# Beleggingspakket03.1

# 

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

## Aanpassingen voor deze versie (Klaar)

* Lijnen in kunnen tekenen  
    
  Functioneel:  
  Horizontale en oplopende aflopende lijnen tekenen en weer kunnen weghalen  
  Technisch:  
  De lijnen worden in een array van lijnen ondergebracht van het CandleStickChart object

Het tekenen van de lijn(en) wordt uit de loop gehaald voor het toevoegen van de candles.

Lijn in CandleStickChart   
  
Een candlestickChart lijn heeft een relatieve x-index en een absolute y-index coordinaat. De index is relatief ten opzichte van het aantal elementen in de grafiek en loopt van 0 tot aantalBeursdagen-1.

Vertaling van en bereik van coordinaten in de grafiek

In de klasse CandlestickClass worden bij initialisatie de volgende parameters een waarde gegeven aan de hand van de gekozen koersreeks en periode binnen de koersreeks:  
**private void** init(String aGekozenAandeel,  
 **int** aAantalBeursdagenKoersreeks,  
 **int** aAantalDagenRetro)

De volgende velden krijgen daarmee een waarde die de vertaling van een punt in de grafiek naar de x,y coordinaat bepaalt

**private double minPrice** = 1000000.0; *// wordt bepaald aan de hand van gekozen deel van koersreeks***private double maxPrice** = 0.0; *// wordt bepaald aan de hand van gekozen deel van koersreeks***private int aantalBeursdagenGrafiek** = 0;  
**private final int aantalDagenLinks** = 1; *// aantal dagen "leeg" links in de grafiek***private final int aantalDagenRechts** = 3; *// aantal dagen "leeg" rechts in de grafiek*

De x-coordinaat loopt van nul tot aantalBeursdagenGrafiek-1 waarbij de eerste aantalDagenLinks en aantalDagenRechts dummy candles zijn respectievelijk aan de linker en aan de rechterkant van de grafiek

### Event handlers

* De buttons op het scherm zijn via JavaFX scenebuilder gelinkt aan de event handlers in de klasse Grafiekenschermcontroller
* De mouseclick in de grafiek is gelinkt via de klasse CandleStickClass:  
  *// Koppel de mouseclick aan een handler*chart1.setOnMousePressed(mouseEvent -> {

Dit geeft momenteel een logging van de index van de x-coordinaat en een y-coordinaat in absolute zin. Op dit moment is het zo dat als er links van de grafiek wordt geklikt een negatieve waarde wordt gelogd. Bij index 1 is dat de eerste candle en dat komt omdat de eerste candle een positie naar rechts staat. Voor de lol en om te demonstreren wordt er op dit moment een lijn getekend tussen 10,10 en 30,30.

**if** (firstTime == **true**) {  
 firstTime = **false**;  
 **double** x1 = getXAxis().getDisplayPosition(10);  
 **double** y1 = getYAxis().getDisplayPosition(10);  
 **double** x2 = getXAxis().getDisplayPosition(30);  
 **double** y2 = getYAxis().getDisplayPosition(30);  
  
 *//seriesPath.getElements().add(new MoveTo(x1, y1));* MoveTo moveto = **new** MoveTo(x1,y1);  
 seriesPath.getElements().add(moveto);  
 LineTo lineto = **new** LineTo(x2, y2);  
 seriesPath.getElements().add(lineto);

**CandlestickClass**De CandleStickClass is verantwoordelijk voor het initialiseren van de parameters die de mapping bepalen tussen de koersreeks en de grafiek (dwz het bijbehorende deel van de koersreeks mappen op de grafiek), het vertalen van coordinaten naar de bijbehorende candles (vertalen van index tussen grafiek en koersreeks), het plaatsen van de candles en het afhandelen van de events zoomen en pannen in de grafiek.

**CandlestickChart**De candlestickChart is een laag onder de CandleStickClass en verwerkt het feitelijke updaten en intekenen in de grafiek als wel het verwerken van events zoals mouseclicks.

* De Grafiekenschermcontroller beschikt over een referentie naar het object van type CandleStickChart via het veld myCandleStickObject dat via creatie in Main.java wordt meegegeven in de methode toonGrafiekenscherm():

|  |
| --- |
| (…)  **grafiekenschermController**.setCandleStickObject(candlestickObject);  **grafiekenschermController**.setCandleChart(candlestickObject.createContent()); candleStage.show(); |

Voor de duidelijkheid refactoren we de naam *setCandleStickObject* naar *setCandleStickClassObject*.  
  
Liefst willen we de afhandeling van events rondom de lijnen en de mouseclicks zoveel mogelijk binnen de grafiekenschermController afwikkelen. De kennis van de mapping van de coordinaten die zijn geklikt naar de grafiek bevindt zich echter binnen de candleStickClass en dient daar te blijven. De setOnMousePressed van de candleStickChart wordt op dit moment in de CandleStickClass geïnitialiseerd bij createContent. Het is beter als het initialiseren van deze mouse handler verhuist naar het candleStickChart object.

*Waar brengen we nu de logica onder van het tekenen van een lijn?*  
Voorstel:

* De toestand van het al dan niet tekenen van een lijn is ondergebracht bij de GrafiekenschermController. Bij het intekenen van een nieuwe horizontale lijn wordt door de GrafiekenschermController gewacht op een mouse event vanuit het candleStickChart object. Het candleStickChart object wordt zo dom mogelijk gehouden en weet alleen van de mouseclick zelf af en niet van wat die inhoudt.
* De mouseclick in het scherm wordt door het candleStickChart object doorgegeven aan het grafiekenschermController object va dat scherm.
* De lijnen zijn op dit moment nog ondergebracht in een array binnen het candleStickChart object. De grafiekenschermcontroller kan besluiten hier lijnen aan toe te voegen of uit te verwijderen, bij te werken enzovoorts. Dat betekent dat de controller wel toegang moet hebben tot de eigenschappen van die lijnen.  
  In feite is het array van lijnen een apart object tussen de candleStickChart en de controller. In de methode layoutPlotChildren methode van de candleStickChart worden de lijnen opnieuw getekend zodra zich een event voordoet waardoor objecten in visuele zin moeten worden bijgewerkt, bijvoorbeeld het inzoemen, het groter of kleiner maken van het scherm, enzovoorts. De vraag is: van wie is nu dit object, is het van het grafiekenscherm of is het van de controller? Op dit moment is de keuze dat het bij het grafiekenscherm hoort.
* Wil het grafiekenschermController iets weten van de informatie rondom de coordinaten (bijvoorbeeld op welke datum is geklikt) dan raadpleegt deze daarvoor het CandlestickClass object met behulp van deze coördinaten.

Het candleStickChart object dient hiervoor een referentie te bevatten naar het grafiekenschermController object. Inmiddels wordt de mouseclick doorgegeven naar de controller – vanuit grafieken object (CandlestickChart) wordt eerst beoordeeld of er een object geraakt is. Twee mogelijkheden: ofwel een lijn ofwel niks. Afhankelijk daarvan wordt in de controller een andere methode aangeroepen:

Line line = inBuurtVanLijn((**double**) x1, (**double**) y1);  
**if** (line != **null**) {  
 **myGrafiekenschermController**.lineClicked((**double**) x1, (**double**) y1, line);  
} **else** {  
 **myGrafiekenschermController**.mouseClick(x1, y1);  
}

De lineClicked methode beoordeeld aan de hand van de eventState of het de bedoeling is de lijn te verwijderen.

## Functioneel nog open van vorige versies:

|  |
| --- |
| * **Fout: redelijk urgent: GESLOTEN** bij pannen (links/rechts aantal beursdagen) houdt op gegeven moment reactie op. Wordt ook niets gemeld.  Heb ik niet meer gezien dat dit zich voordeed sinds versie 02.5 * **Fout:** **OPGELOST**  bij koersen verversen wordt de volgende maand genomen. Ook beroerd. Op 1 maart wordt afgedrukt  “retrieve prices from internet from 3 - 2020 until 3 – 2020“ wordt in methode updatePriceHistory van klasse GetPriceHistoryafgedrukt in plaats van het ophalen vanaf feb-2020.  **Workaround:** alle koersbestanden weggooien en opnieuw alles ophalen. Duurt wel een poosje. Is tevens een reden om af en toe van de koersbestanden ergens anders veilig een kopie op te slaan. * Fout: op de dag zelf worden de koersen niet meer opgehaald **niet zo urgent kan in een toekomstige versie worden opgelost.** |

**Lijnen – Urgent!**

|  |
| --- |
| * Mogelijkheid om lijnen te tekenen en weer te verwijderen **KLAAR** (oplopende en aflopende lijnen, horizontale lijnen).  inmiddels is voorzien in horizontale lijnen. Nog doen: oplopende en aflopende lijnen * Mogelijkheid om de lijnen te bewerken: **NIET MEER VAN TOEPASSING/KLAAR**  links en rechts laten doorlopen - Niet meer van toepassing, alle lijnen lopen door lijnen clippen links of rechts   lijnen verplaatsen - **KLAAR – kunnen nu aanschuiven**   * Mogelijkheid om lijnen tegen de candles aan te drukken respectievelijk van onder of van boven **Matig** **urgent** Horizontale lijnen - **KLAAR** |

**Tooltip**

|  |
| --- |
| * Volumes en datum tonen in de tooltip**, niet zo urgent Datum wordt inmiddels getoond in de tooltip. Volume nog niet maar volume in de grafiek tonen, relatieve volumes, is belangrijker** |

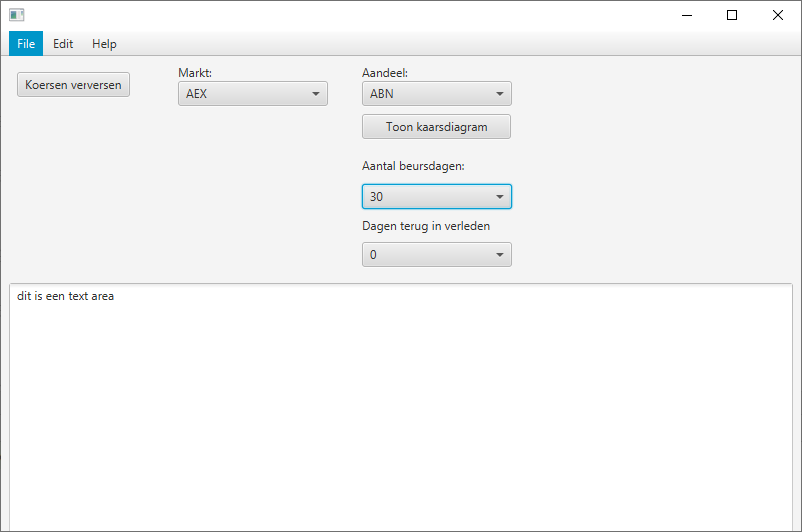
**Indicatoren**

|  |
| --- |
| * MACD in kunnen tekenen **urgent** * RSI in kunnen tekenen **urgent** * OBV in kunnen tekenen **urgent** * Volume in kunnen tekenen **urgent** |

## Technisch nog open van vorige versies

* **nog parametriseren: candle width**
* Het candlestick scherm heeft een eigen controller. - **KLAAR**  
  Binnen deze controller komen alle event handlers te staan die betrekking hebben op de candlestick grafiek. De button handler van de button in het candlestick scherm roept de deze controller aan. Ieder candlestick scherm heeft zijn eigen controller instantie.
* De lijnen worden in een array van lijnen ondergebracht van het CandleStickChart object

Het tekenen van de lijn(en) wordt uit de loop gehaald voor het toevoegen van de candles.  
**KLAAR – bij tekenen van eerste candle worden ook de lijnen ingetekend.**





Candlestick grafiek

Voorlopig ligt de focus even op het correct weergeven van een periode van een bepaald fonds in de vorm van een candlestick grafiek.

De event controller in het MainController object roept de methode showCandlesticks aan van het Main object.  
In het Main object wordt een koersreeks (prices) aangemaakt via de GetPriceHistory module. Voorlopig wordt hier het fonds Philips hardcoded voor gebruikt. Deze koersreeks wordt gebruikt om via de constructor een object van type CandleStickClass aan te maken.

De scene die wordt afgeleverd door het CandleStickClass object toont vervolgens de laatste 31 dagen van deze koersreeks in de vorm van een candlestickgrafiek.

Mouseclick in de grafiek

De button “Klik mij” toont een boodschap in het console van IntelliJ, de event handler daarvan is te vinden in het object van type **CandleStickClass** die een scene aflevert waar de chart van type **CandleStickChart** een deel van uitmaakt.

myButton.setOnAction(**new** EventHandler<ActionEvent>() {  
 @Override **public void** handle(ActionEvent e) {  
 System.***out***.println(**"key event 02"**);  
 }  
});

## Candles tekenen

In de **CandleStickChart** is een methode genaamd **layoutPlotChildren()** die is overridden. De methode tekent alle candles opnieuw en doet de layout van de plot. Binnen de methode is ook een lijn die getekend wordt in de loop van het doorlopen van de data, uiteraard slechts maar één keer, om te demonstreren dat op deze manier een oplopende lijn in de grafiek kan worden getekend. De instructies om de lijn te tekenen worden aan de data van de chart toegevoegd

Series<Number, Number> series = getData().get(index);

De serie wordt eerst leeg gemaakt

**if** (series.getNode() **instanceof** Path) {  
 seriesPath = (Path) series.getNode();  
 seriesPath.getElements().clear();  
}

Vervolgens worden hier de candles aan toegevoegd

**while** (iter.hasNext()) {  
 Axis<Number> yAxis = getYAxis();  
 Data<Number, Number> item = iter.next();

(…)

Node itemNode = item.getNode();

(…)

Candle candle = (Candle)itemNode;  
candle.update(close - y, high - y, low - y, candleWidth);  
candle.updateTooltip(…

### Candle data, candle width, labels, periode, vertikale lijnen en horizontale lijnen

**xAxis** = **new** NumberAxis(0, 30, 5);

Er zijn 31 candles in de grafiek aanwezig, hard gecodeerd (index van 0 tot 31).  
Op dit moment worden bij het aanmaken van het CandleStickClass object een array van dagprijzen meegegeven

**public** CandlestickClass(ArrayList<DayPriceRecord> dayPriceArray) {  
 **myDayPriceArray** = dayPriceArray;  
}

De laatste 31 DayPriceRecord objecten uit de lijst myDayPriceArray worden vervolgens gebruikt om de candlegrafiek te vullen (CandleStickChart object).

Op dit moment wordt de breedte van de candle binnen de CandleStickChart vastgesteld als zijnde

**double** unit = xa.getDisplayPosition(xa.getTickUnit());  
candleWidth = unit \* 0.2;

waarbij de getTickUnit() functie de Tick unit gebruikt uit de x-as.

De tick unit heeft een initiële waarde van 5 gekregen bij het aanmaken van de chart in de createContent methode van het CandlestickClass object. Dat betekent dat om de vijf beursdagen een vertikale lijn wordt getekend.

De tick unit voor de y-as is op het getal 1 gesteld, voor iedere gehele euro een horizontale lijn.

**yAxis** = **new** NumberAxis( 41, 46, 1);

De vertikale range in de grafiek is tussen 41 en 46 euro. Het fonds dat steeds wordt getoond is Philips. De Xas en Yas labels worden op respectievelijk “Datum” en “Prijs” gezet, hardcoded.

Horizontaal worden als labels gebruikt voor de ticks de datums uit de bijbehorende candle. Alleen de datums uit de tick candles (dus iedere vijf beursdagen) worden getoond.

Dit wordt geregeld door de tickLabelFormatter() methode van de X-as als waarde te geven van een methode via de methode setTickLabelFormatter van het X-as object:

**xAxis**.setTickLabelFormatter(**new** StringConverter<Number>() {  
 @Override  
 **public** String toString(Number object) {  
 **int** index = **myDayPriceArray**.size() - 31 + object.intValue(); *// let op zero based* DayPriceRecord dpr = **myDayPriceArray**.get(index);  
 String datum = dpr.printDate();  
  
 **return** datum;  
 }  
  
 @Override  
 **public** Number fromString(String string) {  
 **return** 0;  
 }  
});

Lijnen tekenen  
  
Binnen de methode **layoutPlotChildren()** is ook een lijn die getekend wordt in de loop van het doorlopen van de data, uiteraard slechts maar één keer, om te demonstreren dat op deze manier een oplopende lijn in de grafiek kan worden getekend.

**if** (firstTime == **true**) {  
 firstTime = **false**;  
 **double** x1 = getXAxis().getDisplayPosition(10);

(…)

#### Teken ter demonstratie een lijn

De instructies om de lijn te tekenen worden aan de data van de chart toegevoegd

*//seriesPath.getElements().add(new MoveTo(x1, y1));*MoveTo moveto = **new** MoveTo(x1,y1);  
seriesPath.getElements().add(moveto);  
LineTo lineto = **new** LineTo(x2, y2);  
seriesPath.getElements().add(lineto);

De output als voorbeeld van het console:

clicked!

showCandleSticks aangeroepen voor markt AEX

key event 02

key event 02

button pressed

23.98936170212766, 40.215384615384615

## Technische wijzigingen

Om dit allemaal wat netter en bruikbaarder te maken ga ik het volgende doen:

Technische wijzigingen en verbeteringen

* Parametriseren van de hardgecodeerde constanten en de waarden die periode, candle width  
  geparametriseerd: periode in beursdagen, dagen in verleden (retro)

**nog parametriseren: candle width**

* Het candlestick scherm heeft een eigen controller.   
  **Via JavaFX is nu geregeld dat het .fxml scherm verbonden is met deze controller.   
  De GUI kan nu via java scenebuilder worden herzien. Doorbraak! Champagne!**  
  Binnen deze controller komen alle event handlers te staan die betrekking hebben op de candlestick grafiek. De button handler van de button in het candlestick scherm roept de deze controller aan. Ieder candlestick scherm heeft zijn eigen controller instantie.

* De lijnen worden in een array van lijnen ondergebracht van het CandleStickChart object

Het tekenen van de lijn(en) wordt uit de loop gehaald voor het toevoegen van de candles.

**te implementeren in volgende versie 03.0**

## Functionele wijzigingen

Geïmplementeerd of deels geïmplementeerd:

|  |
| --- |
| * Mogelijkheid om een ander fonds te kiezen **– Klaar, voorlopig alleen AEX fondsen.** * Koersinformatie ophalen – **Nu mogelijk. Alleen laatste beursdag wordt nog niet opgehaald zolang iex de laatste koersdag nog niet heeft verwerkt. Is niet zo erg, zou wel handig zijn als dit wel weer werkt.** * Breder startscherm zodat de buttons te zien zijn en de labels. **Klaar**. * Links en rechts in de grafiek wat ruime – **klaar**, maar zie punt over tick units * Inzoemen automatisch op bereik vertikaal, nog niet ideaal, zie bijv KPN. **Klaar.**   Inzoemen op vertikaal bereik: eerst hoogste en laagste koers bepalen in gekozen periode. KoersHoog en koersLaag. Een vast percentage 10% boven hoogste en onder laagste koers.   * Rekening houden met ontbrekende beursdagenin de reeks **On hold, niet zo belangrijk** Ontbreken er candles omdat de koersen van bepaalde dagen niet beschikbaar zijn dan moeten hier dummy candles worden ingevuld met een negatieve koers zodat ze   niet te zien zijn (ze worden dan automatisch geclipt**).**   * Lengte van periode in beursdagen kunnen kiezen **Klaar** De gebruiker kan via een uitklaplijst kiezen hoeveel beursdagen hij wil zien. * Candle breedte aan de hand van aantal candles in grafiek – **On hold, niet zo belangrijk.** Mogelijkheid om een andere deelperiode te kiezen waarbij de lengte van de periode   bepaalt wat de grootte is van de tick unit. De gekozen periode moet minimaal één week  bedragen die binnen de beschikbare beursdagen vallen.   * Maandagen en laatste candle krijgen datum tick en vertikale lijn. **On hold, niet zo belangrijk.** Dat betekent dus dat er begonnen moet worden bij een maandag binnen de gekozen   periode. Echter: Laatst aanwezige beursdag moet op een vertikale tick vallen zodat je daar  de datum van ziet.   * Periode in het verleden kunnen kiezen, **Klaar** * Kunnen pannen horizontaal over aantal beursdagen, x erbij of eraf, **Klaar** * Kunnen in- en uitzoomen horizontaal over aantal beursdagen, x erbij of eraf, **klaar** * Sizing vertikale as onvoldoende, **klaar** Vertikaal is de grafiek goed gesized maar bij pannen en zoomen wordt niet bijgeschaald. Wanneer window wordt vergroot de grafiek nog laten aanpassen, is een property volgens mij. **– probleem doet zich niet meer voor** |

### Vertaling tussen grafiekindex en koersindex

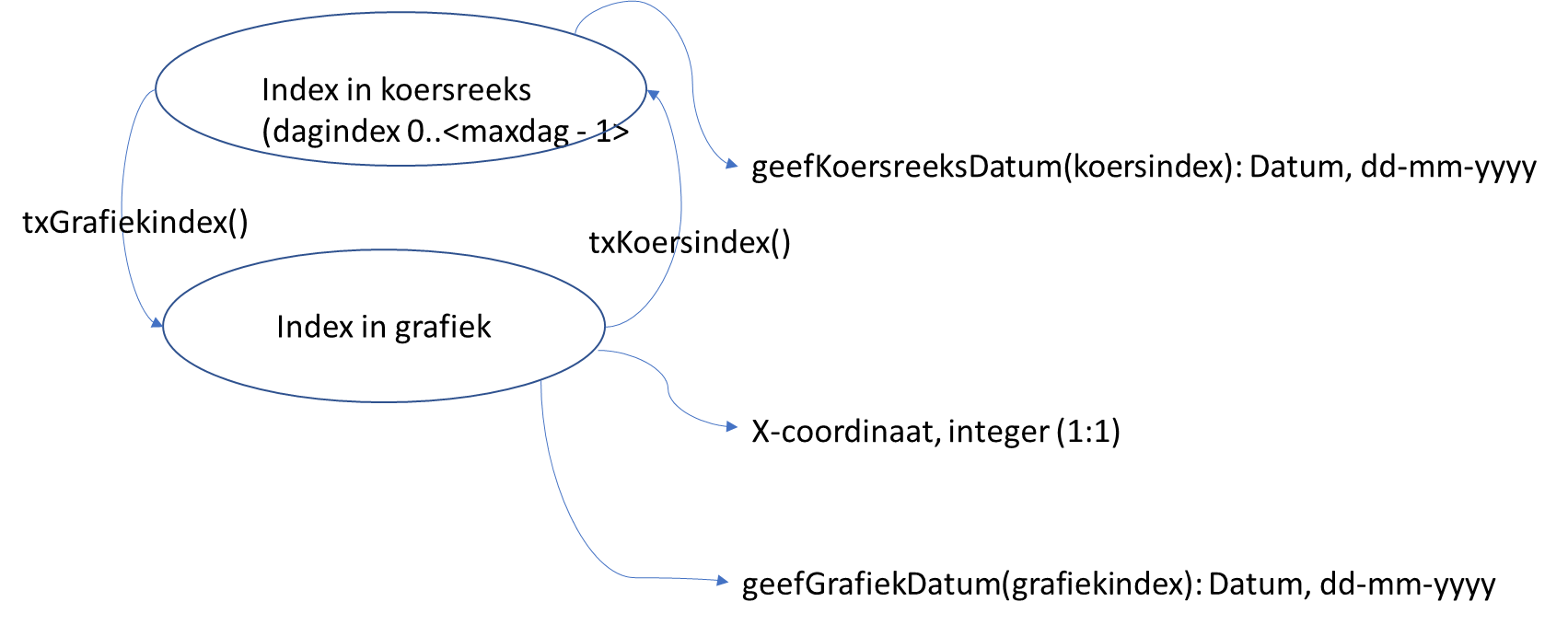
Er is een vertaling nodig van index in de grafiek versus index in de koersreeks. Bij iedere index in de grafiek hoort impliciet een datum.

De koersreeks heeft een beginindex en een eindindex

De index in de koersreeks loopt van beginindex tm eindindex – 1

Omdat er beursdagen in de koersreeks kunnen missen en omdat er links en rechts een hoeveelheid vrije ruimte is in de grafiek kan het zijn dat de index in de grafiek hier en daar overeenkomt met een dummy candlestick.

De methoden txGrafiekindex(koersindex) en txKoersindex(grafiekindex) vertalen van koersindex naar grafiekindex en vice versa.



**Begrippen:**  
startindex de index in de koersreeks die overeenkomt met de eerst getoonde candle   
eindindex de index in de koersreeks die overeenkomt met de laatst getoonde candle  
aantalBeursdagenKoersreeks = eindindex – startindex + 1

koersindex is de relatieve index in het gekozen stuk van de koersreeks.  
koersindex is tussen 0 en <aantalBeursdagenKoersreeks> -1

grafiekindex is de relatieve index in de grafiek, begint bij 0 en telt door tot aantalGetoondeBeursdagen

Let op: vanwege het ontbreken van bepaalde beursdagen in de koersreeks en een aantal niet getoonde candles links en rechts in de grafiek wijkt grafiekindex af van de koersindex.

Een dummy candle is een candle met negatieve koersen die er slechts voor dient om ruimte te scheppen in de grafiek.

aantalBeursdagenLinks – het aantal beursdagen links in de grafiek die altijd zonder candle zijn (dummy candle) om wat lege ruimte links te scheppen  
aantalBeursdagenRechts – het aantal beursdagen rechts in de grafiek die altijd zonder candle zijn (dummy candle) om wat lege ruimte rechts te scheppen  
aantalGetoondeBeursdagen is gelijk aan het aantal ontbrekende beursdagen in het getoonde stuk van de koersreeks + aantalBeursdagenLinks + aantalBeursdagenRechts.

**Index binnen de koersreeks**

koersreeksindex is de index in de koersreeks = startindex + koersindex

**index Range binnen de grafiek**

De grafiekindex hangt van de koersindex af met een afhankelijkheid van de volgende parameters:

aantalBeursdagenLinks  
aantalBeursdagenRechts  
aantalGetoondeBeursdagen   
aantal missende handelsdagen binnen de koersreeks tussen startindex en koersindex (aanvankelijk stellen we die voorlopig op nul, dit voor de eenvoud. Het komt maar zelden voor dat van een bepaald fonds beursdagen missen).