**Valuutakalkulaator 2.0**

Rühmatöö

LTAT.03.003 Objektorienteeritud programmeerimine

Autorid: Erki Veeväli, Markus Savotškin

Projekti teemaks valisime esimese rühmatöö käigus valminud valuutakalkulaatori edasi arendamiseja sellele asutajaliidese loomine. Eesmärgiks, nagu esimese rühmatöö puhul oli luua programm, mis automatiseeriks valuutakursside lugemise

Programmi töö aluseks on valuutakursse sisaldava faili sisse lugemine. Kuna valuutakursid muutuvad igapäevaselt, võimaldab programm kursid sisse lugeda eestipank.ee/valuutakursid lehel kättesaadavast .csv failist. Antud fail peab paiknema projekti kaustas ning allalaadides ei tohi selle nime muuta(nimi peab olema currency-rates.csv). Failist saadud andmete põhjal väärtustatakse valuuta isendid ning salvestatakse. Lisaks loetakse faili päisest eraldi veel kuupäev.

Programmi käivitamisel kuvatakse kasutajale aken, mis võimaldab valida erinevaid funktsioone. Iga funktsioon avaneb erinevas uues aknas. Välju nupp sulgeb programmi ja salvestab tehtud muudatused faili.

Funktsioonide kasutamisel kontrollitakse suure osa sisendite korrektsust ning erinditöötluse abil antakse kasutajale teada, kui andmed pole korrektsed.

**Klasside kirjeldused:**

Klass **Valuuta**: põhieesmärk luua valuuta isend, mis sisaldab selle kurssi ja tähist.

oluliseim meetod on teisenda, mis teisendab vastavalt kasutaja poolt antud summat/väärtust

Klass **FailiLugeja**: põhieesmärk on lugeda failst andmeidandmeid.

tähtsaim meetod on loeKursid, mis tagab vajalikud andmed koodi korrektseks funktsionaliseerimiseks

Klass **Main**: põhieesmärk on jooksutada koodi kasutades eelnevalt mainitud klasse ja luua kasutajaliides.

Olulisim meetod on main.

Klass **EbasobivSisendErind**: põhieesmärk on kasutada loodud erindeid vigastest sisenditest teadaandmiseks.

**Etapid:**

1) Mõelda välja kuidas eelnevale versioonile graafiline liides lisada.

2) Kirjutada graafilise lidese meetodid

3) Kohandada olemasolev programm uute meetoditega sobivaks.

4) LIsada nuppude *handler*id ja erinditöötlus

5) Luua lisafunktsionaalsust.

6) Lõpptestimine (testimine käis samuti iga etapi juurde)

Igas etapis toimus rühmaliikmete vahel pidev koostöö, et oleks arusaadav, mida mingi koodijupp tegema peab

**Rühmaliikmete panused:**

Meetodid mis pärinevad esimeses rühmatööst – vastavalt sealsele kirjeldusele.

Erki:

* Klass EbasobivSisendErind
* Klassis Main – enamus Graafika lõpuga meetoditest.

Markus:

* Klassis Main – start()
* Pildid

**Ajakulu:**

Projekti tegemiseks läinud ajakulu on keeruline öelda, soovitud funktsionaalsuse saamine nõudis palju testimist ja proovimist. Puusalt öeldes kulus projekti tegemiseks 8 tundi.

**Testimine:**

Programmi testimine toimus etapiti vastavalt osade valmimisele. Eelmisest versioonist pärit meetodid olid kindlalt juba töökorras, kui dneid tuli vahel muuta, kuijselgus et need uute lahendustega ei sobi. Probleemid uute ja vanade meetodite omavahel kokkusobimisega tulid välja just testimise käigus.

**Hinnang:**

Arvame et saime projektiga hästi hakkama, püstitatud ülesanne oli jõukohane ning programm töötab. Päris kasutajaliidesega on programm palju lihtsamini kasutatav kui esimese rühmatöö versioonis.