

# LTAT.03.003 OOP Rühmatöö 1

## Rahapäevik

Autorid: Marvin Vihman, Karolin Kivilaan

Programmi üldine eesmärk on hõlbustada oma rahaasjade jälgimist (näiteks nii enda kui teiste tudengite miljonite kohta arve pidamine, mida OOPi rühmatöö valmimisel teenima hakatakse).

Programm küsib kasutajalt, kas tal on olemas kasutajanimi. Kui ei, siis luuakse kasutajanimi kasutaja ees- ja perenimest ning suvaliselt valitud arvust 1-100 ning samuti luuakse ka tekstifail, kuhu hakatakse salvestama sisestatud andmeid. Kui kasutajanimi oli olemas, siis on võimalik jätkata sama tekstifaili andmete sisestamist, mis loodi koos kasutajanimiga, või kaotada olemasolev sisu.

Rahapäevik loob võimaluse vaadata oma konto hetkeseisu, lisada päevikusse sissekandeid, ehk siis kas raha kulus millegi peale või tuli juurde, näha kuu kokkuvõtet, või kui kuu alles kestab, siis näha hetke kokkuvõtet.

### Projekti klassid ning nende meetodid

#### **class Main:**

- Toimub suhtlus kasutajaga, näiteks uuritakse, kas kasutajal on kasutajanimi juba olemas, vajadusel väljastatakse uus loodud kasutajanimi.

#### **class Päevik:**

Klass failiga töötamiseks, andmete lugemiseks ning kirjutamiseks

- failiSisu() - Failist sisu kättesaamiseks, et seda töödelda saaks.
- lisaPäevikusse(String summa) - Faili lisamiseks, palju kulutati või kui palju raha saadi.
- kuuKokkuvõte(String kuupäev) - Etteantud kuu kokkuvõtte kuvamiseks.
- hetkeKuuKokkuvõte() - Käesoleva kuu kokkuvõtte kuvamiseks.

#### **class Kasutaja:**

- looKasutajanimi (String eesnimi, String perenimi) - luuakse kasutajale kasutajanimi juhul, kui kasutaja vastas, et tal puudub see, või soovitakse luua uus fail.

## Projekti tegemise protsess

Kõigepealt arutasime ja leppisime kokku, mis võiksid olla programmi poolt pakutavad võimalused kasutajale, mida ja kuhu programm väljastab või salvestab, jaotasime natuke ülesandeid. Seejärel programmeerisime kõigepealt klassi Main, siis klassi Kasutaja ja klassi Päevik. Mõlemas protsessis osalesid mõlemad rühmaliikmed. Arutelu järgnevate sammude ja ülesannete osas oli pidev.

## Panus:

- Marvin programmeeris klassi Päevik ning selles olevad meetodid. Samuti lisas klassi Main JOptionPane võimalused.
  - Ajakulu, mis kulus ainult projektile, ei oska täpselt öelda, kuid kõigega, puhkuste, googeldamisega, otsimisega, kirjutamisega ning testimisega, koos läks umbes 5-6 tundi.
- Karolin programmeeris klassist Main suurema osa ning klassi Kasutaja
  - Ajakulu on raske hinnata ka orienteeruvalt, sest see koosnes otsimisest ja programmeerimisest ja samuti tehtud asjade korduvas läbi töötamises ning mõttetöös, kas midagi saaks veel teha. Neid tegevusi arvestades läks samuti umbes 6 tundi.

## Mured:

- Otseselt mingeid suuri muresid polnud. Selles mõttes olid meie töömeetodid natuke erinevad, et Marvin googeldas rohkem ja Karolin üritas kasutada suuremas osas OOPi õpikut.

## **Töö lõpptulemus:**

- Lõpptulemusega jäime ise rahule. Suhtlus kasutajaga on interaktiivne, alguses püstitatud eesmärgid erinevate meetodite osas said täidetud, failist lugemine ja sinna kirjutamine toimib hästi.
- Arendamist ootabki ehk see graafilise kasutajaliidese pool. See muudaks programmi veel põnevamaks.

## **Testimine:**

- Testimine toimus jooksvalt töö käigus. Seda oleks saanud lihtsustada ning kiirendada, kui oleks teinud eraldi test klassid, kuid katsetamine sai tehtud Main klassis selleks, et näha alguses kohe ära, kas klass ning funktsioonid toimuvad nii nagu peavad.