

Projektikertomus

OHJELMOINTI 2- HARJOITUSTYÖ

SENJA MOILANEN

23.3.2023

Sisällys

Ajankäyttö	2
Mitä opin tehdessäni ohjelmaa?	3
Arviointia	3
Kehittämisosas: Säästö tavoite - ohjelma	4
Harjoitustyön aihe	4
Olioluokka	4
Ohjelman rakenne	4
Aloitus-ikkuna	5
Luo uusi käyttäjäikkuna	5
Käyttäjän tilin tiedot-ikkuna	6
Onnitteluikkuna	7
Muita tietoja ohjelmasta	7
Ohjelman käyttö- ja asennusohje	7

Ajankäyttö

Aloitin projektin toteuttamisen sen jälkeen, kun ohjelmani idea oli hyväksytty. Keksin ohjelman idean hyvin nopeasti, eikä sen ominaisuuksien keksiminenkään vienyt yhtään enempää aikaa. Sanoisin, että ohjelman idean keksimiseen/suunnittelemiseen meni yhteensä noin tunti.

- 4.3 Ohjelman idean suunnitteleminen noin 1 h

Kun olin saanut ohjelman aiheen hyväksyttyä, aloitin työstämään ohjelman olioluokkaa. Mietin, minkälaisen haluan ohjelman kokonaisuuden olevan ja lähdin sitä kautta rakentamaan olioluokkaa. Sen toteuttaminen sujui hyvin, vaikka työstäminen alkoi hitaasti. Sanoisin, että olioluokkaa, joka lähetettiin hyväksyttäväksi opettajalle, työstin yhteensä reilut 2 h. Suurin haaste tässä vaiheessa oli, että ohjelma tuli omasta ideasta/aiheesta, joten olioluokkaa oli ensin vaikea lähteä rakentamaan. Lopulta kuitenkin olioluokan rakentaminen lähti sujumaan, kun pääsi vauhtiin.

- 12–15.3 Olioluokan suunnitteleminen/toteuttaminen yhteensä reilu 2 h

Sanoisin, että vaikein ja eniten aikaa vievä osuus oli toteuttaa toimiva ohjelma. Kuitenkin myös tämä osuus oli kaikista kiinnostavin, koska pääsi vihdoinkin toteuttamaan itse ohjelman ja näkemään, mitä kaikkea olin oppinut kurssin aikana.

Kun lähdin toteuttamaan käyttöliittymän koodia, oli ensin hankalaa valita, mistä lähtisi liikkeelle, mutta kun vauhtiin pääsi, sujui tämäkin osuus oikein hyvin. Halusin kehittää käyttöliittymää rauhassa ja tarkasti, joten varasin sen luomiseen paljon aikaa. Sanoisin, että 17–21.3 välisenä aikana työstin käyttöliittymää ja täydensin sen toteuttamaa olioluokkaa reilut muutaman tunnin päivässä.

Lisäksi käytin 22.3 aikaa käyttöliittymän ulkoasun työstämiseen. Tähän osuuteen kuului esim. fonttien muuttaminen, asettelun säätäminen, grafiikan luominen ohjelman taustaksi jne. Se vei aikaa noin reilu 3 tuntia.

- 17–21.3 Käyttöliittymä ja olioluokan täydentäminen. Noin reilu 2 h päivässä
- 22.3 Käyttöliittymän ulkoasun työstäminen noin reilu 3 h

Mielestäni harjoitustyön yksi haastavampia osuuksia oli JavaDoc:n luominen. JavaDoc kommenttien luomisessa kesti enintään tunti, mutta itse JavaDoc dokumentaation luomisessa minulla oli haasteita. Kuitenkin monen tunnin väännön jälkeen sain mielestäni senkin luotua onnistuneesti.

- 22.3 JavaDoc:n kommenttien luominen noin 1 h
- 23.3–24.3 JavaDoc dokumentaation luominen yhteensä noin 4 h

Lopuksi tein ohjelman kehittämisoppaan. Kehittämisopas käsittelee ohjelman rakennetta, miten ohjelmaa käytetään, kuinka paljon ohjelman luominen vei aikaa ja ylipäättänsä projektin kulusta ja sen aikana tulleista mietteistä. Kehittämisopas vei aikaa ja oli ehkä haastavimpia osuuksia projektissa. Kehittämisopas vei aikaa noin 4 h.

- 24.3–26.3 kehittämisoppaan luominen, joka vei aikaa yhteensä reilu 4 h

Yhteensä projektiin meni aikaa noin 23 tuntia.

Mitä opin tehdessäni ohjelmaa?

Harjoitustyö oli hyvää kertausta kurssin asioista ja sitä tehdessä tuli opittua myös paljon enemmän. Mielestäni harjoitustyön hankalin asia oli JavaDoc dokumentaation luominen. Olin kurssin aikana yrittänyt luoda sitä, mutta epäonnistunut pahasti. Kuitenkin nyt voin sanoa, että olen ”onnistunut” luomaan JavaDoc dokumentaation.

Arviointia

Mielestäni työni lopputulos on sellainen, mistä voin olla ylpeä. Minulla oli paljon toiminnallisuuksia, joita halusin ohjelman toteuttavan ja olen ylpeä, että sain suurimman osan niistä toteutettua. Löysin myös ratkaisuja moneen haasteeseen, joita minulla oli projektin alussa ohjelman kanssa.

Kuitenkin tiedän, että ohjelmassa on paljon aukkoja, joita voisi lähteä kehittämään. Esim. ohjelmaan ns. kirjautuminen voisi olla tehokkaampaa ja turvallisempaa. Muutenkin ohjelmalle voisi kehittää paljon monipuolisempia ominaisuuksia, koska tämänhetkinen versio on todella yksinkertainen ja pelkistetty versio siitä, mitä se voisi olla.

Kehittämispas: Säästö tavoite - ohjelma

Harjoitustyön aihe

Harjoitustyöni aihe oli ns. säästötavoite – ohjelma, jolla voi luoda käyttäjän ja tarkastella oman säästötavoitteen edistymistä. Ohjelma siis pitää kirjaa käyttäjän säästöjen määrästä ja siitä, kuinka kauan sen saavuttamiseen on.

Ohjelman ominaisuuksiin kuuluu mm.

- Käyttäjä antaa käyttäjänimensä. Jokaisen käyttäjän tiedot tallennetaan oliotiedostoon.
- Jos ohjelma ei löydä annettua käyttäjänimeä, luo se uuden ja kysyy seuraavat asiat:
 - Kuinka paljon haluaa säästää.
 - Millä aikavälillä (Aikavälin saa valita)
 - Haluaako käyttäjä lisätä säästöihin päivittäin, viikoittain vai kuukausittain.
- Jos ohjelma löytää käyttäjätiedot, näyttää se seuraavat asiat:
 - Ohjelma laskee arvion, kuinka paljon käyttäjän pitää säästää valintansa mukaan (päivittäin, viikoittain vai kuukausittain), jotta tavoite täyttyisi ajoissa. Aina, kun käyttäjä kirjaa, kuinka paljon hän on säästänyt, laskuri päivittyy.
 - Ikkunassa on mittari, joka täyttyy sitä mukaan, kun käyttäjä on lisännyt rahaa.
 - Ikkunasta löytyy myös säästöjen määrä.
- Kun käyttäjän säästötavoite on säästetty, ohjelma onnittelee käyttäjää ja kysyy, haluaako hän aloittaa uuden säästötavoitteen. Jos ei, ohjelma poistaa käyttäjän tiedot tiedostosta.

Olioluokka

Käytin työssäni olioluokkaa SaastoTavoite.

Luokassa on seitsemän kenttää:

- Merkkijono käyttäjänimelle.
- Merkkijono, kuinka monesti tilille halutaan lisätä (Päivittäin, viikoittain tai kuukausittain).
- Käyttäjän antama säästötavoite desimaalilukuna.
- Kuinka paljon tavoitetta on jäljellä desimaalilukuna.
- LocalDate:na päivämäärä, johon asti on aikaa säästää.
- LocalDate:na päivämäärä, jolloin säästäminen alkoi.
- LocalDate:na päivämäärä, joka päivittyy aina, kun käyttäjä kirjautuu sisään.

Luokasta löytyy alustaja ja get- ja set-metodit kaikille luokan tiedoille. Lisäksi luokassa on metodeja, joista kolme laskee kuinka monta päivää, viikkoa tai kuukautta käyttäjällä on jäljellä säästää. On myös metodi, joka laskee, kuinka paljon käyttäjän pitää säästää päivässä, viikossa tai kuukaudessa, jotta tavoite täyttyy. Ja on vielä metodi, joka laskee kuinka paljon tavoitetta, on jäljellä, kun käyttäjä lisää rahaa.

Ohjelman rakenne

Ohjelma koostuu kahdesta luokasta: Saastotavoite-olioluokasta ja käyttöliittymän määrittelevästä luokasta SaastoTavoiteOhjelma, jossa löytyy myös ohjelman pääohjelma.

Saastotavoite-luokka käsittelee käyttäjän tilin perustoimintoja. Luokan perustoiminnoista on kerrottu edellä ja niiden yksityiskohtia voi myös tarkastella JavaDoc-dokumentaatiosta.

Luokka SaastoTavoiteOhjelma koostuu muutamasta eri asettelusta.

Aloitus-ikkuna

Kun ohjelma avautuu, avautuu yksinkertainen asettelu, jossa ohjelma pyytää käyttäjää antamaan jo olemassa olevan käyttäjänimen tai uuden. Ikkunasta löytyy myös ohjeita käyttäjälle, miten ohjelma toimii

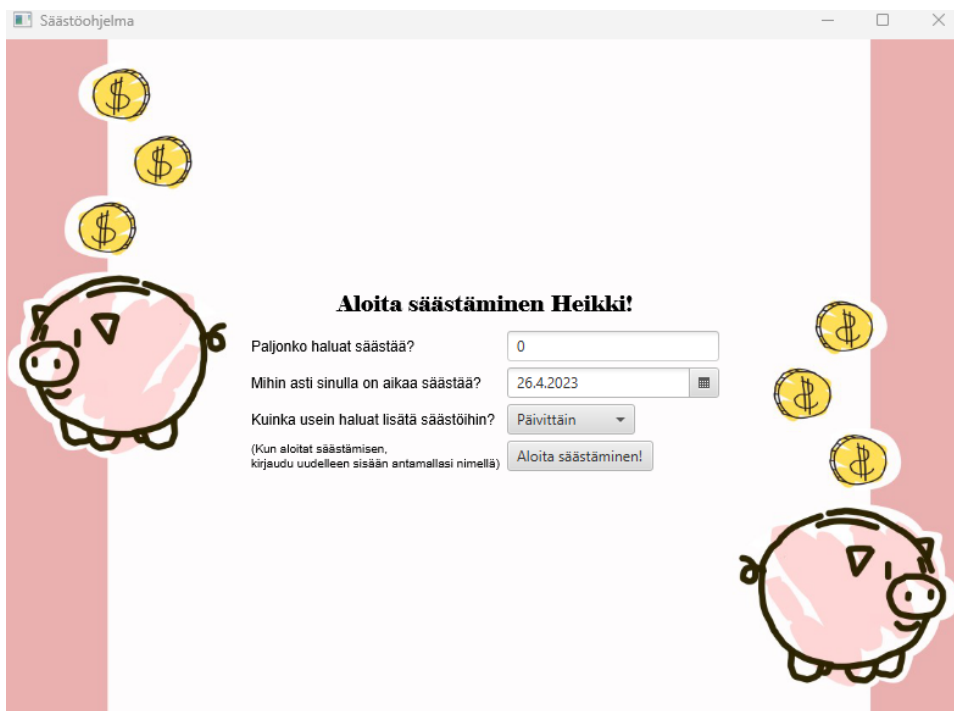
Aloitusikkuna, kun ohjelma avataan.



Luo uusi käyttäjäikkuna

Jos ohjelma ei löydä käyttäjänimeä tiedostosta, muuttuu ikkunan asettelu uuteen. Asettelu pyytää käyttäjää täyttämään tarvittavat tiedot uutta säästötavoitetta varten. Asettelusta löytyy myös ohjeet, että tilin luomisen jälkeen käyttäjän pitää kirjautua sisään aloitusikkunan kautta. Aloita säästäminen!-painike siis luo uuden käyttäjän ja vie käyttäjän takaisin aloitus-ikkunaan.

Luo uusi käyttäjäikkuna



Käyttäjän tilin tiedot-ikkuna

Jos ohjelma löytää käyttäjänimen tiedostosta, avautuu tilin tiedot-ikkuna. Nimensä mukaan ikkunasta löytyy käyttäjän säästötavoite tilin tietoja. Lisäksi käyttäjä voi tilin tarkastelun lisäksi lisätä rahaa säästöihinsä.

Kuten muissa ikkunoissa myös tässä ikkunasta löytyy ohje, miten ikkunan toiminnot toimivat.

Käyttäjän tilin tiedot-ikkuna



Lisäksi ikkunasta löytyy graafinen mittari, joka täyttyy sitä mukaan, mitä käyttäjä lisää rahaa tilille. Kun säästöt ovat puolessa välissä, mittarin yläpuolelle ilmestyy teksti "Puolivälissä!".

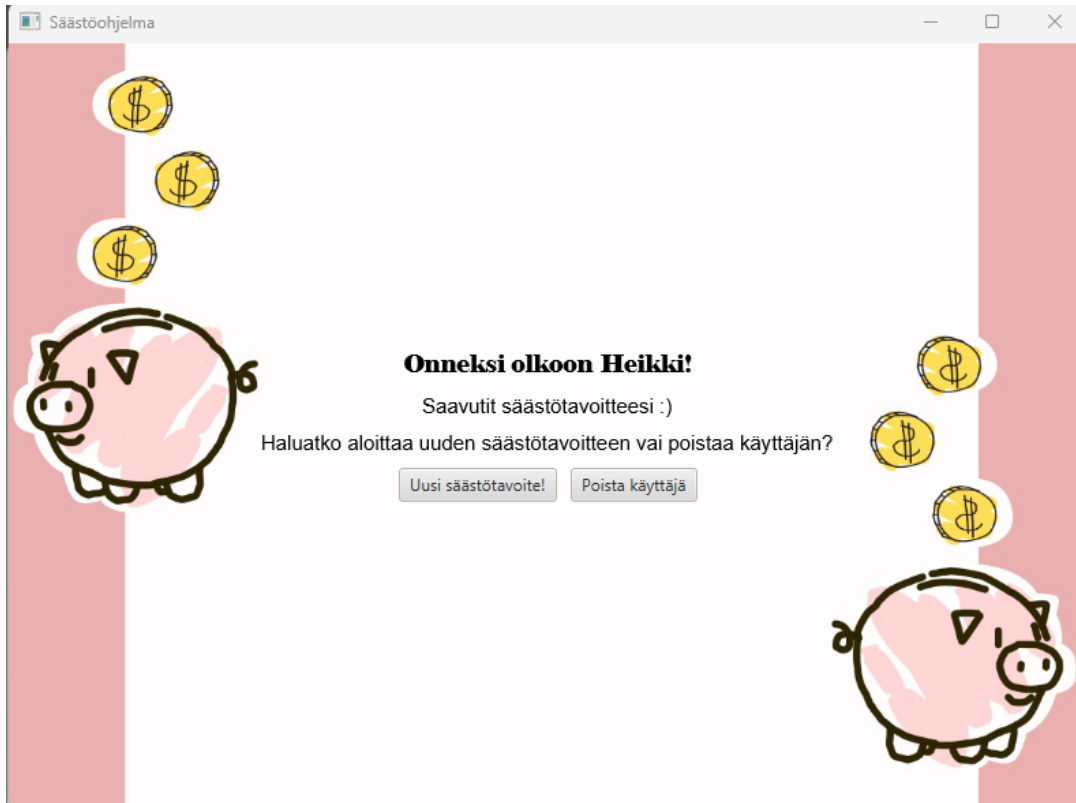
Mittari



Onnitteluikkuna

Kun käyttäjä on päässyt tavoitteeseensa, avautuu onnitteluikkuna. Ohjelma onnittelee käyttäjää ja kysyy, haluaako hän poistaa käyttäjän vai aloittaa uuden säästötavoitteen. Jos käyttäjä poistaa tilin, asettelu muuttuu aloitusikkunaan, ja jos hän haluaa aloittaa uuden tilin, avautuu luo uusi käyttäjä-ikkuna.

Onnittelu-ikkuna



Muita tietoja ohjelmasta

Kun ohjelma käynnistyy, niin se lukee käyttäjä tiedot saastoTavoite.dat tiedostosta. Jos tiedostoa ei löydy, luo se uuden tiedoston samalla nimellä. Tiedot tallennetaan tiedostoon aina, kun luodaan uusi käyttäjä, "kirjaututaan ulos" käyttäjältä tai kun poistetaan käyttäjä. Tiedosto ei siis päivitty automaattisesti, kun ikkuna suljetaan.

Ohjelman käyttö- ja asennusohje

Pura palautuksen "OH2_harjoitustyo" zip-tiedosto sopivaan paikkaan. Puretusta kansioista löytyy kansio "Harjoitustyo_ohjelma", joka on ohjelman projekti eli sisältää ohjelman koodin. Avaa projekti IntelliJ:ssä.

Käännä ja aja ohjelma. Ohjelmassa voit luoda uuden käyttäjän, tarkastella vanhan käyttäjän säästötavoite-tietoja tai lisätä rahaa tilille. Ohjelmasta löytyy ohje kenttiä, jotka kertovat tarkemmin, miten ohjelmaa käytetään. Kuitenkin yksinkertaisesti sanottuna:

1. Kun avaat ohjelma, ensimmäinen ikkuna on ns. kirjaudu sisään-ikkuna:
 - Voit joko kirjautua sisään olemassa olevalla käyttäjänimellä
 - Tai jos sinulla ei ole käyttäjää ohjelmaan voit antaa uuden käyttäjänimen samaan kenttään.
2. Jos luot uuden tilin, ohjelma ohjaa sinut Luo uusi käyttäjäikkunaan. Ikkunassa voit asettaa säästötavoitteesi, mihin asti sinulla on aikaa säästää ja kuinka monesti aiot lisätä tilille (Päivittäin,

viikoittain vai kuukausittain). Kun painat, aloita säästäminen! -nappia ohjelmaa ohjaa sinut takaisin kirjaudu sisään – ikkunaan, jossa voit nyt kirjautua sisään uudelle käyttäjällesi.

3. Jos sinulla on käyttäjä, voit kirjautua sisään. Tällöin ohjelma ohjaa sinut ikkunaan, jossa voit tarkastella tavoitteesi edistymistä. Tässä ikkunassa löytyy mm:
 - Kuinka paljon tavoitetta on jäljellä.
 - Mittari, joka esittää edellä mainitun tavoitetta jäljellä-ominaisuuden käyttäjälle visuaalisesti.
 - Kuinka monta päivää, viikkoa tai kuukautta tavoitteen saavuttamiseen on jäljellä.
 - Kuinka paljon keskimäärin pitää säästää päivässä, viikossa tai kuukaudessa.
 - Lisätä säästöihin rahaa.
4. Jos käyttäjän säästötavoite täyttyy, onnittelee ohjelma käyttäjää ja kysyy, haluaako hän aloittaa uuden säästötavoitteen vai poistaa käyttäjän.