

Henri Haverinen

henri.s.haverinen@student.jyu.fi

Jyväskylän Yliopisto, Tietotekniikan laitos

Keskiviikko 6.7.2016

TIEA306 itsearviointi

[Tiivistelmä](#)

[Tehtävä](#)

[Käytännön toteutus](#)

[Oman työn arviointi](#)

[Työssä käytetyt lähteet](#)

[Ohjeita ylläpitäjille ja kehittäjille](#)

[Yhteenveto](#)

Tiivistelmä

TIEA306 ohjelmointityönä toteutin graafisen käyttöliittymän sql-kyselyiden tekemiseksi mysql-tietokantaan. Ohjelma on suurin kokonaan itse tekemäni projekti, ja aikaa kuluikin paljon uuden opetteluun sekä hyvien käytänteiden läpikäymiseen, pohtimiseen ja oppimiseen. Lopputulos on mielestäni hyvä, ja projektin aikana opin paljon uusia asioita.

Tehtävä

Tehtävänäni oli toteuttaa graafinen käyttöliittymä, jolla voidaan ottaa yhteys tietokantaan ja ajaa yksinkertaisia sql-kyselyitä sinne (SELECT, INSERT, UPDATE, DELETE). Pää tarkoituksenani oli oppia projektin rakenteen hallintaa, suunnittelua, mvc-mallia, järkevää toteutusta sekä "asioiden oikein tekemistä". Lisäksi halusin tutustua JavaFX ohjelmointikehykseen. Työtä lähdin tekemään nolla kokemuksella JavaFX:ään sekä "suurempiin" projekteihin. Eihän tämä projekti mikään iso ole, mutta silti suurin kokonaan itse toteuttamani.

Käytännön toteutus

Katso ohjelmasta Help > About > Instructions for developers and maintainers > Developing > Architecture

Oman työn arviointi

Alkuperäinen tavoite toteutui pääasiassa hyvin. Suurin puute tavoitteista on useamman kuin yhden sql-tietokantatyypin tukeminen. JDBC ajureiden kanssa tuli haasteita, sillä eivät toimi tietokantavalmistajien kanssa aina samalla lailla. Lisäksi mielessäni oli tehdä dynaaminen ajureiden lataaminen, mutta tämäkin osoittautui aivan liian haastavaksi ja aikaa vieväksi.

Haastavaa oli JDBC ajureiden kanssa pelaaminen. Lisäksi graafisen käyttöliittymän ja ohjelmalogiikan kanssa oli välillä haasteita. Käytin lisäksi todella paljon aikaa projektirakenteen sekä "best practicien" miettimiseen, sekä suunnitteluun. Toissä olen usein törmännyt haasteisiin juurikin loogisen projekti- ja luokkarakenteen tekemisessä, sekä parhaiten käytänteiden miettimisessä. Itse koodaaminen onnistuu, mutta itsellä on suuri halu tehdä asioita oikein ja järkevästi, eikä vain "saada ne toimimaan". Tämän vuoksi käytinkin paljon aikaa niiden käsittelyyn ja oppimiseen, ja mielestäni olenkin oppinut paljon.

Edellämäinitsemani projektirakenne ja käytänne ongelmat olivat siinä mielessä odottamattomia, että niiden läpikäymiseen kului enemmän aikaa kuin olin ajatellut. Ennakointini graafisen käyttöliittymän sekä eri tietokantavalmistajien tukemiseen liittyvissä ongelmissa osui nappiin, ja esimerkiksi eri tietokantavalmistajien tukeminen jäikin käytännössä kokonaan pois ohjelmasta.

Ohjelma muutoin on mielestäni aika toimintavarma, enkä ole sitä saanut kaatumaan. Virheenikäsitteystä olen pyrkinyt tekemään varmaa sekä selkeää, joskin parannettavaahan varmasti löytyy aina. Lisäksi olen oppinut, että omalle koodille alkaa tulla "sokeaksi", eikä omia virheitään välttämättä huomaa.

Projekti on ollut todella opettavainen työ, ja olen oppinut paljon uusia asioita, käytänteitä, tekniikoita, sekä ihan perus ohjelmointia. Olen todennut kuitenkin, että nautin paljon enemmän backend-ohjelmoinnista, eikä graafisten käyttöliittymien tunkkaaminen ole mieluisinta hommaa. Komentoriviohjelmat ovat myös paljon mielekkäämpiä tehdä kuin graafiset. JavaFX oli kuitenkin todella hyvä valinta projektiin, sillä fxml erottaa käyttöliittymän hyvin ohjelmakoodista (vrt. Java Swing). Lisäksi Mavenin käyttö projektin hallintaan, riippuvuuksien hallintaan sekä pakkaamiseen oli oiva valinta. Lisäksi käytin IntelliJIDEA kehitysympäristöä, joka on vain todella hyvä. Työn lopputulos ei ainakaan työkalujen huonoudesta siis jäänyt kiinni!

Aikataulusta myöhästyin pari viikkoa (juhannukseen mennessä olisi pitänyt olla valmista), joka on itseasiassa yllättävän vähän mitä jossain vaiheessa kuvittelin. Alkukevällä kandin tekeminen ja kesän alussa olleet muiden kurssien loppukiristykset sekä työkiireet vaikuttivat aikatauluun. Työkiireet olivat hieman odottamattomia, joka vaikutti ehkä eniten työn aikataulun myöhästymiseen. Lisäksi kuten aiemmin mainitsin, projektin rakenteen hallintaan ja miettimiseen meni odotettua pidempi aika, joka viivästytti työtä myös jonkun verran.

Itse koodin kirjoittaminen vastasi mitä luennoillakin on opetettu ja koodattu, sellaistaahan se koodailu on. Kuitenkin useasti mainitsemani projektin rakenne, parhaat käytänteet sekä työkalujen ja teknologioiden käyttö oli pitkälti oman mielenkiinnon, harrastuneisuuden ja itseoppimisen varassa. Mielestäni näitä asioita saisi opettaa paljon enemmän yliopistolla, mutta ymmärrän myös että asia ei ole näin yksinkertainen. Esimerkiksi erilaisia työkaluja ja teknologioita on aivan järjetön määrä, ja on mahdotonta poimia sieltä sellaiset joita olisi hyvä opettaa opiskelijoille. Lisäksi vasta nyt kandidivaiheen lopussa on alkanut ymmärtää kunnolla näiden asioiden päälle, ja toivonkin että maisterivaiheessa on kursseja jotka painottuvat enemmän esimerkiksi arkkitehtuurien ja ohjelmistojen suunnittelun opettamiseen.

Työssä käytetyt lähteet

Lähteinä työssä käytin pääasiassa seuraavia:

- Java-ohjelmointi, 7., uudistettu painos 2008. Mika Vesterholm - Jorma Kyppö
- www.stackoverflow.com
- docs.oracle.com
- dev.mysql.com
- <https://github.com/TomasMikula/RichTextFX/>
- <http://squirrel-sql.sourceforge.net/> (kuvista mallia UI:hin)
- <https://sites.google.com/site/gson/gson-user-guide>
- <https://maven.apache.org/>

Ohjeita ylläpitäjille ja kehittäjille

Katso ohjelmasta Help > About > Instructions for developers and maintainers

Yhteenveto

Kokonaisuudessaan ohjelmointityö oli todella opettavainen kokemus. Kurssi antaa mahdollisuuden opiskelijalle oppia niitä asioita joista hän on kiinnostunut. Omalla kohdalla oppiminen painottui "asioiden oikein tekemiseen" sekä uusien teknologioiden käyttämiseen, ja tuli ohella toki opittua paljon muutakin. Lisäksi samalla tulee tehtyä toivon mukaan hyödyllinen ohjelma, esimerkiksi opetukseen.