

TEKNINEN SUUNNITELMA

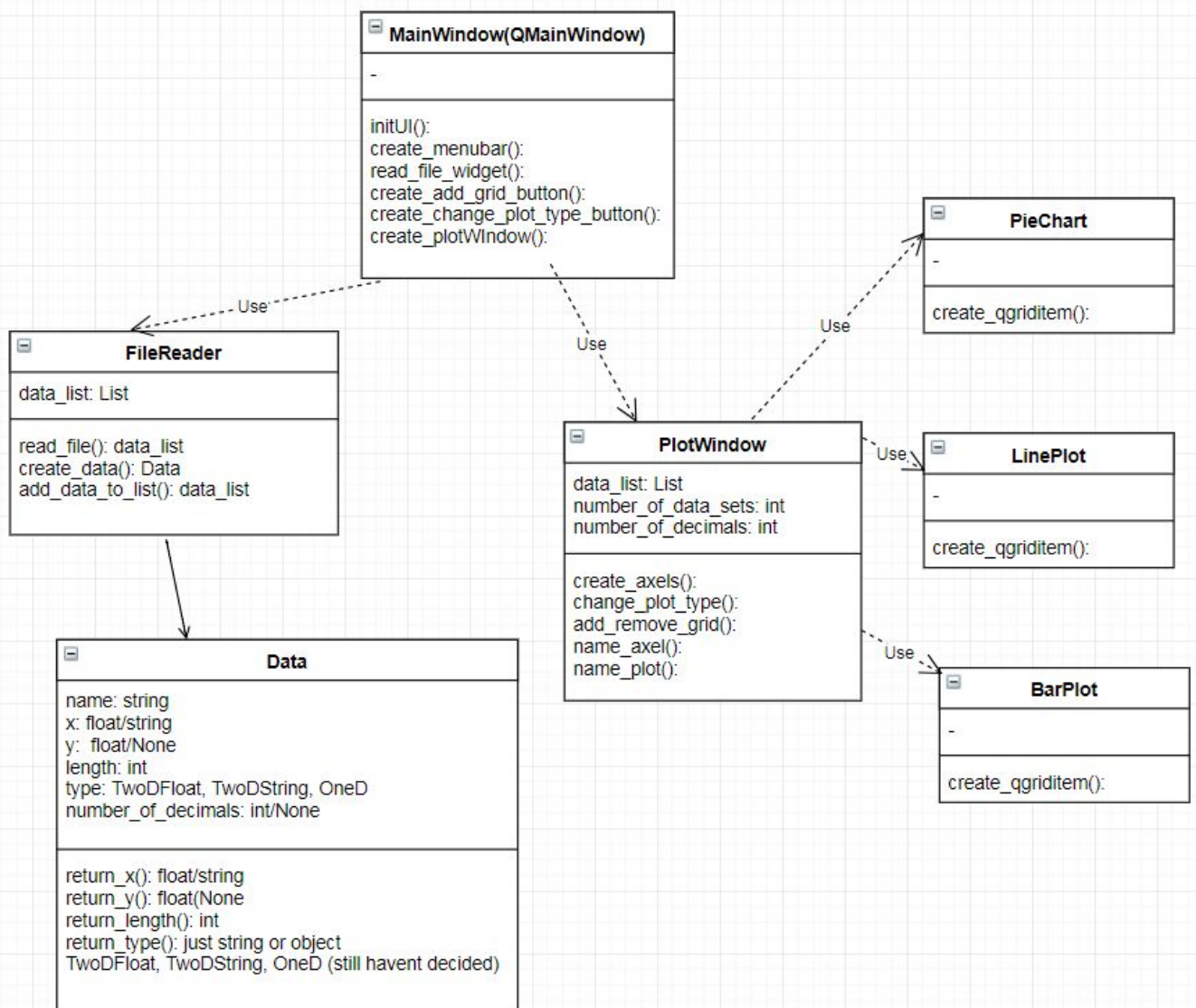
1. Ohjelman rakennesuunnitelma

Ohjelman jako luokkiin:

- **MainWindow(QMainWindow)**
 - Ohjelman käynnistyessä käyttäjälle aukeaa tyhjä ikkuna, jossa yläpalkista voidaan lukea tiedosto. Tiedoston luku aukeaa erilliseen ikkunaan. Tämä alkuikkuna on MainWindow olio, joka on käyttäjän ensimmäiseksi näkemä osa ohjelmasta. MainWindow luokka luo graafisen käyttöliittymän ja koordinoi muiden luokkien käyttöä.
- **FileReader**
 - Tiedosto luku
 - MainWindow kutsuu FileReader oliota, kun käyttäjä aktivoi tiedoston luvun käyttöliittymän read file painikkeella. FileReader tarkistaa, että tiedosto on vaatimusten mukainen, ja luo tiedostosta yhden tai useamman Data-olion. Data-oliot tallennetaan listaksi. Virheellisen tiedoston yhteydessä FileReader nostaa virheen. FileReader palauttaa Data oliot listana MainWindow:lle kuvaajan piirtämistä varten.
- **Data**
 - Jokainen tiedostossa oleva näyte luodaan omaksi dataolioksi, jolla on x ja y koordinaatti, pituus, datan tyyppi ja desimaalien lukumäärä (jos numeerista dataa)
 - desimaalien lukumäärää käytetään koordinaattisysteemin luomisessa ja datan normittamisessa (vielä en ole ainakaan keksinyt parempaa tapaa saada desimaalilukuja liitettyä koordinaatteihin ja piirrettyä)
- **PlotWindow**
 - Kuvaajan piirron ja sivupalkin luomisen hoitaa PlotWindow olio. PlotWindow on joko erillinen ikkuna, tai QMainWindow:n central Window + toolwidget (tällöin jaan tämän luokan vielä mahdollisesti osiin). PlotWindow saa MainWindow:lta parametrina Data oliot sisältävän listan. PlotWindow tarkistaa, kuinka monta datasettiä listassa on, mikä on datan tyyppi, ja kuinka monta desimaalia datassa on. PlotWindow piirtää datan alustavan kuvaajaan ja sivupalkin, jossa näkyvät kaikki käyttäjälle mahdolliset toimenpiteet. PlotWindow myös vaihtaa kuvaajaa käyttäjän sivupalkin valintojen mukaan. PlotWindow käyttää luokkia PieChart, LinePlot sekä BarPlot kuvaajan piirtoon, riippuen sen tyypistä.
- **PieChart**

- Hoitaa piirakkadiagrammin piirron. Yksiulotteiselle datalle (vain x-koordinaatti) lasketaan jokaisen arvon suhteellinen osuus. Jokaisen arvon esiintyvyys esitetään tietyssä osuutena ympyrän alasta. Ympyrän jokainen osa piirretään eri värillä. Piirtää myös taustaruudukon, jos käyttäjä niin valitsee.
- **LinePlot**
 - Hoitaa perusviivadiagrammin piirron. Jokainen datapiste piirretään QGridLayoutin avulla. Piirtää myös taustaruudukon, jos käyttäjä niin valitsee.
 -
- **BarPlot**
 - Hoitaa peruspylväsdiagrammin piirron. Käyttää myös mahdollisesti QGridLayoutia. Piirtää myös taustaruudukon, jos käyttäjä niin valitsee.

UML-KAAVIO:



2. Käyttötapauskuvaus

Ohjelman käynnistyessä käyttäjälle aukeaa tyhjä ikkuna, jossa yläpalkista voidaan lukea tiedosto. Tiedoston luku aukeaa erilliseen ikkunaan. Alkuikkuna on MainWindow olio, joka on käyttäjän ensimmäiseksi näkemä osa ohjelmasta, ja joka hoitaa koordinointia käyttöliittymän piirron ja muun tyyppisten toimenpiteiden välillä.

Seuraavaksi käyttäjä syöttää tiedoston, ja tiedosto luetaan. Tiedoston lukuun MainWindow kutsuu FileReader oliota. FileReader tarkistaa, että tiedosto on vaatimusten mukainen ja luo tiedostosta yhden tai useamman Data olion tallentaen ne listaksi. Virheellisen tiedoston yhteydessä FileReader nostaa oikean tyyppisen virheen. FileReader palauttaa Data oliot listana MainWindow:lle.

Seuraavaksi piirretään kuvaaja. Tämän osuuden hoitaa PlotWindow olio. PlotWindow saa MainWindow:lta parametrina Data oliot sisältävän listan. PlotWindow tarkistaa, kuinka monta datasettiä listassa on, mikä on datan tyyppi, ja kuinka monta desimaalia datassa on. PlotWindow luo alustavasti datatyyppin mukaan joko PieChart (1-D data) , LinePlot (2-D numeerinen data), BarPlot (2-D merkkijonodata) olion. PieChart/LinePlot/BarPlot olio luo datatyyppin mukaisen graafisen kuvaajan joko erilliseen ikkunaan tai pääikkunaan. Käyttäjälle näkyy myös kuvaajan otsikko, akselien nimet, sekä ruksattava valikko mahdollisista eri kuvaajatyypeistä, jotka ovat tälle tietyille datatyyppille mahdollisia. Mikäli käyttäjä muuttaa jotakin muokattavaa ominaisuutta, esim kuvaajan tyyppiä tai akselin nimeä, PlotWindow vastaanottaa signaalin ja tekee muutoksen graafiseen käyttöliittymään.

Käyttäjä voi myös mahdollisesti tallentaa kuvaajansa (ei vielä varmaa).

Kommentteja:

En ole vielä aivan varma, toteutanko kuvaajan piirron täysin erillisessä ikkunassa, vai samassa ikkunassa, missä kuvaajan ominaisuuksia muovaavat palkit ovat. Tämän tiedän vasta, kun pystyn tarkemmin perehtymään pyQt5 toimintaan.

3. Algoritmit

Algoritmi koordinaattisysteemin piirtoon (kuvattu x-koordinaatille, mutta pätee myös y-koordinaatille):

1. otetaan kaikki piirrettävät x arvot.
2. järjestetään x arvot pienimmästä suurimpaan
3. kerrotaan x arvot kokonaisluvuiksi (selvittämällä desimaalien lukumäärä ja kertomalla oikealla kymmenluvulla desimaalit pois)
4.
 - a. mikäli pienin koordinaatti on negatiivinen, normitetaan koordinaatit alkamaan nolasta, lisäämällä minimikoordinaatti kaikkiin koordinaatteihin.
 - b. mikäli pienin koordinaatti on positiivinen, normitetaan koordinaatit alkamaan nolasta, vähentämällä minimikoordinaatti kaikista koordinaateista..
5. nyt meillä on nolasta lähtevä x_norm, jonka pisteet ovat samassa suhteessa toisiinsa kuin aiemmin. Tämä normitettu versio on helpompi piirtää koordinaattisysteemiin.

4. Tietorakenteet

- Käyttäjältä luettava data sopii parhaiten olioksi, sillä käsittelemme useaa eri tyyppistä dataa, ja eri datatyypit tulee pystyä erottamaan toisistaan eri puolilla ohjelmaa. Olion käyttö on tässä tilanteessa esimerkiksi sanakirjaa tai listaa joustavampaa.
- Toisaalta eri puolilla ohjelmaa tarvitaan mahdollisesti useampaa data oliota samanaikaisesti (jos esim piirretään useamman datan viivadiagrammia), Useampi dataolio on kätevää kuljettaa eri puolilla ohjelmaa listana. Valitsin listan sanakirjan sijaan, sillä oleellisin tieto on dataolioiden lukumäärä, eikä yksittäisen data-olion nimellä/avaimella ole merkitystä.

5. Aikataulu

- Arvioitu projektin kesto 80h jaettuna 8 viikolle (10h/vko), jolloin se olisi valmis noin 21.4

FileReader koodaus, Data olion luonti	viikko 9 (10h)
Graafisen ulkonäön lukkoon lyönti ja MainWindow koodaus datan lukuun	viikko 10 (10h)
graafisen osuuden laajennus: PlotWindow	viikko 11 (10h)
LinePlot + testejä tähän	viikko 12 (10h)
BarPlot + testejä tähän	viikko 13 (10h)
PieChart + testejä tähän	viikko 14(10h)
viimeistelyä/aikataulujoustoa, jos joku kohta venyy	viikko 15 (10h)
Testausta kokonaisuutena	viikko 16(10h)

6. Yksikkötestaussuunnitelma

- Oleellinen osa-alue testauksessa on datan lukemisen testaaminen ja kuinka ohjelma käyttäytyy virheellisen datan kanssa:
 - FileReader luokan `read_file()` testaus:
 - syötetään ei- tekstitiedosto
 - FileReader luokan `create_data()` testaus:
 - syötetään tiedosto, jossa liikaa koordinaatteja rivillä
 - syötetään tiedosto, jossa liian vähän koordinaatteja
 - syötetään tiedosto, jossa useampi datasetti
 - syötetään tiedosto, jossa osa arvoista merkkijonoja, osa numeroita
- Toisaalta myös graafisen osan toimivuutta tulee testata:
 - LinePlot luokan `create_griditem()` testaus:
 - syötetään dataa, jossa negatiivisia lukuja ja desimaalilukuja (tulisi pystyä silti luomaan koordinaattisysteemi oikein)
 - PlotWindow `change_plot_type()` testaus:
 - syötetään eri datatyypppeja, jolloin metodin tulee tunnistaa, mitkä ovat mahdollisia visualisointivaihtoehtoja, ja olla visualisoimasta mahdottomia vaihtoehtoja.
- Tässä työssä tulee luoda monta erilaista datasettiä testaukseen, joten apurakennelmilta on vaikea välttyä.

7. Kirjallisuusviitteet ja linkit

<https://www.zeolearn.com/magazine/10-steps-for-getting-started-guis-with-python> 20.2.2019

<https://data-flair.training/blogs/python-pyqt5-tutorial/> 20.2.2019

<https://pynative.com/python-check-user-input-is-number-or-string/> 21.2.2019

<https://www.qt.io/> 23.2.2019

<http://zetcode.com/gui/pyqt5/> 22.2.2019

<https://docs.python.org/3/> 23.2.2019

8. Liitteet

-