattuna ei tarvitse kuitenkaan tehdä, kunhan tiedot ovat annetussa järjestyksessä.)

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu, ja
- Tiedoston jollain rivillä etäisyyden paikalla ei ole kokonaisluku.

Näistä 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 viisi toisistaan pilkulla erotettua osaa. Ohjelman ei siis tarvitse osata käsitellä esim. sellaisia virheitä, joissa rivi on tyhjä tai ei sisällä kaupungin nimen lisäksi etäisyyksiä.

4. Olio-ohjelmointi (25 p)

Olet kilpailutoiminnasta vastaavana eräässä harrastuskoirayhdistyksessä, ja tehtäviisi kuuluu kilpailurekisterin ylläpito. Edeltäjäsi jäljiltä koirien rekisterin tiedot ovat puutteelliset, joten päätät kirjoittaa ohjelman rekisteritietojen ylläpitämiseksi ja kilpailumaksujen keräämiseksi. Yhdistys järjestää sekä harrastus- että kilpatoimintaa. Kilpailuun osallistuvilla koirien kohdalla

pitää maksaa kilpailuoikeudesta. Yhdistyksen rekisterissä ihmiset ovat sivuseikka, sillä rekisteriin tallennetaan koirakohtaiset tiedot. Kirjoita siis ohjelmistoa varten luokka Dog seuraavan kuvauksen mukaisesti (lue myös metodit, koska ne kertovat millaisia tietorakenteita tarvinnut):

Dog-oliolla on oltava seuraavat kentät:

- `__name` koiran virallinen nimi
- `__nickname` koiran kutsumanimi
- `__owner` koiran omistajan nimi
- `__licence` on True, jos koiralla on voimassa oleva kilpailuoikeus, ja muuten False
- `__number_of_competitions` kilpailujen lukumäärä
- `__total_points` koiran kaikista kilpailuista saamat pisteet yhteensä
- `__record` koiran ennätyspisteet (suurin kilpailuista saatu pistemäärä)

Määrittele luokkaan seuraavat metodit. Jos metodin kuvauksessa ei ole kerrottu mitään metodin palauttamasta arvosta, metodin ei tarvitse palauttaa mitään. (Muitakin metodeja olisi järkevä määrittellä, mutta ne eivät ole tässä tenttitehtävässä tarpeen.)

- `__init__(self, name, nickname, owner)` luo uuden Dog-olion. Luotavan koiran perustiedot annetaan metodin parametrina. Aluksi sillä ei ole kilpailulisenssiä eikä tuloksia.
- `get_record(self)` palauttaa koiran ennätyksen.
- `has_licence(self)` palauttaa arvon True, jos koiran kilpailuoikeus on voimassa, muuten arvon False.
- `add_licence(self)` asettaa koiran kilpailuoikeuden voimaan.
- `remove_licence(self)` poistaa koiran kilpailuoikeuden voimassaolon.
- `add_competition(self, points)` lisää koiralle tiedon uudesta kilpailusta, jos koiran kilpailuoikeus on voimassa. Kilpailusta saadut pisteet annetaan metodin parametrina. Metodi muuttaa sekä koiran kilpailujen määrää että yhteispisteitä sekä tarvittaessa myös ennätyspisteitä. Jos kilpailuoikeus ei ole voimassa, metodi ei muuta mitään.
- `calculate_average(self)` laskee ja palauttaa koiran kaikista kilpailuista saaman pisteiden keskiarvon. Jos koira ei ole ollut kilpailuissa, metodi palauttaa arvon 0.0.
- `is_better(self, other_dog)` palauttaa arvon True, jos tämän koiran ennätys on parempi kuin parametrina annetun toisen Dog-olion. Jos ennätys on sama, tarkastellaan kilpailujen keskiarvoja. Jos molemmilla koirilla on sekä sama ennätys että keskiarvo, koirat ovat yhtä hyviä. Tällöin palautetaan False, samoin kun jos tämä koira on huonompi.
- `__str__(self)` palauttaa merkkijonon, joka sisältää pilkuilla erotellusti koiran virallisen ja lempinimen sekä omistajan nimen, ennätyksen ja koiran käymien kilpailujen lukumäärän sekä joko tekstin "licence is valid" tai "licence is not valid" sen mukaan, onko koiran kilpailuoikeus voimassa.

Kirjoita lisäksi pääohjelma, joka luo kaksi Dog-oliota, kutsuu molemmille ensin add_licence-metodia, sitten kaksi kertaa metodia add_competition ja sen jälkeen ensimmäiselle koiralle get_record-metodia. Pääohjelman pitää myös tulostaa metodin palauttama ennätys. Tämän jälkeen pääohjelma selvittää metodin is_better avulla, onko ensiksi luotu koira parempi kuin toiseksi luotu koira, ja tulostaa tämän mukaan joko "ensimmäinen koira on parempi" tai "ensimmäinen koira ei ole parempi". Lopuksi pääohjelman on tulostettava molempien koirien nimi, ennätys, kilpailujen lukumäärä ja tieto siitä, onko koiran kilpailuoikeus voimassa. Voit päättää luotavien koirien ja kilpailujen pisteet itse. Pääohjelman ei siis tarvitse kysyä mitään käyttäjältä. Voit kirjoittaa pääohjelman valintasi mukaan joko samaan moduuliin luokan kanssa tai eri moduuliin.

