

Sissejuhatus tarkvaraarendusse

Martti Raavel

martti.raavel@tlu.ee

Teine loeng

- Meenutame eelmist loengut
- Konsultatsioon
- Koduste tööde tagasiside
- .gitignore
- Markdown
- Harjutus nõuete väljaselgitamiseks
- Tarkvaraarenduse meetodid
- Koodi kvaliteet

Eelmine loeng

- Tarkvara
- Tarkvaraarendus
- Tarkvaraarenduse elutsükkel
- Git
- Github
- .gitignore - ei jõudnud
- Markdown - ei jõudnud
- Github issue

Kodustest töödest

- Kaustade struktuur!
- Edaspidi kasutame `Homework_03` jne (edaspidi ka koduse töö juures kirjas)
- Enne kodutöö tegemist `merge main -> oma_branch` ([juhend](#))

.gitignore

`.gitignore` on konfiguratsioonifail, mida Git kasutab, et määrata, millised failid ja kataloogid tuleks versioonihaldussüsteemist välja jätta.

Markdown

Markdown on märgendikeel, mis võimaldab kasutajatel kirjutada lihtteksti ja vormindada seda lihtsa süntaksiga, et luua dokumente, mida on lihtne lugeda ja kirjutada.

Nõuete väljaselgitamine

Harjutame Programmeerimine I aines loodava rakenduse jaoks nõuete paikapanemist, prioritseerimist ja disaini.

Kuidas me seda teeme?

Moodustame grupid

Tarkvaraarenduse meetodid

Tarkvaraarendusmeetodid on struktureeritud lähenemisviisid tarkvara arendamiseks, mis loovad raamistiku tarkvaratoodete kavandamiseks, kujundamiseks, ehitamiseks, testimiseks ja tarnimiseks.

Koodi kvaliteet

- Loetavus (*Readability*)
- Järjepidevus (*Consistency*)
- Lihtsus (*Simplicity*)
- Hooldatavus (*Maintainability*)
- Töökindlus (*Reliability*)

Koodi loetavus (*Readability*)

Kood peaks olema kergesti loetav ja arusaadav. See peaks olema hästi struktureeritud ja organiseeritud, järjekindlate nimetamisreeglite, vahede ja taandega.

Koodi järjepidevus (*Consistency*)

Kood peaks olema stiililt ja lähenemisviisilt ühtne. See hõlbustab teistel arendajatel koodi mõistmist ja muutmist.

Koodi lihtsus (*Simplicity*)

Kood peaks olema lihtne ja arusaadav. See vähendab vigade tõenäosust ning muudab koodi aja jooksul muutmise ja hooldamise lihtsamaks.

Hooldatavus (*Maintainability*)

Kood peaks olema aja jooksul hooldatav. See tähendab, et nõuete muutumisel peaks seda olema lihtne muuta, parandada ja värskendada.

Töökindlus (*Reliability*)

Kood peaks olema töökindel ja prognoositav. See peaks töötama ootuspäraselt ja käsitlema vigu selliselt, et need ei põhjusta programmi töös tõrkeid.

Kuidas koodi kvaliteeti tagada?

- Järjepidev koodi vormindamine
- Kommentaarid ja dokumentatsioon
- Ühiktestimine
- Koodiülevaatused
- Vigade käsitlemine
- Koodi refaktoreerimine

Kodune töö

- Endanimelises kaustas README.md lühikese ülevaade oma kodutööde kohta
 - Kuidas ülesande lahendamisele lähenesid
 - Kas oli probleeme
 - Kuidas probleemid lahendasid
- [ESLint](#) rakendamine Programmeerimine I kodutöodes
 - Enne koduse töö üles laadimist `merge main -> koduseTooBranch`
 - Veendu, et test õnnestub
- [Rühmatööna õppematerjali täiendamine](#)