Kontrolltöö 1 aines *Objektorienteeritud programmeerimine*

Automaattestimise võimaldamiseks peavad kõik klassid asuma kindlas paketis ja kõik Java failid peavad olema kindla kodeeringuga. Antud juhul lepime kokku, et klassid asuvad vaikepaketis (st failide alguses ei ole `package` direktiivi) ja failide kodeering on UTF-8.

Toidukullerifirma pakub klientidele võimalust erinevaid toite tellida. Toidud on esitatud failis järgmisel kujul:

```
Kanasalat,5.00,10
Shishi sushi,19.00
Kartulivorm,6.50,15
Pardikoib,11.50
```

Iga rida algab toidu nimetusega. Nimetusele järgneb hind. Kui tegu on kampaaniapakkumisega, siis nimetusele ja hinnale järgneb soodustuse protsent. Eraldajaks on koma.

Kontrolltöö seisneb toitude tellimusi käsitleva programmi koostamises. Programm peab vastama alltoodud nõuetele (isegi kui need kummalised tunduvad).

Programm peab sisaldama klasse `Toit`, `KampaaniaToit`, `Kuller`, `Tellimus`, `Klient` ning peaklassi. Peaklassis loetakse sisse toidud ja koostatakse tellimused. Peaklassis testitakse ka erinevate isendimeetodite tööd. Kõikide klasside kõik isendiväljad peavad olema privaatsed.

1. (2 p) Klassis `Toit` peavad olema privaatsed isendiväljad toidu nimetuse (`String`) ja hinna (`double`) jaoks.
   1. Klassis peab olema kahe parameetriga konstruktor isendiväljade määramiseks.
   2. Klassis peavad olema meetod `getHind` (`get`-meetod isendivälja jaoks).
   3. Klassis peab olema ka meetod `toString` toidu info tekstina esitamiseks, näidates lisaks toidu nimetusele ka hinda kasutades `get`-meetodit.
   4. Klass `Toit` peab realiseerima liidese `Comparable<Toit>`, kusjuures `compareTo` meetod realiseeritakse nii, et võrdlemine toimub hinna (`get`-meetodi) alusel.

2. (2 p) Klass `KampaaniaToit` on klassi `Toit` alamklass. Ülemklassis olemasolevaid isendivälju siin uuesti mitte kirjeldada. Lisaks peab olema privaatne isendiväli soodustuse protsendi (`int`) jaoks.
   1. Klassis peab olema kolme parameetriga konstruktor, mille abil saab määrata toidu nimetuse, hinna ja soodustuse protsendi.
   2. Klassis peab olema meetod `getHind`, mis arvutab hinna arvestades soodustuse protsendiga (valem: *hind*(100-protsent)/100*).

3. (3 p) Klassis `Kuller` peavad olema privaatsed isendiväljad kulleri nime ja temale antud hinnangute nimekirja (`List<Integer>`) jaoks.
   1. Klassis peab olema ühe parameetriga konstruktor, mille abil saab määrata kulleri nime.

2. Äsjaloodud kulleril ei olegi ühtegi hinnangut. Hinnangute lisamiseks peab olema `void`-tüüpi meetod `lisaHinnang`, mis jätab argumendiks antud hinnangu (täisarvu) meelde.
3. Klassis peab olema `double`-tüüpi parameetriteta meetod `keskmineHinnang`, mis arvutab hinnangute keskmise.
4. Klassis peab olema ka meetod `toString` kulleri nime ja keskmise hinnangu tekstina esitamiseks.

4. (3 p) Klassis `Tellimus` peavad olema privaatsed isendiväljad kulleri (`Kuller`) ja tellitud toitude nimekirja (`List<Toit>`) jaoks.

1. Klassis peab olema ühe parameetriga konstruktor kulleri määramiseks.
2. Äsjaloodud tellimuses ei olegi ühtegi toitu. Toitude lisamiseks peab olema `void`-tüüpi meetod `lisaToit`, mis jätab argumendiks antud `Toit`-tüüpi isendi meelde.
3. Klassis peab olema `Kuller`-tüüpi parameetriteta meetod `getKuller`.
4. Klassis peab olema `double`-tüüpi parameetriteta meetod `summa`, mis tagastab tellimuse kõikide toitude kogumaksumuse.
5. Klassis peab olema ka meetod `toString` tellimuse kogumaksumuse ja tellitud toitude arvu tekstina esitamiseks.

5. (2 p) Klassis `Klient` peavad olema privaatsed isendiväljad kliendi nime (`String`) ja olemasoleva rahasumma (`double`) jaoks.

1. Klassis peab olema kahe parameetriga konstruktor isendiväljade määramiseks.
2. Klassis peab olema `void`-tüüpi meetod `maksa`, mis võtab parameetrina `Tellimus`-tüüpi isendi ja vähendab olemasoleva rahasumma tellimuse kogumaksumuse võrra.
3. Klassis peab olema `void`-tüüpi meetod `hinda`, mis võtab parameetrina `Tellimus`-tüüpi isendi ja lisab selle tellimuse toonud kullerile hinnangu. Hinnang genereerida juhuslikult lõigust [2, 5] (täisarvuna).
4. Klassis peab olema ka meetod `toString` kliendi nime ja olemasoleva rahasumma tekstina esitamiseks.

6. (4 p) Peaklass peab olema nimega `Peaklass`. Klassis peab olema staatiline avalik meetod `loeToidud` tagastustüübiga `List<Toit>`, mis võtab argumendiks failinime ja tagastab selles failis olevad toitude andmed. Meetod võib visata erindi (st meetodi signatuuris võib olla `throws Exception`). Toitude faili formaat on ülalpool toodud. Toitude arv failis ei ole teada (programm peaks töötama suvalise arvu toitudega). Kui failist lugemist ei õnnestu programmeerida, siis kirjutage selles meetodis vastav list programmi sisse (vähendab tulemust 2 punkti võrra).

Peameetodis tehakse järgmised tegevused.

1. Rakendatakse vastavat staatilist meetodit, et lugeda failist toitude andmed.
2. Sorteeritakse toidud vastavalt meetodis `compareTo` kirjeldatud järjekorrale.
3. Toitude info väljastatakse ekraanile.
4. Luuakse 4 klienti rahasummaga 100 (nimed mõtelge ise välja).
5. Kõikidest klientidest tehakse `Klient`[]-tüüpi massiiv. (Massiivi võib ka enne klientide tegemist luua ja järjest täita.)

6. Luuakse 2 kullerit (nimed mõtelge ise välja).
7. Luuakse 4 tellimust (2 ühe kulleriga ja 2 teise kulleriga).
8. Kõikidest tellimustest tehakse `Tellimus`[]-tüüpi massiiv. (Massiivi võib ka enne tellimuste tegemist luua ja järjest täita.)
9. Iga tellimuse jaoks valitakse juhuslikult 3 toitu (võib eeldada, et failis on vähemalt 3 toitu). Selleks tuleb kasutada `Collections.shuffle` meetodit. Antud meetod võtab argumendiks listi ning järjestab selle suvalises järjekorras. Toitude list järjestada iga tellimuse jaoks uuesti ümber ning lisada tellimusse 3 esimest toitu.
10. Tellimuste info väljastatakse ekraanile.
11. Kliendid maksavad tellimuste eest ja hindavad kullereid (üks klient maksab ühe tellimuse eest ja hindab selle tellimuse toonud kullerit).
12. Klientide info väljastatakse ekraanile.
13. Kullerite info väljastatakse ekraanile.

Programmi väljund peab olema arusaadav ja loetav.

Andmete fail on aadressil http://kodu.ut.ee/~marinai/toidud.txt. Salvestage see oma arvutisse. Fail on kodeeringus UTF-8.

Kontrolltöö ajal on Moodle'is kättesaadav automaatne test, mis kontrollib, kas lahendus sisaldab nõutud komponente. Meetodite sisu see test ei kontrolli.

Palun esitada viimane töötav versioon! Palun esitada kahel viisil:

1. e-postiga helle.hein@ut.ee, Subject: OOP KT1