TARTU ÜLIKOOL

Mida räägivad logid programmeerimisülesande lahendamise kohta?

Heidi Meier

09.02.2019

TARTU ÜLIKOOL

Miks on ülesannete lahendamise käigu kohta info kogumine oluline?

- Üha rohkem erinevas eas inimesi õpib programmeerimist.
  - Kuidas nad õpivad?
- Suur osa iseseisval lahendamisel.
  - Õpetajad ei näe, kuidas tulemuseni jõuti.
  - Eriti MOOC-idel.

Võib aidata näha

- lahendamisviiside leidlikkust ja ebaratsionaalsusi.
- millised raskused õppijatel on.
- mis on need teemad, millele õpetamisel rohkem tähelepanu pöörata.

TARTU ÜLIKOOL

Programmeerimiskeskkond Thonny

- Autor Tartu Ülikooli õppejõud Aivar Annamaa.
- Loodud õppimiseks ja õpetamiseks (eelkõige algajatele).
- Logifailid lahendamise käigu jälgimiseks ja detailseks uurimiseks.

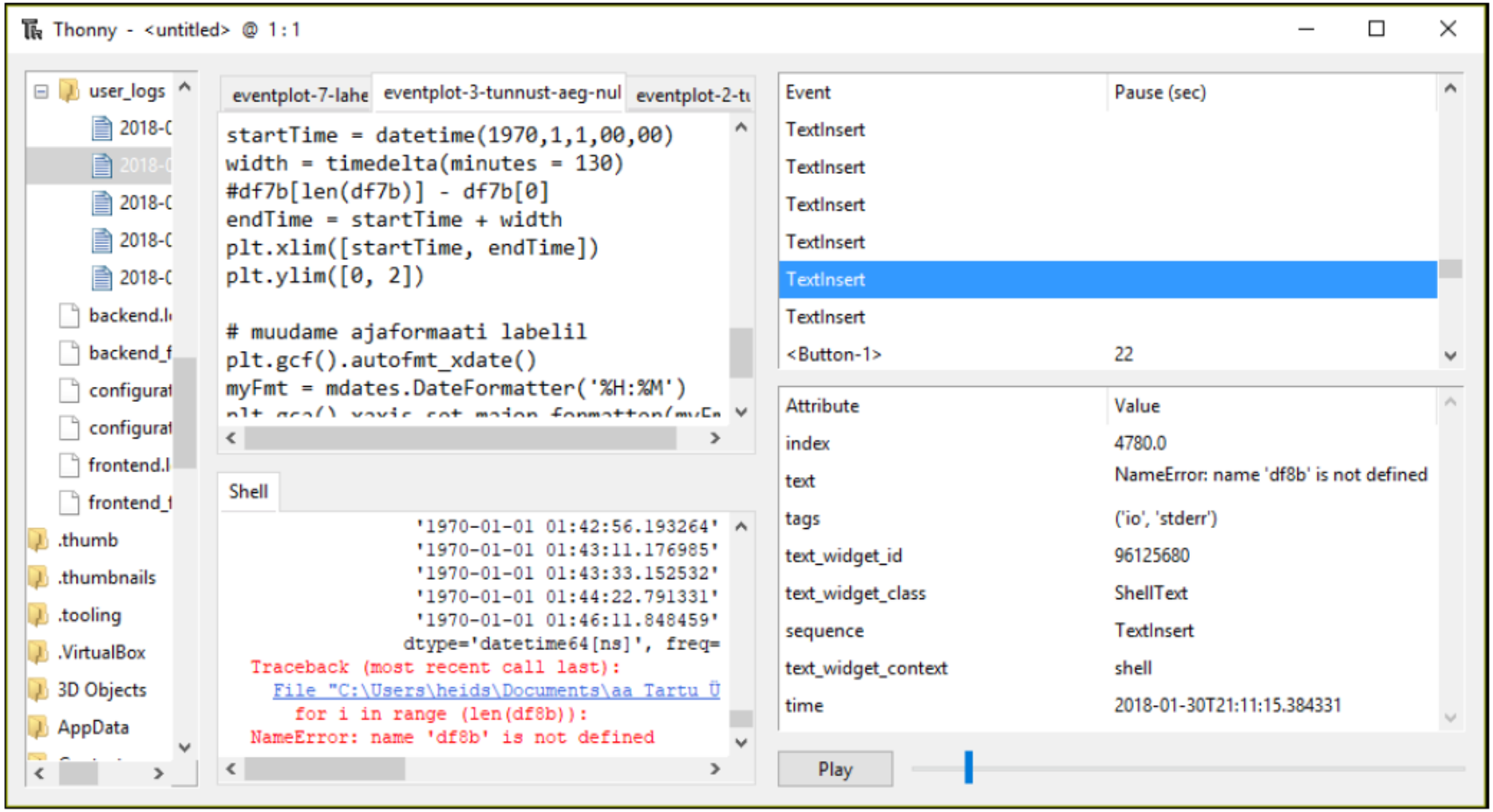
Olulised omadused

- Käsurida koodiredaktoriga samas vaates.
- Käsureal sisend ja väljund visuaalselt selgelt eristuvad.
- Võimalus programmi tööd samm-sammult jälgida ehk siluda (ingl *debug*).
- Võimalus kasutussessiooni taasesitada.

Kasutussessiooni taasesitamine Thonnys

Kuidas?

https://courses.cs.ut.ee/2019/poemp/Main/Periood2Thonnylogid



Mille kohta saab Thonny logidest infot kasutussessiooni taasesitamise abil?

TARTU ÜLIKOOL

- Milline oli programm erinevatel ajahetkedel?
- Millal käivitas lahendaja programmi esimest korda?
- Millised veateated tulid ja millised neist kordusid?
- Kuidas proovis lahendaja programmi parandada?
- **Kas ja kuidas kasutas lahendaja**
  - võimalusi väiksemate programmiosade töö kontrollimiseks?
  - abistava vahendina silumise funktsionaalsust?
- **Kas ja kui palju lahendaja**
  - avas Thonnys varasemate lahenduste faile?
  - käivitas Thonnys teisi faile?

Fragment logifailist

TARTU ÜLIKOOL

2017-12-04_17-01-20_0.txt - Notepad

File Edit Format View Help


```
    {
        "index": "37.0",
        "text": "    fail = open(failinimi, encoding=\"UTF-
8\")\n",
        "tags": "('io', 'stderr')",
        "text_widget_id": 55230736,
        "text_widget_class": "ShellText",
        "sequence": "TextInsert",
        "text_widget_context": "shell",
        "time": "2017-12-04T17:25:31.169461"
    },
    {
        "index": "38.0",
        "text": "NameError: name 'failinimi' is not defined\n",
        "tags": "('io', 'stderr')",
        "text_widget_id": 55230736,
        "text_widget_class": "ShellText",
        "sequence": "TextInsert",
        "text_widget_context": "shell",
        "time": "2017-12-04T17:25:31.169461"
    },
    {
        "index": "39.0",
        "text": ">>> ",
        "tags": "('toplevel', 'prompt', 'vertically_spaced')",
        "text_widget_id": 55230736,
        "text_widget_class": "ShellText",
        "sequence": "TextInsert",
        "text_widget_context": "shell",
        "time": "2017-12-04T17:25:31.169461"
    },
    {
        "widget_id": 70886640,
```

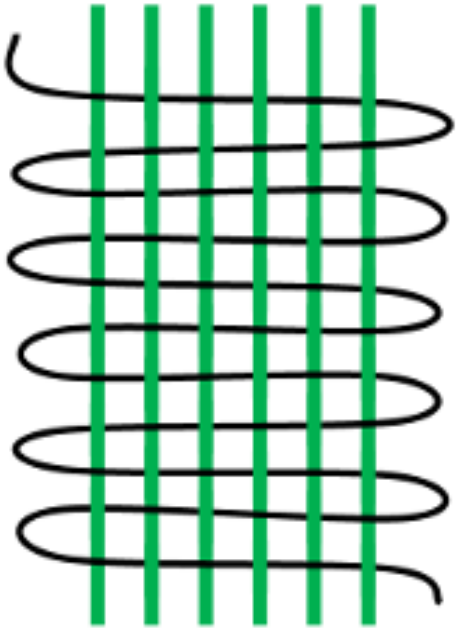

Ajatempliga kirjed erinevate sündmuste kohta, nt:

- käivitamised,
- veateated,
- silumised,
- teksti sisestamised,
- kopeerimised,
- kleepimised.

Ülesanne

Tõenäoliselt on paljud näinud kaltsuvaipu, kuid võib-olla pole mõelnud sellele, kuidas neid valmistatakse. Kaltsuvaipu kootakse spetsiaalsetel telgedel, millele kinnitatakse hulk tugevast materjalist niite ehk lõimi, mille vahele hakatakse põimima kaltsuribasid. Lõimed on järgneval joonisel kujutatud rohelisena. Meie eesmärk on vajaliku lõimeniidi kogupikkuse leidmine. Kuna lõimed on telgedel pinges, siis pärast valmimist tõmbub vaip mõnevõrra kokku. Kokkutõmbumise kompenseerimiseks võetakse vaiba algpikkus soovitavast lõpp-pikkusest 20% suurem. Samuti arvestatakse, et iga lõime kumbagi otsa tuleb jätta sidumiseks 25 cm varu.

Koostada funktsioon **lõimede_pikkus**, mis võtab argumentideks vaiba lõpp-pikkuse (ujukomaarv) ja lõimede arvu (täisarv), arvutab ja tagastab vaiba lõimede kogupikkuse ümardatuna sajandikeni.



Koostada programm, mis

- küsib kasutajalt
  - failinime, kus on vaipade lõpp-pikkused ujukomaarvudena meetrites eraldi ridadel,
  - kõrvuti olevate lõimede arvu 5-meetriste ja pikemate vaipade puhul (täisarv),
  - kõrvuti olevate lõimede arvu lühemate vaipade puhul (täisarv);
- loeb failist vaipade pikkused,
- arvutab (funktsiooni **lõimede_pikkus** abil) ja väljastab ekraanile iga vaiba lõimede kogupikkuse,
- arvutab ja väljastab ekraanile, kui palju läheb lõimeniiti vaja kõigi vaipade peale kokku ümardatuna sajandikeni.

Õppijate erinevad käitumismustrid

TARTU ÜLIKOOL

- 27 õppijast 15 avas ülesande lahendamise alguses Thonnys ühe varasema lahenduse faili, kust vajadusel abi saada. Kuus alustas ilma teisi faile avamata ja kuus avas neid kohe rohkem (üks lahendajatest avas kohe 21 faili).
- Viis lahendajat 27-st vajas ülesande lahendamise jooksul vaatamiseks rohkem kui 10 varasema ülesande lahendust.
- Kuus lahendajat 27-st uuris varasemate ülesannete lahendusi sellisel viisil, et pidas vajalikuks neid ka käivitada.
- Suuremate osade kopeerimist teise ülesande lahendusest kasutas kuus õppijat 27-st.

Õppijate erinevad käitumismustrid

TARTU ÜLIKOOL

- Neli õppijat 27-st lahendas ülesannet nii, et jättis varasema ülesande koodi Thonny programmiaknasse ja asus seda modifitseerima ning täiendama.
- Kaks lahendajat kopeeris programmiaknasse ka ülesande püstituse ja hoidis seda lahendamise ajal väljakommenteerituna silme ees.
- Viis õppijat 27-st kasutas programmiosa töö kontrollimiseks eelnevat väljakommenteerimist, viis tegi katsetusi mõnes teises failis ja kolm kustutas ülejäänud koodi ära või kleepis korraks mujale.
- Seitse õppijat 27-st kasutas programmi tööst aru saamiseks silumist.

Õppijate erinevad käitumismustrid

TARTU ÜLIKOOL

- 10 õppijal 27-st tuli ette ka seda, et nad käivitasid pärast veateadet programmi uuesti, ilma et oleks proovinud vahepeal viga parandada.
- Kolm õppijat 27-st lahendas ülesannet rohkem kui ühes failis.
- 12 õppijat 27-st kirjutas esialgu valmis funktsiooni ja asus siis seda testima.
- Koguni 11 lahendajat aga käivitas esimest korda alles siis, kui kogu lahenduse esimene versioon oli peaaegu valmis või täiesti valmis.

Lahendajatüübid

“**Müüriladujad**”. Seda saab võrrelda müüriladuja tööga, kuna töö tulemusel kerkib müür järk-järgult. Ta laob eelmise telliserea ära ja alles siis alustab järgmisega.

“**Kiviraiujad**”. Seda saab võrrelda kiviraiuja tööga, kus on kõigepealt vaja algmaterjali, millest hakata sobivat eset raiuma. See on raske töö, mis nõuab palju vaeva. Samuti on vaja palju erinevaid tööriistu.

“**Meistrid**”. Teda võib võrrelda meistriga, kes on osav ja teab, mida teeb.

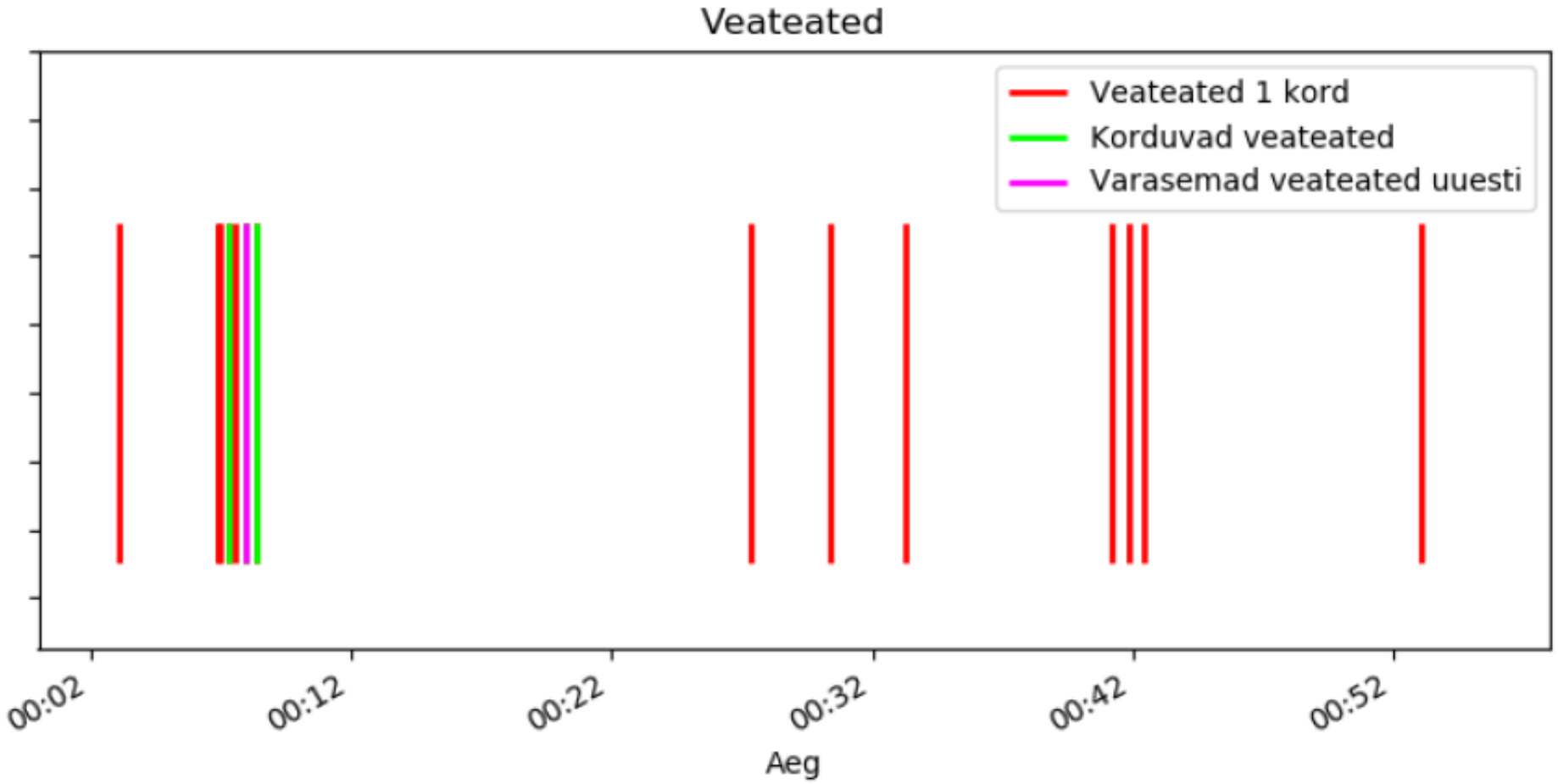
Lahendajatüübid

TARTU ÜLIKOOL

- **“Müüriladujad” töötasid järk-järgult.**
  - **Kõigepealt üks osa programmist (funktsioon),**
  - **seejärel testimine,**
  - **järgmine osa.**
- “Kiviraiujad” kirjutasid suurema osa programmist valmis ja hakkasid seda seejärel lihvima.
  - Palju vaeva ja abivahendina oli vaja palju teisi faile.
  - Teiste gruppidega võrreldes läks rohkem aega ja tuli palju korduvaid veateateid.
- “Meistrid” kirjutasid samuti suurema osa programmist valmis
  - Tuli hästi välja.
  - Veateateid oli vähe või suutsid õppijad vead kiiresti parandada.

Ühe “müüriladuja” veateadete paiknemine ajateljel

TARTU ÜLIKOOL

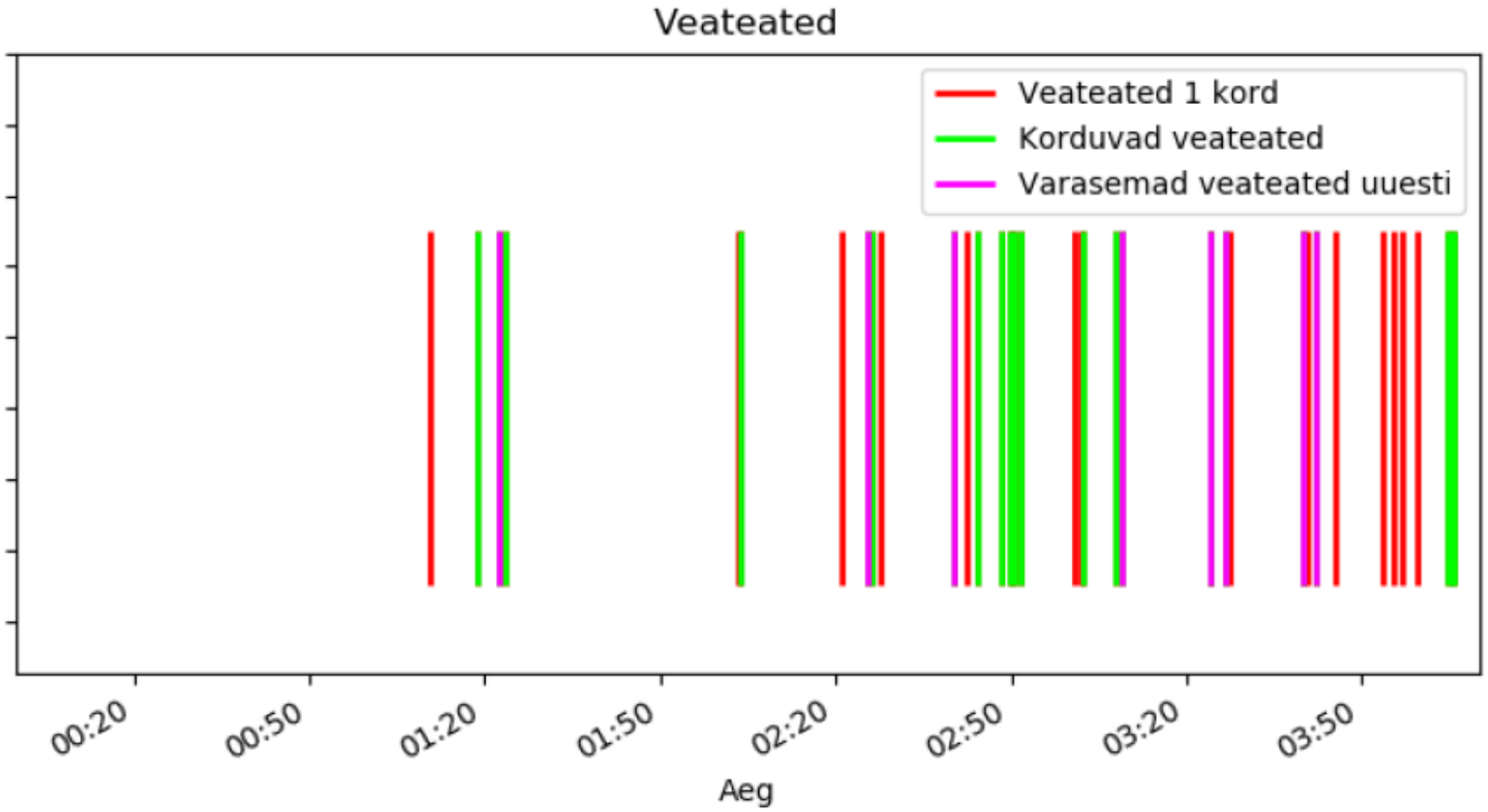


Lahendajatüübid

TARTU ÜLIKOOL

- “Müüriladujad” töötasid järk-järgult.
  - Kõigepealt üks osa programmist (funktsioon),
  - seejärel testimine,
  - järgmine osa.
- **“Kiviraiujad” kirjutasid suurema osa programmist valmis ja hakkasid seda seejärel lihvima.**
  - **Palju vaeva ja abivahendina oli vaja palju teisi faile.**
  - **Teiste gruppidega võrreldes läks rohkem aega ja tuli palju korduvaid veateateid.**
- “Meistrid” kirjutasid samuti suurema osa programmist valmis
  - Tuli hästi välja.
  - Veateateid oli vähe või suutsid õppijad vead kiiresti parandada.

Ühe “kiviraiuja” veateadete paiknemine ajateljel

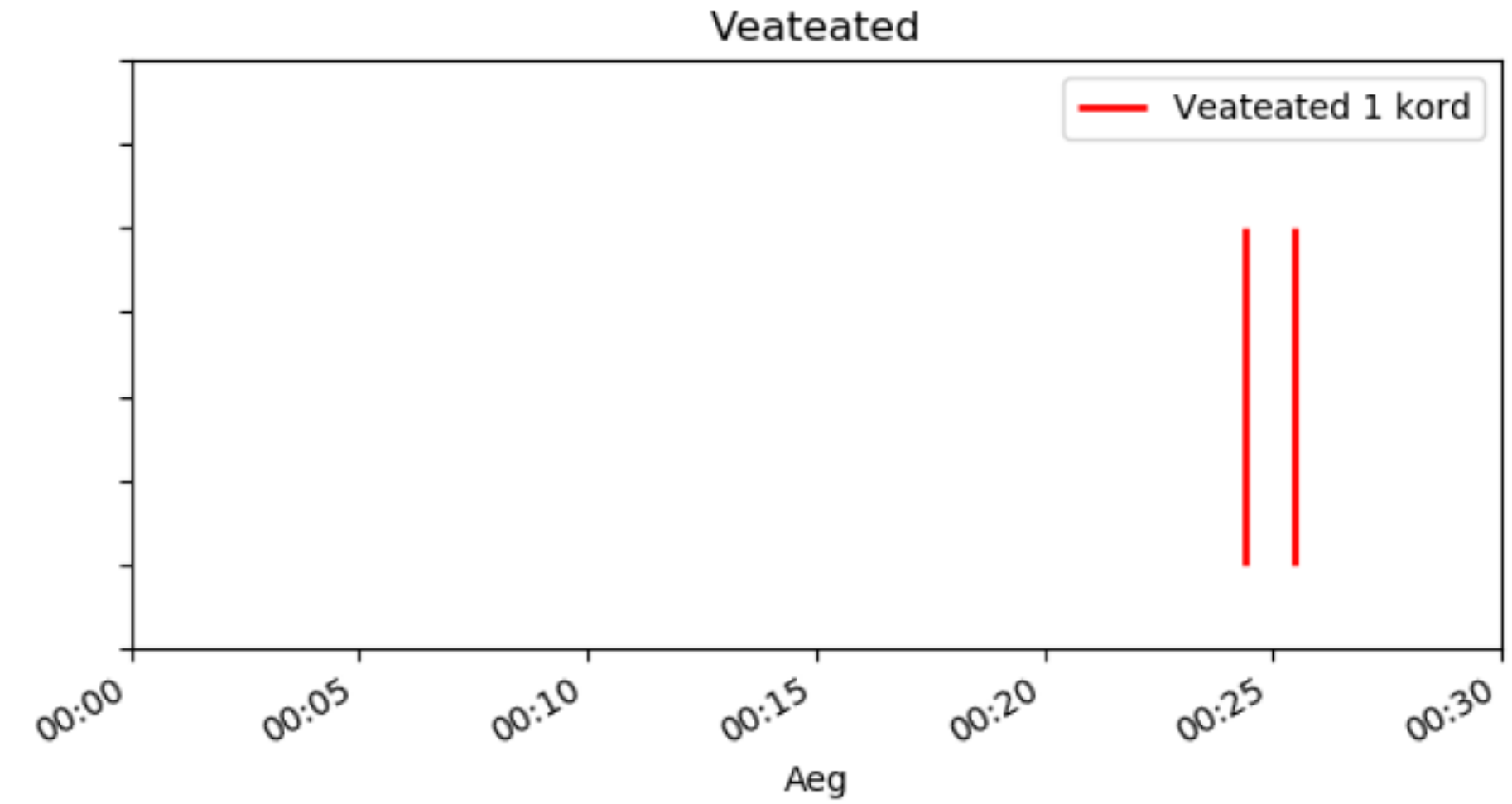


Lahendajatüübid

TARTU ÜLIKOOL 1632

- “Müüriladujad” töötasid järk-järgult.
  - Kõigepealt üks osa programmist (funktsioon),
  - seejärel testimine,
  - järgmine osa.
- “Kiviraiujad” kirjutasid suurema osa programmist valmis ja hakkasid seda seejärel lihvima.
  - Palju vaeva ja abivahendina oli vaja palju teisi faile.
  - Teiste gruppidega võrreldes läks rohkem aega ja tuli palju korduvaid veateateid.
- **“Meistrid” kirjutasid samuti suurema osa programmist valmis**
  - **Tuli hästi välja.**
  - **Veateateid oli vähe või suutsid õppijad vead kiiresti parandada.**

Ühe “meistri” veateadete paiknemine ajateljel



Lahendajatüübid ja lahendamise sujuvus

- Kõik 12 "müüriladujat" lahendasid ülesande kiiremini kui ükski seitsmest "kiviraiujast".
- Kõigil seitsmel “kiviraiujal” oli ka rohkem veateateid ja korduvaid veateateid kui ühelgi “müüriladujal”.
- Võib öelda, et seitsmest ülesannet kõige vähem sujuvalt lahendanud õppijast ei töötanud keegi järk-järgult.

Lahendajatüübid ja lahendamise sujuvus

- "Meistrite" grupi viis liiget kirjutasid enne testimist valmis suurema osa ülesande lahendusest, kuid said sellega hästi hakkama.
- Ei ole teada
  - kas "meistrid" töötavad ikkagi järk-järgult, aga MOOC-i kokkuvõttev arvestusülesanne oli nende jaoks sedavõrd lihtne ja nad käsitlesid seda ühe osana
  - või on olemas õppijate grupp, kelle puhul mitte-järkjärguline käitumismuster toimib.

Mis vajab rohkem tähelepanu?

TARTU ÜLIKOOL

- **Järkjärgulisus.** Võimalik, et järkjärgulise töötamise soovitamine võib aidata algajatel programmi tööst paremini aru saada ja vähendada töö käigus olukordi, kus koodis on korraga palju vigu ja lõpuks on raske mõista, mille tõttu veateade tuli.
- **Viisid, kuidas programmiosade tööd testida.**
- **Funktsiooni väljakutsumine.** Kuna õppijatel oli üsna palju raskusi funktsiooni väljakutsumisega, vajab ka see eraldi tähelepanu.

TARTU ÜLIKOOL

Mida räägivad logid programmeerimisülesande lahendamise kohta?

Heidi Meier

09.02.2019