

# Programmeerimise 1. vaheeksami järeltöö

Kolmapäev, 12.01.2015, kell 10:15 – 12:00

## 1. Sõnade ühendamine (5p) *yl1.py*

Kirjuta funktsioon `ühenda`, mis võtab argumendiks sõnade listi, ja tagastab sõne, mis on moodustatud listis olevate sõnade tühikutega ühendamisel. Kui listis on kusagil järjest kaks või rohkem samasugust sõnet, siis tulemuses tuleb seda sõnet selles kohas kasutada ühekordselt.

Näited:

```
>>> ühenda([])
''
>>> ühenda(['aaa'])
'aaa'
>>> ühenda(['aaa', 'bbb'])
'aaa bbb'
>>> ühenda(['aaa', 'aaa', 'aaa', 'bbb', 'aaa'])
'aaa bbb aaa'
>>> ühenda(['', 'aaa '])
' aaa '
```

## 2. Olulised tehingud (10p) *yl2.py*

Failis *tehingud.txt* (kodeering UTF-8) on ühe pangakonto tehingute andmed. Tehingute summa võib olla antud eurodes (EUR), USA dollarites (USD) või Suurbritannia naelsterlingites (GBP). Kirjuta programm, mis kuvab ekraanile info oluliste tehingute kohta. Olulisteks tehinguteks loeme 500 eurost suuremaid väljaminekuid (negatiivseid summasid) ja 100 eurost suuremaid sissetulekuid (positiivseid summasid). USD ja GBP tehingute puhul tuleb seejuures arvestada järgnevaid kurse

$$1 \text{ EUR} = 1.16 \text{ USD} = 0.77 \text{ GBP}.$$

Näide. Kui *tehingud.txt* sisu on

```
Tehingu kuupäev;Summa;Teise osapoolle konto;Kommentaar
31.12.2014;870.0 EUR;76889239845;Detsembri palk
01.01.2015;-5.0 USD;3987934123334;Amazon order 32343451
05.01.2015;-581.0 USD;3987934123334;Amazon order 32343894
12.01.2015;80.0 EUR;92398223453;Laen
12.01.2015;40.0 EUR;33339122393;Laen
12.01.2015;240.0 EUR;234822672373;Laen
12.01.2015;-383.0 GBP;3457934123441;Amazon UK order 876234
```

siis programm peab väljastama

```
31.12.2014, 870.0 EUR, Detsembri palk
05.01.2015, -500.86 EUR, Amazon order 32343894
12.01.2015, 240.0 EUR, Laen
```

**NB! Väljundis peavad kõik summad olema antud eurodes, ümardatuna 2 komakohani.**

Programmis peab olema defineeritud ja kasutatud funktsioon `oluline_tehing`, mis võtab argumentideks tehingu summa (ujukomaarvuna) ja valuuta tähise, ning tagastab `True` või `False` vastavalt sellele, kas tehing on eeltoodud kriteerimite järgi oluline või mitte. Näide

```
>>> oluline_tehing(-550, "EUR")
True
>>> oluline_tehing(75, "GBP")
False
```

## 3. ülesanne on lehe pöördel

---

Programmide peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). **Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhu jaoks. Kui funktsiooniga ei oska, siis tee ilma.** Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.

### 3. Pojad ja tütreid (3+2p) yl3.py

Failis *lapsed.txt* on kirjas lapsevanemate ja nende laste nimed.

Iga pere kohta on kõigepealt rida isa ja ema nimedega. Sellele järgneb rida, kus komaga eraldatuna on kirjas kõigepealt nende poegade arv ja seejärel nende tütarde arv. Seejärel tulevad poegade ja tütarde nimed, iga nimi eraldi real.

#### Ülesande põhiosa (3p)

Kirjuta programm, mis kuvab failis *lapsed.txt* olevate tütarde nimed selles järjekorras, nagu need failis esinevad.

Näide. Kui faili *lapsed.txt* sisu on

```
Aadu Allikas ja Marit Mustikas
2,1
Madis Loosaar
Juhan Mustikas
Mirjam Allikas
Peeter Tulp ja Triin Tulp
0,0
Hans Rooste ja Kairi Rooste
0,1
Annabell Rooste
Tauri Hiie ja Siiri Klumm
1,0
Priit Hiie
Georg Eirit ja Ulla Eirit
1,1
Madis Eirit
Piret Eirit
```

siis programmi käivitamisel peab ekraanile ilmuma

```
Mirjam Allikas
Annabell Rooste
Piret Eirit
```

Võib eeldada, et kõik autonumbrid ja omanike nimed on pikemad kui 3 sümbolit.

#### Ülesande lisa (2p)

Täispunktide saamiseks tuleb lisaks väljastada tühi rida ja seejärel kõik erinevad poegade eesnimed. Võib eeldada, et kõik nimed koosnevad ühest eesnimest ja ühest perenimest. Näidisfaili kasutamisel peaks programmi põhiosa ja lisa kogu väljund olema

```
Mirjam Allikas
Annabell Rooste
Piret Eirit

Madis
Juhan
Priit
```