

Reeglid:

- Kasuta puhta koodi printsiipe.
- Tee ise! Plaigaat on läbikukkumine. Tee vähem aga tee ise. Void küsida ka mõne keerukama koha jaoks abi aga pane kirja ainult selline kood, millest sa saad aru!
- **Lisa ühikteste: vähemalt 20.**
- Planeeri aega. Ülesanne pole (vähemalt mitte hästi) tehtav ühe õhtuga.
- Välti koodi duplikatsiooni.
- Lahenduse tüüp: .net core konsoolirakendus.
- Baasosa annab max 24p, keerukam lisaosa 4p.
- Enne tegema hakkamist mõtle läbi: kui vaatad kuidas ringkonna ja isikumandaate antakse, siis arvutamine ja loogika pole keeruline tegelikult (aga tahab natuke süvenemist). Kõik andmed on olemas ning testitulemusi/näiteid/oodatud tulemusi on kerge mõelda: selle põhjal saad teha ka teste.

Rules:

- Use clean code principles.
- Do it yourself! Coming with code that is not written by you results in a failure! Do less but do it yourself. You can ask help with some parts but write down only code that you can understand.
- **Add unit tests. Add at least 20 tests. There is lots of data and you can create data for yourself for testing based on this.**
- Plan enough time! You can not do it in one evening.
- Avoid code duplication.
- Solution type: .net core console app.
- Base part gives you 24p; advanced is more difficult and gives 6p.

2023. aastal on järgmised riigikogu valimised. Olete palgatud looma programmi, mis valimistulemuste alusel leiab valimiste võitjad, jaotab kohad riigikogus erakondade vahel ning leiab isikumandaadid saanud kandidaadid.

Next parliament elections will be held at 2023. You are hired to create a program which based on election results finds winners, places in parliament between parties and finds candidates who got personal mandates.

Märkused:

- Näited ringkonna- ning isikumandaatide leidmiseks. Näited nii eesti-, vene- kui inglise keeles.
 - <https://www.riigikogu.ee/tutvustus-ja-ajalugu/riigikogu-ulesanded-ja-tookorraldus/riigikogu-valimine/>
 - <https://www.valimised.ee/et/valimiste-meelespea/valimistulemuste-kindlakstegemine-riigikogu-valimistel>
- Kuna me kompensatsioonimandaate ei arvesta, siis riigikogus võivad mõned kohad tühjaks jääda (valitud kandidaatide arv olla alla 101). Suurusjärgud antud aasta parlamendivalimiste tulemuste ning erakonna kohtade arvuga peaksid klappima kuid täpsed numbrid mitte.

- Partei nime puhul on mõnes failis tema terviknimi ning mõnes kohas lühend, näiteks REF – Eesti Reformierakond. Väljatrüki oleks hea kasutada partei pikka nime.
- Eesti on jaotatud 12ks valimisringkonnaks. Valimisringkondade info ning iga ringkonna mandaatide arv (seda ringkonda parlamendis esindavate kohtade arv) on failis *ElectionRegions.txt*. Väljatrüki kasuta võimalusel lisaks valimisringkonna numbrile ka tema nime, näiteks kujul: 8 (Järva- ja Viljandimaa).
- Partei tulemustes (osad 1 ja 2) on üksikkandidaatide häälte arvu vaja ainult selleks, et leida kogu häältearvu. Üksikkandidaadid parteina arvesse ei lähe.

Notes:

- Examples for finding election winners and explanations about voting: in English, Estonian and Russian.
 - <https://www.riigikogu.ee/tutvustus-ja-ajalugu/riigikogu-ulesanded-ja-tookorraldus/riigikogu-valimine/>
 - <https://www.valimised.ee/et/valimiste-meelespea/valimistulemuste-kindlakstegemine-riigikogu-valimistel>
- We do not take into consideration compensation mandates: thereby our number of elected parties/people by our system can be less than 101 and a bit different to official results.
- Party names: in some places there are abbreviations and in some places full names. In print methods it would be good to use full names (REF = Eesti Reformierakond etc)
- Estonia is divided into 12 electoral districts. District info and number of mandates (places in parliament for each district) is in file *ElectionRegions.txt*. Use district full name when printing out info.
- When finding results for political parties (part 1 and 2), the number of votes for single candidates (*üksikandidaadid*) is necessary only for getting total number of votes. Single candidates do not count as parties.

Abiks ka | [More help: https://mandaat.valimised.ee/](https://mandaat.valimised.ee/), <https://m.delfi.ee/rk2019tulemused/> (2019 a tulemused koos näidetega)

Baasosa: 24p | Base: 24p

Osa 1: Üldine riigikogu valimistulemuste analüüs. Aluseks failid *ElectionResultsYEAR.txt*. Antud failides on üldine info erinevate aastate valimistulemuste kohta.

Part 1: General parliament election analysis. Use files *ElectionResultsYEAR.txt* as basis. These files contain general information about election results from different years.

Tee meetodid :| [Create methods:](#)

- 1) Meetod selle aasta valimiskünnise (5% koguhäältest) leidmiseks.
[Method for finding electoral threshold value.](#)
- 2) Meetod leidmaks millised parteid ületasid valimiskünnise ning millised mitte, prindi tulemus (valimiskünnis: erakonnad, kes koguvad üle riigi vähemalt 5% kõikidest häältest).

Method for finding which parties exceeded threshold value (got at least 5% of total votes),
print results

*Näide: kokku anti valimistel 32300 häält. Erakond X sai 500 häält, erakond Y sai 10000 häält. X sai kokku $500 / 32300 * 100 = 1.57\%$ ning ei ületanud valimiskünnist. Erakond Y sai $10000 / 32300 * 100 = 30.96\%$ häältest ehk ületas valimiskünnise 5%.*

- 3) Meetod valimiste võitja (enim hääli saanud erakond) ning valimiste kaotaja (vähim hääli saanud erakond) leidmiseks

Method for finding winner and loser parties (who got most votes and least votes)

- 4) Meetod, mis näitab mitu protsenti hääli läks "raisku" ehk parteidele, kes ei ületanud valimiskünnist (ega saanud riigikokku).

Method for showing the % of votes that were a "waste": votes for parties who did not exceed election threshold and did not make it to the parliament.

- 5) Meetod, omal vabal valikul, mis antud andmete põhjal leiab midagi (nt kui suur oli häälte erinevus võitjal ja kaotajal vms)

Method, on your own free choice, that uses this data and finds smth (example: what was difference of votes between winner and loser etc)

- 6) Meetod, mis võimaldab välja printida kõikide eelpooltoodud meetodite (1-3) tulemused. Meetodi parameetrik on valimiste aasta. Vaata [lehel](#), mis aastatel on riigikogu valimised toimunud. Kui kasutaja annab aasta, kus valimisi pole olnud, siis ütle seda. Kui kasutaja annab aasta, kus valimised olid aga mille kohta meil pole andmeid, siis ütle seda. Kui kasutaja annab aasta kus peaks toimuma valimised aga mis on tulevikus, siis ütle seda.

Method which allows to print results from previous methods. This method takes election year as a parameter. Look [here](#) to see in which year parliament (riigikogu) elections took place. If user enters a year when there were no elections happening then print message about it. If user enters a year where election took place but we don't have data about it then print it also. And if user enters a future year, then print notification about it.

DisplayElectionsInfo(2015) -> prindib info 2015a valimiste kohta.

- 7) Meetod, mis 2019 aasta valimistulemuste kohta prindib valimiskünnise ületanud erakondade kohta, et mitu % häältest said koalitsiooniparteid (valitsuse moodustanud erakonnad) ning mitu % opositsiooniparteid. Protsente saad kontrollida saad siit:

<https://www.delfi.ee/news/rk2019/tulemused/> (NB! See meetod on ainult 2019a andmete kohta, teiste aastate mitte.). Hetke koalitsiooniparteid:

<https://www.valitsus.ee/et/peaminister-ministrid/valitsuse-koosseis>

Method, which based on 2019 election results (and this works only for 2019) finds parties that exceeded electoral threshold and what is the % of votes that current coalition (government) parties got and what is the % for opposition parties. Current coalition parties :

<https://www.valitsus.ee/en/objectives-activities>

Osa 2: 2019 valimiste detailsem analüüs; isiku- ja ringkonnamandaatide leidmine. Selle aluseks on failid *ElectionRegions.txt*, kust saab info iga valimisringkonna ning tema mandaatide arvu kohta ning *DetailedResults_2019.txt*, kus on 2019a igas piirkonnas iga erakonna saadud häälte arv.

Part 2: Detailed analysis for 2019 parliament elections. Finding regional and personal mandates: people who got elected to parliament. File *DetailedResults_2019.txt* contains detailed information about results

for every electoral district results. *ElectionRegions.txt* contains information about electoral districts in Estonia and number of mandates (places in parliament for this region, total is 101).

Tee meetodid | Do methods:

- 8) Meetod iga valimisringkonna lihtkvoodi leidmiseks: selleks jagatakse valimisringkonnas hääletanute arv antud ringkonna mandaatide arvuga

Method for finding simple quota for each electoral district: for this divide total number of votes for this district by number of mandates (from *ElectionRegions.txt* file).

Näide: Valimisringkonnas nr 7 (Ida-Virumaa) anti kokku 20000 häält. Antud ringkonna mandaatide arv on 7. Seega lihtkvoot on $20000/7 = 2857.14$.

- 9) Meetod leidmaks, mitu ringkonnamandaati iga erakond kokku sai (eelmises meetodis leitud lihtkvoodi põhjal). Erakond saab nii mitu ringkonnamandaati, kui mitu korda ületab summeeritud häälte arv lihtkvooti.

Method for finding how many district mandates each party got (based on simple quota from previous method). The party shall receive as many mandates as many times the number of the summed-up votes exceed the simple quota. Such mandates are called the district mandates.

Näide: Valimisringkonnas nr 7 (Ida-Virumaa) anti kokku 20000 häält. Antud ringkonna mandaatide arv on 7. Seega lihtkvoot on $20000/7 = 2857.14$. Erakond X sai 5000 häält. $5000 / 2857.14 = 1.75$ ehk 2 koht riigikogus.

NB! Väärtus alates .75 ümardatakse üles: 1.5 -> 1; 1.8 -> 2

- 10) Meetod iga valimisringkonna tulemuste printimiseks (piirkonna nimi, lihtkvoot ning mis erakond kui mitu mandaati sai)

Method for printing results for each electoral district. (Name of district, simple quota and which party got how many places)

Edasijõudnutele: 6p | Advanced: 6p

- 1) Leia isikumandaadid. Selleks kasuta faili "PersonalVotes.csv", kus on iga kandidaadi erakond ning valimistulemused ringkonniti. Isikumandaatide leidmiseks jagatakse valimisringkonnas hääletanute arv mandaatide arvuga. Tulemuseks saadakse ringkonna lihtkvoot. Iga kandidaat, kes saab lihtkvoodiga võrdselt või rohkem hääli, osutubki isikumandaadiga valituks.

Find personal mandates, use file "PersonalVotes.csv", where are listed all candidates, their party, electoral district and number of votes. At first, the number of voters in an electoral district is divided by the number of mandates. This gives the simple quota of the district. Every candidate who received more votes than the simple quota or equally to it, shall be elected, i.e. receives a personal mandate.

Formula: $\text{number of voters} / \text{number of mandates} = \text{simple quota}$

Näide: Näide: Valimisringkonnas nr 7 (Ida-Virumaa) anti kokku 20000 häält. Antud ringkonna mandaatide arv on 7. Seega lihtkvoot on $20000/7 = 2857.14$. Kandidaat XX sai 3000 häält ($3000 \geq 2857.14$) ning seega isikumandaadi. Kandidaat YY sai 500 häält, mis jääb alla lihtkvoodi ning seega ei saa isikumandaati.

Tee antud meetod *overload* meetodiks: [Make this method into overloaded method](#)

- a) *FindPersonalMandates()* ilma parameetrita prindib välja kõikide valimisringkondade isikumandaadi saanud kandidaadid ja nende häälte arvu ning antud ringkonna lihtkvoodi.

Without parameter this method prints out info for all 12 electoral districts.

- b) *FindPersonalMandates(int regionNr)* – võtab parameetriks valimisringkonna numbri ning prindib info ainult konkreetse ringkonna kohta.

With numeric parameter it prints out info for only one electoral district.

- c) *FindPersonalMandates(List<int> regionNrs)* – võtab parameetriks listi piirkona numbritest

Takes list of district numbers as a parameter

- 2) Arvesta ringikonnamandaatide leidmisel (ülesande põhiosa) ka isikumandaate: Kui erakond on ringkonnas juba saanud isikumandaate, arvatakse need ringikonnamandaatide arvust maha. Use personal mandates for finding district mandates (from base). If a party has already received personal mandates in a district, these are subtracted from the number of district mandates.

Mõisted:

- Mandaat on saadikule antud volitus esindada ja kaitsta valijate huve. Mandaat on ka valimisringkonnale eraldatud saadikukoht.
- Valimiskünnis: parlamendivalimistel 5% partei nimekirjale üle riigi antud häältest
- Lihtkvoot on häälte hulk, mille eest antakse nimekirjale üks saadiku mandaat. Leidmine: valimisringkonnas antud häälte arvu ja mandaatide arvu jagatis.

Soovitused:

- Kasuta `Console.WriteLine()` meetodit (või *debugimist*), et veenduda et väärtused on sellised nagu sa arvad. Näiteks kui lisad väärtusi listi, siis prindi need enne listi lisamist välja ja pärast prindi välja kogu list, et näha kas väärtused on õigel kujul.
- Paljudes meetodides on korduvkasutatavaid osasi (näiteks valuutakursi leidmine nime järgi). Katsu need üles leida ning tee eraldi abimeetodid (Selleasemel et sama koodi eraldi meetodite sees duplitseerida).
- Katsu vältida üle 20 realisi meetodeid. Mõned võivad olla aga enamus ei tohiks!
- Iga meetodi juures mõtle ka negatiivsetele tulemustele; näiteks valuutat ei leita. Loo ilusad veateated.
- Testida saab ainult tagastustüübiga meetodeid (void meetodeid mitte).
- Mõtle mida lisada klassimuutujaks (väljaks). Kui andmed on vaja ühe meetodiga paika panna ning teistes kasutada, siis tõenäoliselt on välja vaja. Kui andmeid on vaja ainult ühes meetodis, siis tõenäoliselt mitte.
- Mõtle oma andmestruktuurile; et andmed oleksid igas kohas samal kujul. Tühikud sõna lõpus või erinevas formaadis tähed võivad anda ootamatuid tulemusi. Kasuta igas meetodis ühtset andmekuju, tee lisameetodid andmete kuju ühtlustamiseks.
"kala " == "kala" //false, space in the end of the word
"kala" == "kaLa" //false, one letter is upper case
<https://www.dotnetperls.com/trim>

Recommendations.

- Use `Console.WriteLine()` (or debugging) a lot to see if values are what you expect them to be. Use `string.length`, `list.Count` etc methods to check if your data.
- Some methods have many repeating part (for example finding currency rate by name). Try to find them and create re-usable methods (instead of duplicating same code in different methods).
- Try to avoid methods longer than 20 lines (some are ok but if most of your methods are longer then its not good).
- Think of negative scenarios for all methods also! (Currency not found etc). Create nice error messages.
- Only methods with return type can be tested! (void methods not).
- Think what values to add as a class field (variable). If you set value in one method and need to use it in other methods then create variable. If you need to use value in only one method then no point in creating class variable.
- Think of your data structure. If you compare strings then extra spaces in the end of the word or different casing results in unexpected results. Think of common data format for all methods!
"kala " == "kala" //false, space in the end of the word
"kala" == "kaLa" //false, one letter is upper case
<https://www.dotnetperls.com/trim>