

# Programozás alapjai házi feladat

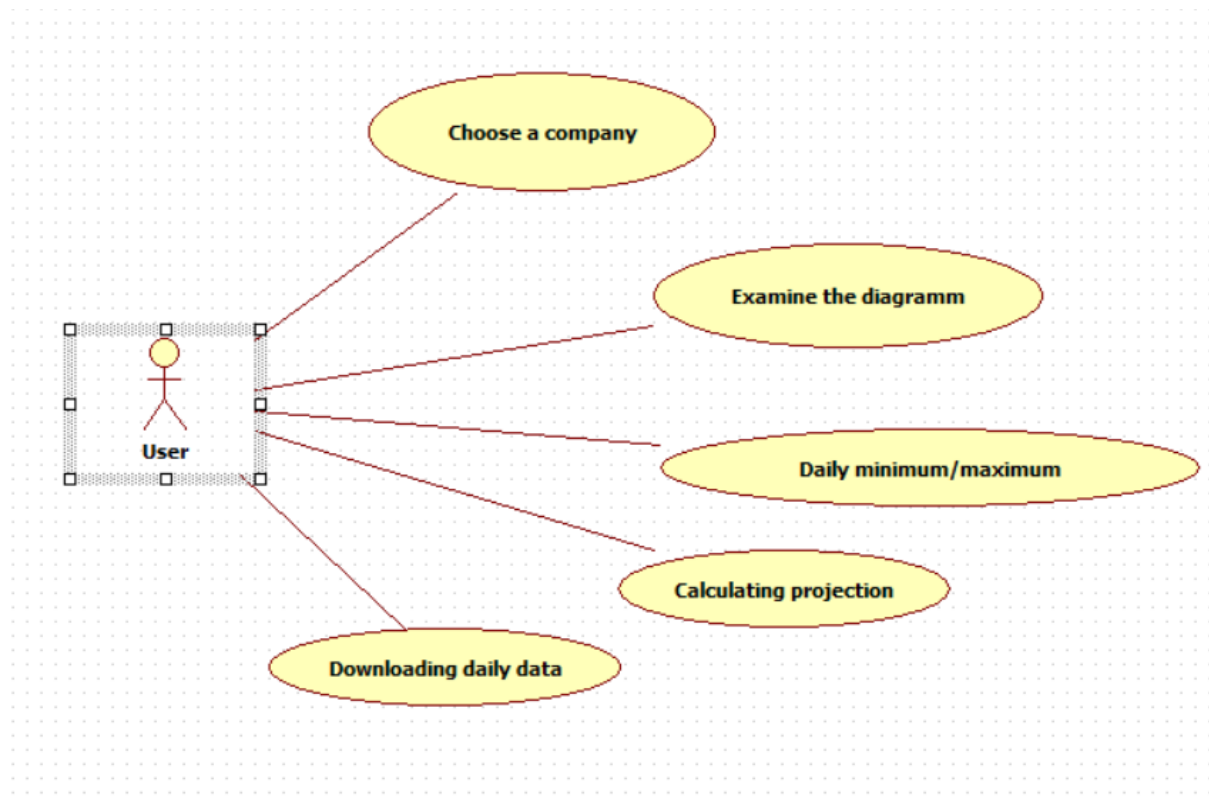
**Szalóki Kristóf**

**DDZ590**

## Specifikáció:

Házi feladatnak egy olyan programot készítek el, mely feldolgozza a tőzsdei tranzakciókat. A program naponta letölt az internetről egy tömörített fájlt, mely az előző, napi tranzakciókat tartalmazza. Ennek feldolgozása során a felhasználó a grafikus felületet használva kiválaszthatja, hogy mely cégnek az adatai érdeklik őt, és így megtudhat számos dolgot róla. A grafikus felületen megjelenik a cég részvényárfolyama egy grafikonon, valamint a napi eladott részvények darabja, valamint a maximális, illetve minimális eladási ár. Hogy a felhasználó többi részvényről is kaphasson ezzel egyidejűleg információt, így a grafikus felületben megjelenik a többi cég részvénye és napi középárfolyama folyamatosan mozogva. Tehát a program a tőzsdén jelenlévő informatikai cégek részvényeit szemlélteti a felhasználók számára könnyen értelmezhetően.

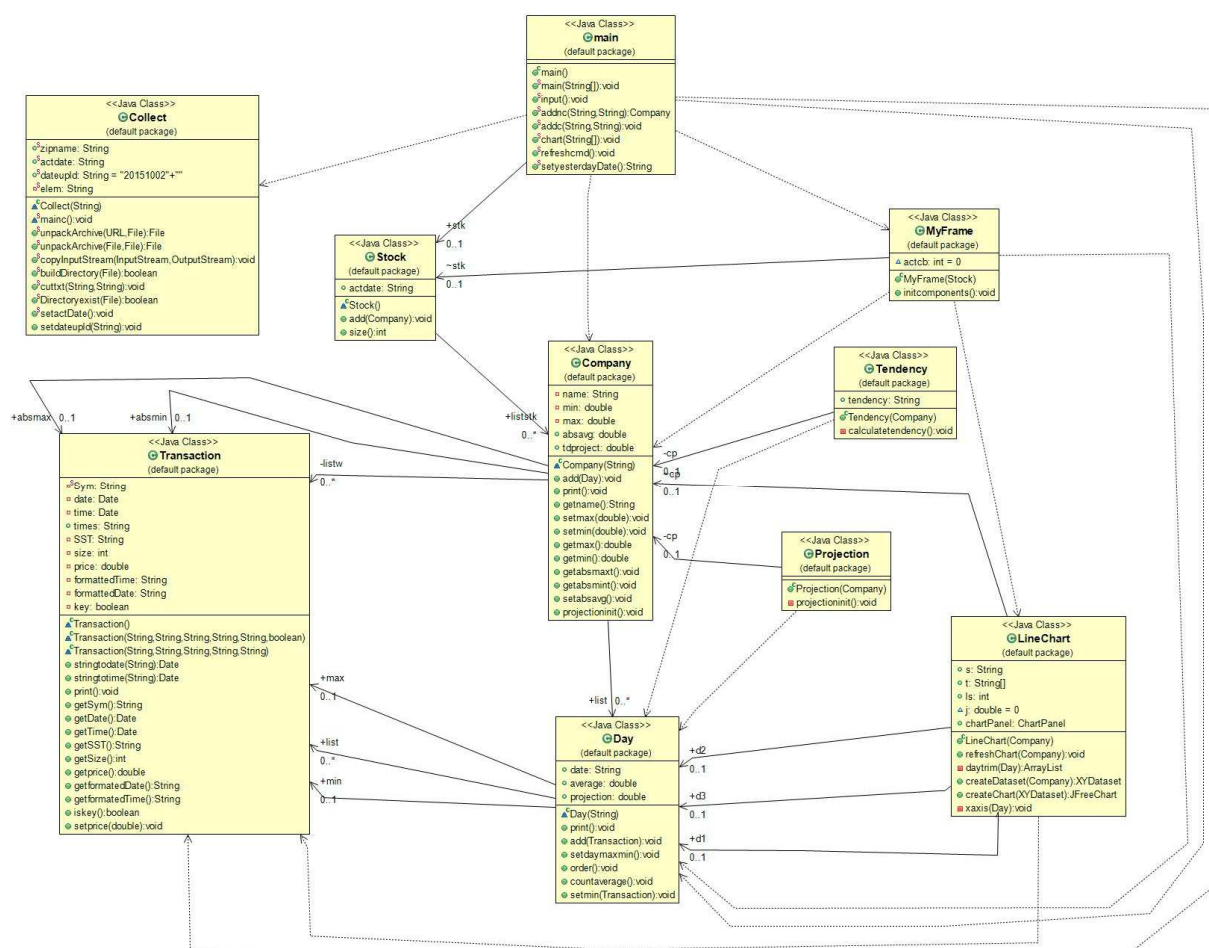
## Use-case diagram :



## Megvalósítás:

Mivel valós adatokkal szeretnék dolgozni, ezért az első lépés, az adatoknak a megszerzése, lesz egy osztályom, mely az adatok összegyűjtésével foglalkozik, benne olyan metódusok, melyek letöltik a fájlt, kicsomagolják, vagy esetleg feltördelik kisebb részekre. Miután megvannak a szükséges adatok, azokat el kell helyezni az erre kialakított adatszerkezetben. A programban lesz a Stock osztály, amely tartalmaz cégeket (Company osztály), ezek a cégek tartalmazzák a tőzsdén kereskedéssel töltött napjaikat (Day osztály), melyekben a tranzakciók találhatók meg (Transaction osztály elemei). Tehát miután a program felhasználható állapotba hozta az adatokat, azok után el kell helyezni azokat ebben a szerkezetben. A program futása során a felhasználó számára ezeknek az adatoknak az egyes részei jelennek meg strukturáltan, tehát például grafikonon ábrázolva, vagy napi minimum illetve maximumkén. Valamint létrekarak hozni majd egy Projection osztályt, melynek metódusai segítségével megadhatom, hogy másnapra milyen lesz a részvény várható árfolyama (ehhez a Valószínűség számításból szerzett tudásomat fogom felhasználni).

## A Program osztálydiagramja:



### **A Program részletes leírása:**

A program az informatika világában jelenlévő tíz nagy vállalat tőzsdei részvényeit figyeli meg , és vizsgálja bizonyos algoritmusokkal. AZ általam vizsgált vállalatok a következők:

- Facebook
- Oracle
- Apple
- Microsoft
- Avgo
- Microsoft
- Epam
- Jcom
- TCX Invesment
- NTES

A program a Bats Exchange nevű amerikai cég oldalát használva letölti az adatokat. A cég minden nap végén kibocsájt egy zip fájlt , melyben megtalálható a cég egész napos kereskedelmi adatai. A zip-ben található txt megközelítőleg 200 MB , és 2 millió sort tartalmaz. Az adatok feldolgozása során a Felhasználó választhat a cégek közül egy legördülő listában ,és megvizsgálhatja az általa választott cégnek a napi teljesítményét illetve az utóbbi tizenkét napban keletkezett összesített eredményeket. A felhasználó rendelkezésére áll egy diagram amin az elmúlt három napi adatok vannak vizualizálva. A projection gombra kattintva becslést kap a következő napra a cég tőzsdei részvényének a középárfolyamáról. Hogy egyszerre több adatról is értesüljünk a felhasználói felület jobboldalán láthatók a cég részvényeinek a napi középárfolyam adatai. A képernyő közepén található egy nyíl , mely a tendenciát mutatja , ez az utóbbi tizenkét napos intervallumra vonatkozik , így nem mindig hasonlít a grafikonon látottakhoz, így ez megtévesztő is lehet.

### **A megvalósítás részletes leírása:**

Mivel a felhasználni kívánt adatok hatalmas méretben érkeznek meg letöltésük után , felkell darabolni azokat , és mappába kigyűjteni. Ez a fájlok méretéből adódóan több időt is felemészt. Az adatok feldarabolása után a programban lévő objektumok feltöltődnek a megfelelő adattal .Az adatszerkezet felépítése down to up módszerrel működik. Először a Tranzakciókat hozza létre , majd Azokat elhelyezi egy nap objektumba , végül a napokat egy céghez rendeli. A hierarchia tetején a Stock objektum áll, mely a tőzsdét hívatott megvalósítani , és így az a cégeket tartalmazza. Az adatszerkezet feltöltése után már az készen áll rá, hogy a függvények meghívásával kiértékelhessük annak bizonyos részeit.

### **Osztályok:**

A megvalósításhoz 11 osztályt hoztam létre, melyek a következők:

- **Collect:** Az osztály célja az interneten lévő adatok letöltés kicsomagolása ,majd feldarabolása. Az osztály feladata elvégzése után az adatok naponként csoportosítva találhatók meg mappákban ,a cég nevével ellátott txt fájlokban.

- **Transaction:** Talán a tőzsdei felépítés legkisebb elemeként lehetne jellemezni , egy nap megközelítőleg 20 ezer tranzakció zajlik le, melyek mindegyike ilyen Tranzakció típusú.
- **Day:** A Day osztály a Transaction osztály elemeit tartalmazza , célja, hogy az egy nap lezajlott tranzakciókat összegyűjtse , és műveleteket hajtson végre rajtuk , mint például a napi maximum , minimum kiválasztása.
- **Company:** A Company osztály a cégeket szimbolizálja , melyek a tőzsdei kereskedés során napokat tartalmaznak , emellett olyan adatok ,melyek egy cégre jellemzők lehetnek , ilyen a cég részvény árfolyamának a tendenciája, illetve a várható értéke.
- **Stock:** A tőzsdét , és annak működését modellezi , egy listában tárolja a tőzsdén lévő cégeket
- **MyFrame:** Az osztály öröklődik a JFrame osztályból , és célja a program grafikus felületének a létrehozása.
- **LineChart:** A grafikus részben található diagram létrehozása, és megjelenítés a megfelelő módon a rendelkezésére álló adatokkal .Az osztályhoz felhasználtam a JFreeChart nevű könyvtárat.
- **MovingText:** Az osztály feladata a grafikus felület jobboldalán látható folyamatosan mozgó részvény nevek árak léptetése és megjelenítése. Az osztály a JPanelből öröklődik.
- **Comparators:** Az osztályban olyan függvények találhatók meg ,melyekkel különféle összehasonlítás végezhető az adatok megfelelő sorrendbe rendezése érdekében.
- **Tendency:** Az osztály által létrehozott objektum az elmúlt 12 napot vizsgálva meghatároz egy növekvő illetve csökkenő tendencia figyelhető e meg.
- **Main:** A program kezdete , ez tartalmaz e Stock ot , mely az egész tőzsdét szimbolizálja.

### **Fontosabb függvények szöveges bemutatása:**

*Collect:*

**public static** File unpackArchive(File theFile, File targetDir): A függvény egy zipfájlt csomagol ki , és visszatér a mappával ,ahol a kicsomagolt fájl található.

*MovingText*

**public void** paint(Graphics g) A függvény a Thread segítségével bizonyos időnként megjeleníti egy adott koordinátára a neki átadott adatot.

*Tendency*

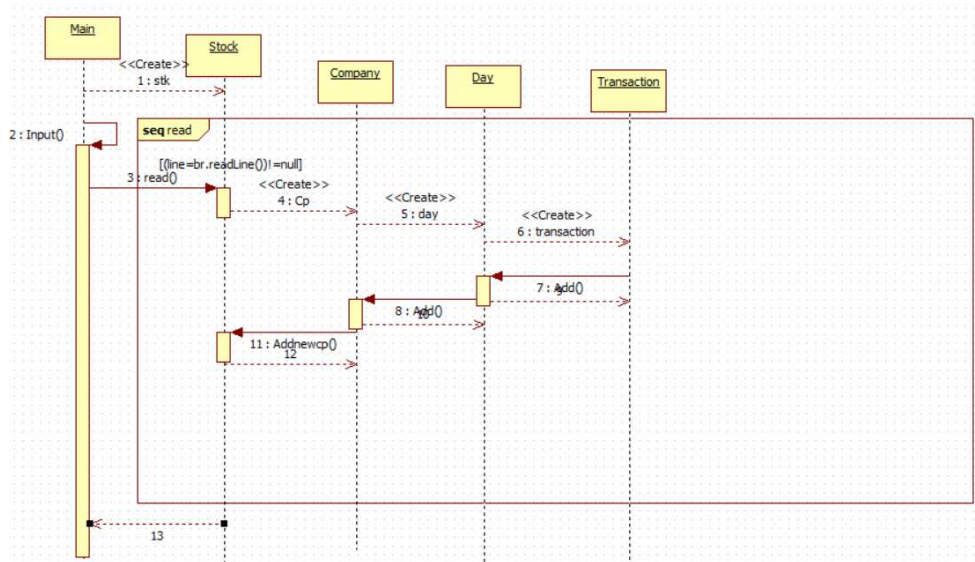
**private void** calculatetendency() Kiszámolja egy cég tendenciáját ,ehhez olyam matematikai módszert használ , mely alapján az elmúlt 10 nap átlagát nézi , és mivel 12 napnyi adatunk van , így az utolsó 3 nap változásáról értesülünk , abban az esetben ha a 3 változás egyre kisebb értelem szerűen csökkenő tendencia , ha nagyobb akkor növekvő ellenkező esetben pedig regnáló lesz a tendenciája az adott cégnek.

*Company*

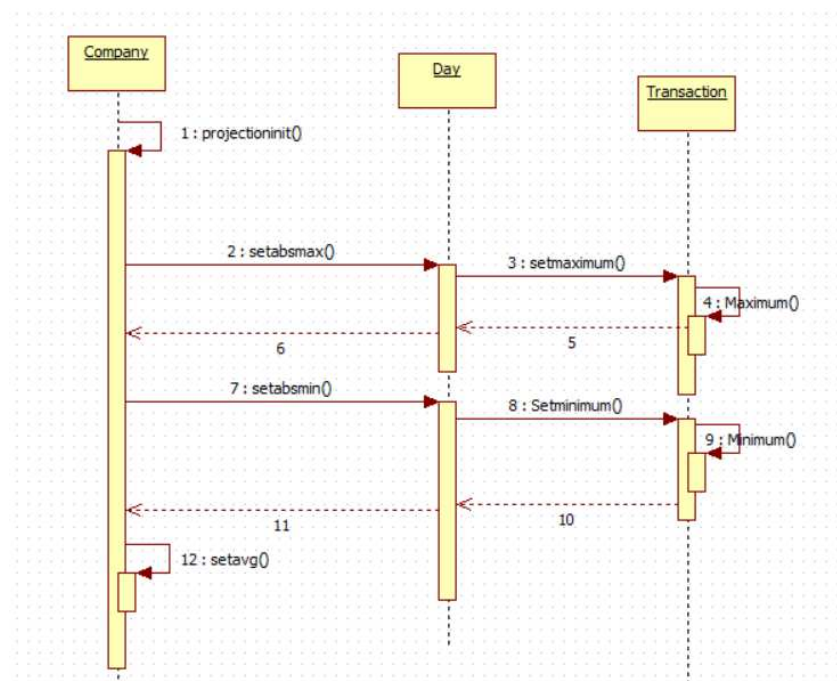
**public void** projectioninit() A függvény becslést ad a részvény következő napi árára, ezt oly módon , hogy az előző napra adott becslést 0.4 el szorozza , még a mai nap átlagát 0.6-al , a kettő összege a becsült érték , ezt a módszert exponenciális el simításnak nevezik.

## Fontosabb működést bemutató szekvencia diagrammok :

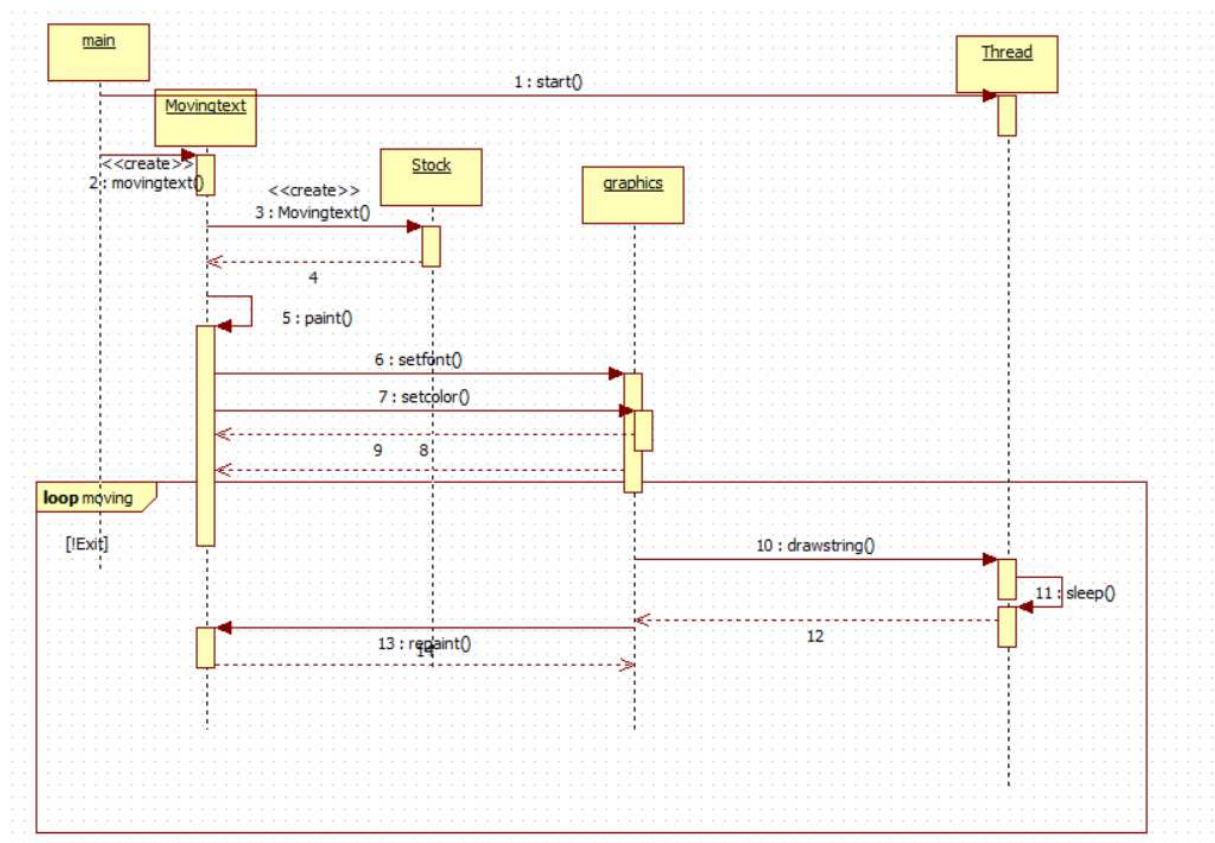
### Inicializálás:



### Projection:



## Movingtext:



## Unit test

A tesztelés során több osztály különböző függvényeit teszteltem a Junit test segítségével. A tesztelés során leggyakrabban az Asser.assertEquals függvényt használtam.

### Személyes megjegyzés :

A program írása során törekedtem arra , hogy csak angol nyelvet használjak , valamint arra hogy a program később továbbfejleszthető legyen , így található benne egy kommentekkel határolt rész , azt később szeretném felhasználni a kódom fejlesztésére. Azok olyan funkciók lesznek majd , melyeket nem tartalmazott az előzetes specifikációm , csupán saját magam fejlesztése érdekében szeretnék megvalósítani , valamint későbbiekben optimalizálni szeretném az adatok beolvasását , mivel rendkívül hosszú időnek gondolom a jelenlegit.

A Jportára a házi feladatnak egy olyan verzióját töltöttem fel , melyből hiányzik a JFreeChartal használt elemek, tehát a grafikon , mivel a könyvtár mérete túl nagy lenne , és nem lehet már akkora méretű fájlt feltölteni a Jportára, így a grafikonos verziót bemutatáson fogom megmutatni . Addig egy kép a működő programról. A grafikkonnal.

