Rajapinnat lopputyö

**Tehtävän annot**

Ohjelma, joka Scrooge tarvitsee työkalun analysoidakseen bitcoinin markkina-arvoa haluamallaan ajanjaksolla. Ohjelma voidaan toteuttaa, millä tahansa tekniikalla.

Lisätietoja Scrooge antoi, että ohjelman pitäisi kerätä seuraavat tiedot eri ajanjaksoilta:

* alkamis- että lopetuspäivät tulee sisältyä tarkkaan ajanjaksoon.
* Päivähinta haetaan aina, jolloin hinnan määritys aika on lähimpänä 00:00 UTC aikaan
* Käyttäjän tulee voida syöttää alkamis- ja lopetuspäivämäärät (esimerkiksi syöttökentillä)

Päätin valita Rajapinnat lopputyöksi T2 esimerkin rajapinnan ja toteutin sen annettujen ohjeiden mukaisesti. Tein tehtävät A-D.

**Layoutin suunnittelu**

Lähdin aluksi miettimään, miltä haluan ohjelmani näyttävän ja miten sijoitan kaikki elementit järkevästi. Päätin lisätä 2 textBoxia, johon syötetään alkupäivämäärä ja loppupäivä määrä. Seuraavaksi tuli ongelma, kun en halunnut suorittaa kaikkia tehtäviä peräkanaan. Loin tätä varten Comboboxin, johon lisäsin jokaiselle tehtävälle omat osiot. Lisäksi lisäsi napin, jolla suoritan ohjelman ja datagridviewin, johon sijoitan datan tehtävistä.

**Tehtävän pohjan luominen**

Lähdin miettimään, mitä tarvitsisin. Tarvitsen MVC mallia varten kansiot Model, View ja Controller. Jokaiseen luokkaan hain tarvittavat tiedot käyttäen konstruktoria.

Model: lisäsin luokan Bitcoin Data, tänne voin tallentaa kaiken tulevan datan coingecko apista.

Controller: Loin sinne seuraavat luokat ja miksi.

* AllButtonsController, jota voin kutsua Formissa, jotta ei tarvitse jokaista luokkaa kutsua erikseen, tämä mielestäni selkensi ohjelmaa paljon.
* ApiController, joka hoitaa apin käsittelyyn kaiken tarvittavan homman, kuten hakee tiedot, järjestää keskiyöksi datan ja kaikkea muuta.
* DateTimeParse, joka hoitaa unixtimen muuttamisen normaaliin aika muotoon. Lisäksi tämä luokka käsittelee kaikki päiviin liittyvän homman.
* FormController, joka hoitaa kaikki formiin liittyvät jutut, esimerkiksi nappien toiminta, syötteiden tarkistustuksen.
* Tämän lisäksi loin controller kansioon kansion task A-D, johon laitoin kaikki tehtävät omaan luokkaansa

Kansiossa view on vaik formiin tarvittavat kansiot.

**Tehtävä A – Toteutus**

Tehtävän anto: Minä päivänä tietyllä päivämäärävälillä oli halvin ja korkein hinta?

Lähdin miettimään, miten minun kannattaisi lähteä toteuttamaan tehtävää. Tajusin, jos haluan halvimman ja korkeimman hinnan niin järjestän hinnat listan hinnan mukaan ja tämän jälkeen otan sen ensimmäisen itemin. Halvimmassa järjestin käyttäen order by ja korkeimmassa OrderByDescending.

Tämän jälkeen piti muuttaa hinnan value doubleksi, koska tätä ennen se oli objekti. Lisäksi piti muuttaa timestampin päivämääräksi käyttäen DateTimeParsessa olevaa methodia. Lopuksi lisäsin kummastakin saadut tiedot datagrid elementtiin.

**Tehtävä B – Toteutus**

Tehtävän anto: Minä päivänä tietyllä päivämäärävälillä kaupankäyntivolyymi oli pienin ja suurin?

Tämä tehtävä toimii käytännössä samalla tavalla kuin Tehtävä A:n toiminta. Ainut eroavaisuus on siinä, että tässä käytetään volume listaa ja A tehtävässä prices listaa.

**Tehtävä C – Toteutus**

Tehtävän anto: Kuinka monta päivää on pisin lasku- (laskeva) ja nousutrendi (nouseva) tietyllä aikavälillä?

Lähdin luomalla methodit Pisinlasku ja Pisinnousu. Näin pystyn erottelemaan kummankin ominaisuudet toisistaan. Käytännössä kummassakin on sama toiminta. Käyn kummassakin prices listan läpi käyttäen foreach silmukkaa. Tämän jälkeen määrittelin foreachista tulevan itemin ensimmäisen osan olevan aika ja toisen osan hinta. Tämän jälkeen ajattelin, että if lausetta käyttäen saisin ohjelmaan toivotut ominaisuudet ja näinhän se oli. Lopuksi lisäsin saadut tiedot datagridviewiin.

**Tehtävä D - Toteutus**

Tehtävän anto: Sovelluksen tulisi pystyä kertomaan tietyllä aikavälillä paras päivä bitcoinin ostamiseen ja paras päivä myydä ostettu bitcoin voittojen maksimoimiseksi ja päinvastoin, esim. myy ensin ja osta takaisin myöhemmin.

Ensimmäiseksi määritin 2 methodia nimeltä GetBestDateToSellOrBuy ja GetBestDateToBuyorSell.

Käytännössä GetBestDateToSellOrBuy methodissa myydään ensin ja ostetaan myöhemmin lisää. Tässä piti ensimmäiseksi järjestää lista suurimmasta pienimpään, jonka jälkeen otin ensimmäisen itemin listasta, jossa oli suurin hinta ja tämä oli se myynti hinta. Tämän jälkeen käyn listan läpi foreach silmukalla, jolla saan jokaisen itemin yksitellen listasta. Lisäsin tarvittavat if lauseetta, jolla tarkistin ettei mene sillein, että myyn 10.10.2024 ja ostan 4.10.2024, eli ei mene järjestyksessä ja tarkistan, onko tässä itemissä suurempi profit kuin toisessa. Lopuksi lisäsin saadut tiedot datagridviewiin.

**Loppuraportti**

Minusta onnistuin tehtävän toteutuksessa todella hyvin. Sain kaikki toiminnallisuudet toimimaan halutulla tavalla ja luomaan ohjelman rakenteen mahdollisimman yksinkertaisesti ja tehokkaasti, niin että sitä on helppo myöhemmin muokata tai lisätä ominaisuuksia ilman, että koko rakenne menisi rikki. Antaisin itselleni arvosanaksi 5, koska ohjelma on luotu laadukkaaksi, selkeäksi ja helposti muutettavaksi myös aloittelija tasolle.