



RIKASTU HELPOSTI

TKO_2023 Harjoitustyö

12.4.2016

Ryhmä 29

Sanni Savonen	513561	Historia	smmsav@utu.fi
Peter Soinu	65121	TKT	petesoi@utu.fi

1. Tehtävän kuvaus ja analysointi

Alkuperäisenä tehtävänantona oli ratkaista pienehkö tietojenkäsittelyongelma, jonka sai kuitenkin itse valita. Ryhmämme tarkoituksena oli luoda androidille sovellus, joka hakee netistä reaaliaikaista tietoa pörssikursseista ja osakkeiden hinnoista. Android-sovellusten tekeminen ei ollut meille entuudestaan tuttua, mutta otimme tämän tilaisuutena opetella uutta. Ongelmien ilmettyä päätimme kuitenkin siirtyä toteuttamaan työmme loppuun puhtaasti Javaa käyttäen. Vaikka työmme pääasiallinen tarkoitus on hakea tietoa käyttäjän valitsemasta yhtiöstä, ajattelimme lisätä ”aloitusnäyttöön” erilaisia toimintoja mahdollisuuksien mukaan.

2. Ratkaisuperiaate

Käytämme työssämme graafista käyttöliittymää. Kun ohjelma ajetaan, ruudulle aukeaa uusi ikkuna. Ikkunassa on pudotusvalikko, josta käyttäjä voi valita mieleisensä yhtiön. Kun yhtiö valitaan, ruudulla näkyy sen tiedot: Last bid (viimeinen tarjottu hinta), Last ask (viimeinen pyyntihinta), Daily Change (päivämuutos) ja Daily volume (vaihdetut osakkeet päivän aikana tähän mennessä). Rahayksikkönä on Yhdysvaltain dollari (USD).

Lisäksi ruudulla näkyy koko ajan kolmen tehtävän käynnistyksessä annetun osakkeen hinta sekä muutos prosentteina. Jos muutos on positiivinen, tiedot esitetään vihreällä värillä. Jos se puolestaan on negatiivinen, tiedot esitetään punaisella värillä. Ruudulla näkyvää Tips-nappia painamalla käyttäjä pääsee tutkimaan osakesuosituksia. Sivulta aukeaa taulukko, jossa on eräitä kriittisiä osakkeen tunnuslukuja. Taulukko ladataan ohjelman käynnistyessä eli luvut ovat reaaliaikainen näkymä ohjelman käynnistyshetkellä. Tässä versiossa osakkeet on lajiteltu P/E luvun mukaiseen järjestykseen. Pe (price per earnings) eli voittokerroin ilmaisee osakeyhtiön hinnan ja tuloksen suhdetta. PE-tunnusluvulla tarkoitetaan, että kuinka monta vuotta kestää, jotta yritys tuottaa voittoa siihen sijoitetun rahan verran. Luku saadaan siis jakamalla hinta osakekohtaisella tuloksella.

3. Ohjelman ja sen osien kuvaaminen

Ohjelma koostuu kahdesta luokasta stockUI ja stockMain. StockMain sisältää vain kutsun UI:n luomiseen valituilla oletusosakkeilla. Graafisessa toteutuksessa käytössä javan swing-luokka ja sen eri komponentit.

StockUI:n konstruktorissa luodaan tarvittavat komponentit luokkametodeilla:

```
JFrame uiFrame = createStockFrame(); // Ohjelman pääkehikko uiFrame  
  
final JPanel comboPanel = new JPanel(); //Ensimmäinen sivu comboPanel (oletus)  
  
final JPanel tipPanel = new JPanel(); //Toinen sivu tipPanel  
  
addStockInfo(uiFrame, comboPanel); //Lisätään osakeinfo UI ekalle sivulle
```

```
createTextPane(uiFrame, comboPanel, stock1, stock2, stock3); //Lisätään staattinen osakenäkymä ekalla sivulle.
```

```
createTips(uiFrame, tipPanel); //Lisätään vihjetaulukko tokalle sivulle
```

```
createTipButton(uiFrame, comboPanel, tipPanel); // Lisätään vihje-nappi sekä ekalle & tokalle sivulle.
```

3.1 Osakeinfo UI

Muodostetaan HTML tekstiruutu, johon ladataan tarvittavat tiedot osakkeista. Klikkaamalla pudotusvalikosta osake, sen tiedot päivittyvät tekstiruutuun.

3.2 Vinkit

Vinkit taulukko luodaan HTML pohjaisesti. Taulukko järjestetään P/E luvun mukaiseen suositusjärjestykseen kuplalajittelun avulla.

3.3 Tekstiruutu

Tekstiruutu luodaan `createTextPane(JFrame uiFrame, JPanel panel)`-metodilla, jota kutsutaan konstruktorista käsin. Ensin metodi luo kolme muuttujaa, joihin tallennetaan tiedot kolmest annetusta osakkeesta. Sitten luodaan muuttujia, joihin tallennetaan tiedot kunkin yhtiön osakkeen hinnasta ja hinnan muutoksesta. Ruudussa näytettävä teksti tallennetaan pala palalta `StringBuffer`-olioon, joka sitten asetetaan luotuun `JTextPaneen` tekstiksi. Tekstin väri määräytyy sen mukaan, onko muutos positiivinen vai negatiivinen. Tämä testataan ennen kuin yhtiön tiedot tallennetaan `StringBuffer`-olion jatkeeksi. Lopuksi varsinainen tekstiruutu luodaan ja se asetetaan paikalleen.

4. Testausjärjestely

Kun vielä teimme työtämme Androidille, testasimme sitä lähinnä asentamalla sovelluksemme uusimman version käytössä olevaan Android-puhelimeen. Näin saimme aika hyvän käsityksen sen toimivuudesta. Kun siirryimme Javan käyttöön, sovellusta testattiin ajamalla se aina kun muutoksia tehtiin. Lopuksi vielä testasimme erikseen, että jokainen nappi varmasti toimii niin kuin pitää ja että tiedot näkyvät oikein.