

# Olio-ohjelmointi CT60A2411, kevät 2020

## Harjoitustyö, ActivityLogger

Tekijät: Ilmari Vahteristo 0545212, Erik Kuitunen 0537275

### Ohjelman kuvaus

ActivityLogger-ohjelma on ikään kuin päiväkirja puhelimesi. Sillä pystyy arvostelemaan päivän, merkkamaan nukutut tunnit ja sosiaalisen ajan ja sillä voi tallentaa päivittäisiä aktiviteetteja. Tallennettaviin aktiviteetteihin kuuluu mm. liikunta, opiskelu ja ystävien kanssa vietetty aika. Jokaiselle aktiviteetille on lisäksi omia, aktiviteettikohtaisia kysymyksiä.

Tallennettuja päiviä pystyy muokkaamaan jälkikäteen, ja niitä voi tarkastella omassa ikkunassaan. Lisäksi on mahdollista katsoa yhteenveto kaikista ohjelmalla tallennetuista päivistä.

### Kuka teki, mitä teki

Käyttöliittymän ja toiminnallisuuksien suunnittelu: Ilmari

Käyttöliittymän rakentaminen: Ilmari ja Erik

Tietojen tallennus ja lukeminen: Erik

Jo tallennettujen tietojen vieminen vastauskenttiin: Ilmari ja Erik

Tietojen analysointi: Erik

Fragmenttien toteuttaminen: Erik ja Ilmari

### Miten ohjelma suunniteltiin ja toteutettiin

Ilmarilla oli valmis idea käyttöliittymineen ja valintapuineen. Lähdettiin työstämään, kumpikin omaa tahtia ja toistemme koodia välillä täydennellen. Käytännössä koko toteutuksen suunnittelu oli kuitenkin ns. vituillaan, sillä teimme paljon päällekkäistä työtä, ja Gitin käytön opettelu tuotti ajoittain tuskaa sekä kadonnutta koodia. Vasta viimeisen parin viikon aikana onnistuimme jakamaan tehtäviä hieman paremmin, ja tämä näkyi myös työnteon tehokkuudessa. Koko työn toteutusta tekemisen laadun kannalta voisi kuvailla sanomalla ”kunhan tehtiin”.

Ohjelma toteutettiin muutamalla omalla näkymällään, joissa käytettiin hyödyksi erilaisia androidista löytyviä käyttöliittymäelementtejä. ActivityScreen-näkymässä on lisäksi käytetty hyödyksi fragmentteja, joiden avulla aktiviteettikohtaiset kysymykset on saatu vaihtumaan sulavasti sitä mukaa, kun aktiviteetti valitaan.

Tiedon tallentaminen, lukeminen ja analysointi on toteutettu omalla DataProcessor-luokallaan. Luokka toimii singleton-periaatteella, ja on MainActivity:n luomisen jälkeen käytössä kaikissa MainActivitystä periytyvissä luokissa. DataProcessor sisältää muutaman metodin, joiden avulla dataa käsitellään. Näitä metodeita olisi voinut tehdä paljon yksinkertaisemmiksi modulaarisuuden kasvattamiseksi, mutta tässä työssä se ei ollut tarpeen.

Tiedon tallennus tapahtuu seuraavasti: day-olioon tallennetaan kaikki yhden päivän sisältämä tieto, jonka jälkeen se muutetaan internetistä löytyneen GSON-kirjaston avulla yhdeksi riviksi JSON-esitysmuotoista dataa. Jokaiselle riville annetaan avaimeksi rivin päivämäärä, jonka jälkeen rivi tallennetaan key-value -parina Shared Preferenceen XML-tiedostoon. JSON-muotoinen data on helppo lukea XML-tiedostosta GSON:n tarjoamien metodien avulla ja ottaa uudelleen käyttöön.

### Luokkakaavio ja linkki esittelyvideoon

Linkki luokkakaavioon:

<https://i.gyazo.com/0b99112be21ffb11029cd38c8133277c.png>

Linkki videoon:

<https://www.youtube.com/watch?v=MHYUgxBS5pAM&t=2s>

### Toteutuneet ominaisuudet

Aiheesta riippumattomat ominaisuudet löytyvät: olio-ohjelmoitu, vähintään viisi erilaista luokkaa & oliota, tietojen kirjoitus tiedostoon sekä aihepesifit perustoiminnot.

Työmme perustoiminnot olivat seuraavia:

- Tietojen kirjaaminen päivälle
- Pystyy tarkastelemaan mille tahansa päivälle kirjattuja tietoja
- Päivälle kirjattujen tietojen muokkaaminen
- Viisi erilaista aktiviteettia (Opiskelu, urheilu, kaverit, ryyppääminen ja parisuhde)
- Tallennetun aktiviteetin muokkaaminen
- Päivän tietojen ja aktiviteettien tallennus tiedostoon (XML-tiedosto, jokainen päivä yhdellä rivillä JSON-muotoisena merkkijonona)

Muut ominaisuudet:

- Antaa varoituksen, jos ollaan poistumassa tallentamatta muokkauksia
- Voi lisätä tekstin (päivän tai) aktiviteetin tietoihin
- Osa analysoinnista:
  - o Kaikkien kirjattujen päivien ratingin, nukkumisen ja sosiaalisen ajan keskiarvot

- Binääristen suureiden lukumäärä kaikista kirjatuista päivistä (esim. 14 pv kirjattuna, joista viitenä tavannut uusia ihmisiä)

Edellä mainittujen perusteella arvioisimme pistemäärän olevan 25+ p.

### Käytetyt tunnit

Kumpikaan meistä ei ole pitänyt minkäänlaista kirjaa työtunneista tai edes työpäivistä. Hommia on paiskittu silloin tällöin ja viimeisellä viikolla hieman ahkerammin.

Arviomme ovat:

Erik: noin 100 h

Ilmari: noin 90 h

### Mitä opin harjoitustyössä

Erik: Yksinkertaisten android-sovellusten tekeminen ei ole kovin vaikeaa. Omaan käyttöön tulevat quality of life-softat olisi melko nopea tehdä ihan tämän kurssin pohjalta. Ei kuitenkaan tarvita hirveästi pintaraapaisua syvempää naarmua, kunnes hommat alkavat mennä jo hieman monimutkaisemmiksi. Harjoitustyö opetti tehokkaasti internetin hyödyntämistä ongelmien ratkonnassa, ja yleinen ongelmanratkaisukyky pääsi myös koetukselle. Vaikkei softasta ihan niin hienoa ehtinyt tulla, kuin alussa kuvittelin, niin kyllähän tuosta ylpeä on oltava.

#### Ilmari:

Opin luonnollisesti paljon Javasta. Huomasin, että Java vaatii huomattavasti enemmän teorian osaamista, kuin C / Python. Täytyy tietää mitä on staticit, finalit, abstractit, mikä periytyy, mistä voi kutsua mitäkin metodia yms.

Vaikeinta oli käyttöliittymien kenttien luku ja kirjoitus. Myös onCreaten yms. sielun elämää oli joskus vaikeaa ymmärtää, sillä toimivathan ne eri tavalla kuin perusfunktiot. GitHub oli myös suuri murheenkryyni. Harvoin

Kannattaa käydä C ohjelmoinnin kurssi ennen Javaa, koska sitten kun on koodannut Javalla, niin ei tahdo takaisin C –kieleen.