

# FinalChartStructure Project

## plot.data:

1. AxisData: Tengely kirajzolásához szükséges adatokat tároló osztály.
  - Tárolt adat(ok):
    - a, TitleData\*: Tengely címének adatait tárolja
    - b, LabelData\*: Felosztáshoz tartozó címke adatait tárolja
    - c, majorStep (double): A nagy felosztás léptéke
    - d, minorStep (double): A kis felosztás léptéke
    - e, LineData\*: Tengely vonalának adatait tárolja
2. BorderData: Grafikon köré rajzolt keret adatait tároló osztály
  - Tárolt adat(ok):
    - a, List<EWNS>: Enum segítségével tárolja a kívánt keret oldalait (égtájak szerint)
    - b, LineData\*: Keret vonalának adatait tárolja
3. DataSeries: Ábrázolni kívánt adatokat tároló osztály
  - Tárolt adat(ok):
    - a, name: Adatsor elnevezése
    - b, samplingFreq: Mintavételezési frekvencia
    - c, data[] (double): Ábrázolni kívánt adatokat tárolja
    - d, timeWindow: Adatok mérésének időintervalluma
    - e, startTime: Mérés kezdetének időpontja
    - f, LineData\*: Adatsor vonalának adatait tárolja
    - g, maxValue: data[] tömb legnagyobb értéke
    - h, minValue: data[] tömb legkisebb értéke
4. GridLineData: Grafikon gridline adatait tároló osztály
  - Tárolt adat(ok):
    - a, List<Direction>: Enum segítségével tárolja, hogy milyen gridline-t szeretnénk berajzolni (Horizontal, Vertical)
5. LabelData: Tengely felosztásához tartozó feliratok adatait írja le
  - Tárolt adat(ok)::
    - a, font: Felosztáshoz tartozó Font
    - b, decimals: Megadja, hogy hány helyiértéket ábrázoljunk
6. LegendData: A Legend adatait tároló osztály
  - Tárolt adat(ok):
    - a, LineData\*: Jelmagyarázat vonalainak adatait tárolja
    - b, TitleData\*: Jelmagyarázat feliratának adatait tárolja
7. LineData: Ábrázolni kívánt vonal adatokat tárol
  - Tárolt adat(ok):
    - a, width: A vonal szélessége
    - b, color: A vonal színe (Color osztály)
    - c, Type: Enum segítségével tárolja, hogy milyen típusú vonalat szeretnénk ábrázolni (SOLID, DASH, DOT\_LINE)

8. MarkerData: Marker adatokat tárol (nincs használva)

- Tárolt adat(ok):

- a, size (double): A marker méretét írja le
- b, shape: Marker típusát adja meg (Shape osztály)
- c, color: A vonal színe (Color osztály)

9. TitleData: Megjelenített szöveg adatait tárolja

- Tárolt adat(ok):

- a, value (String): Szöveg tartalma
- b, font: Szöveghez tartozó Font

### **plot.utils:**

1. RandomDataUtil: Szabályozhatóan véletlen adatokat generáló osztály (teszteléshez)

- Függvény(ek):

- a, getRandomData() : float[] - Véletlen adatokat generál

2. VisualUtility: A koordináta tengelyek felosztásainak koordinátáit generálja le

- Függvény(ek):

- a, xTickCoordinateGenerator() : ArrayList<Double> - X koordinátákat generál
- b, yTickCoordinateGenerator() : ArrayList<Double> - Y koordinátákat generál

### **plot.view:**

1. Axis (extends Group): Tengely (1 db) kirajzolásáért felelős osztály

- Függvény(ek):

- a, Konstruktor: Létrehozza a tengely vonalat, hozzáadja a tickline-okat és a felirataikat is, illetve beállítja a Listener-öket
- b, bind: Tetszőleges propertykhez bind-olja a tengelyvonal koordinátáit

2. Chart (extends BorderPane): Külső BorderPane tárolja a belső BorderPane-t és az abban levő PlotGroup-ot is. Ehhez az osztályhoz tartoznak a user által használt chain függvények, amelyek beállítják a grafikon tulajdonságait, illetve a build függvény ami felépíti a grafikon

3. FigureContainer (extends BorderPane): Belső BorderPane, ennek a centerében található a grafikon

- Tárolt adat(ok):

- a, plotPane: Erre a Pane-re helyezzük a PlotGroup-ot (ez azért szükséges, mert a Group nem méreteződik automatikusan a BorderPane segítségével)

- Függvény(ek):

- a, addPlot(): Ezzel lehet hozzáadni a PlotGroup-ot az osztály plotPane változójához

4. Legend (extends Group): Az ábrázolt DataSeries-ek alapján legenerálja a jelmagyarázatot

5. PlotGroup (extends Group): Egyszerű group kiegészítve custom Propertykkel (width, height), hogy bind-olható legyen

6. Series (extends Group): Legenerálja a DataSeries-ek PolyLine-jait, beállítja a Listenereket
- Tárolt adat(ok):
    - a, seriesLayoutY: Y szerinti eltolást tárolja el (azért szükséges, mert a JavaFX layoutBound-jai katasztrofálisak)