

Moduł GUI – dokumentacja

Jeśli nie napisano czy jakaś funkcja coś zwraca to nic nie zwraca.

Jeśli coś nie ma za sobą nawiasów to nie jest metodą, tylko polem.

Przedstawiono głównie atrybuty odpowiadające za logikę i sposób działania, natywne elementy GUI będą pomijane.

Klasa MainWindow

Jest to klasa w której tworzymy i uruchamiamy okno aplikacji oraz ładujemy elementy GUI z modułu mainGui.

Konstruktor:

`__init__(self, parent=None):`

obiekt klasy MainWindow dziedziczy po klasie Qwidget

Pola i metody:

gui – kontener, obiekt klasy mainGui.GuiMainWindow zawierający wszystkie obiekty GUI

closeEvent(event) – metoda wywoływana w odpowiedzi na sygnał zamknięcia głównego okna aplikacji, W metodzie zapisywana są obecnie otwarte zakładki z wykresami i zapisywane do pliku o nazwie tabHistory.wsf oraz zapisywane są również obecne ustawienia z zakładki Settings do pliku settingsList.wsf. Zapisywane są również pobrane podczas bieżącej sesji obiekty finansowe do pliku save.wsf

keyPressEvent (self, QkeyEvent) – przeciążenie metody klasy Qwidget. Obsługa trybu FullScreen na zdarzenie naciśnięcia F11- przełączanie się z trybu i ESC – wyjście z trybu pełnoekranowego.

Klasa GuiMainWindow

Jest to klasa odpowiedzialna za stworzenie wszystkich elementów GUI i częściową komunikację z pozostałymi modułami, załadowywanie i inicjalizację modelu danych dla obiektów finansowych

Pola i metody:

indexModel, stockModel, forexModel, resourceModel, bondModel, futuresModel – modele przechowywania danych dla indeksów giełdowych klasy mainGui.ListModel

home – obiekt klasy home.Home, przechowujący stronę startową

tabA - zakładka odpowiadająca za wyszukiwanie obiektów finansowych

settingsTab – zakładka zawierająca i umożliwiającą konfigurację własnych ustawień dotyczących wyświetlania formacji wykrytych przez moduł analizy technicznej formacji

compare() - metoda tworzy wykres w trybie porównania kilku obiektów na jednym wykresie. Metoda jest uruchamiana gdy w oknie Search jest włączony tryb Enable compare, oraz wpisane są co najmniej dwa symbole instrumentów finansowych. Następnie funkcja wyszukuje odpowiednich modeli i wywołuje metodę newCompareTab

newCompareTab(qModelIndex, nameTab, listName) – funkcja na podstawie listy obiektów finansowych przekazanych w qModelIndex tworzy nową zakładkę klasy TabA w trybie porównywania wykresów. (przekazanie listy modeli obiektów finansowych)

newIndexTab(), newStockTab(), newForexTab, newBondTab,newResourceTab, newFuturesTab() - argumenty:

qModelIndex – model obiektu finansowego

nameTab – symbol obiektu finansowego

settings –lista zawierająca aktualne opcje wyświetlania wykresu

tabType – jeden z typów (index ,stock , forex, bond,, resource, furures)

Metoda tworzy odpowiedni obiekt klasy TabA w trybie wyświetlania wykresu dla konkretnego obiektu finansowego

settings() - funkcja pobiera aktualnie zaznaczone opcje z tabA(początek i koniec okresu wyświetlania wykresu listę zaznaczonych wskaźników, listę zaznaczonych oscylatorów , typy wyświetlania wykresu, typ wyświetlania skali, włączona opcja możliwości rysowania linii na wykresie , ukrycie Wolumenu. Funkcja zwraca słownik z ustawieniami

closeTab(i) – funkcja wyłącza zakładkę o indeksie i . Zakładek Home , Search , Settings nie da się wyłączyć

findIndexModel(name) – funkcja szuka numeru indeksu modelu obiektu finansowego o nazwie name i jego typu zwraca krotkę (typ, numer)

tabHome(name) - metoda szuka i wyświetla zakładkę z wykresem z symbolem name. Jest wywoływana na zdarzenie naciśnięcia na symbol obiektu w zakładce Home

bigFiltre(text) – metoda filtruje modele danych dla wyrażenia regularnego text i wyświetla tylko dopasowane wzorce, Uruchamiana w odpowiedzi na wpisanie tekstu do pola tekstowego nad listami instrumentów finansowych.

nasdaqFiltre(),nyseFiltre(),wigFiltre(),amexFiltre(),wig20Filtre(),allFiltre() - funkcje pomocnicze do bigFiltre, edytują odpowiednio wyrażenie regularne

ourWebsiteTab() - metoda uruchamiającą zakładkę z możliwością przeglądania strony projektu

class ListModel

- klasa definiująca modele przechowywania listy dla poszczególnych instrumentów finansowych

Klasa TabA

-Obiekt klasy widget odpowiedzialny za wyświetlanie zakładki do wyszukiwania obiektów finansowych oraz ustawiania podstawowych opcji wyświetlania wykresu. W przypadku przekazania do konstruktora modelu obiektu finansowego widget jest w trybie wyswietlania wykresu, jeżeli przekazana jest lista modeli , wtedy wyświetlamy porównywanie kilku obiektów na jednym wykresie

Konstruktor:

__init__(finObjType = None,indexModel=None,stockModel=None,forexModel=None,bondModel=None, resourceModel = None,futuresModel = None, qModelIndex = None,settings = None,listName=None,showLists = True):

finObjType – typ obiektu finansowego

qModelIndex – obiekt finansowy, w przypadku trybu porównania wykresów lista obiektów finansowych

settings – ustawienia opcji wykresu

listName, showList – tryb wyświetlania listy jeśli showList = false wyświetlamy TabA w trybie wyszukiwania obiektów finansowych w przeciwnym przypadku jako zakładkę wyświetlającą wykres

Pola i metody:

initUi() - załadowanie elementów GUI

clearDrawnFormations() - czyszczenie na wykresie narysowanych linii

newAnalyzeTab() - utworzenie obiektu Strategy z modułu analizy technicznej i utworzenie nowej zakładki z wynikiem analizy.

ShowChartPatterns() - metoda pobiera wartości ustawione w zakładce Settings i ustawia w nowo utworzonych obiektach Strategy i FormationDrawer z modułu analizy technicznej oraz rysuje wybrane i znalezione formacje.

updateScale(), updateChartType(), updateStep(), updateDate(), updateOscillator(), updateHideVolumen(), updateEnablePainting(), compareChanged(), smaChanged(), emaChanged(), wmaChanged(), bollingerChanged(), updateIndicator(), updateDrawTrend(),
-metody do odświeżania i odrysowywania wykresu

checkDate() - metoda sprawdzająca poprawność wpisania dat okresu wyświetlania wykresu w polach daty. Sprawdzenie czy data początkowa nie jest > niż data końcowa

addSymbolToCompareLine() - w trybie porównania wykresów dodanie do linii porównania klikniętego symbolu obiektu finansowego na liście

paint2Chart() - główna metoda rysująca wykres z odpowiednimi parametrami w trybie pojedynczego instrumentu finansowego

paintCompareChart() - główna metoda rysująca wykres z odpowiednimi parametrami w trybie listy instrumentów finansowych

setOption() - metoda ustawia opcje w menu pod wykresem z jakim został wywołany w zakładce Search

getSettings() - funkcja pobiera aktualnie zaznaczone opcje z Search

showChartsWithAllIndicators() - tworzymy nowe okno Popup klasy MyPopup i wyświetlamy wykresy obok siebie z wybranymi wskaźnikami artykułu

Klasa RSSReader

Klasa odpowiedzialna za parsowanie dokumentu html i stworzenie struktury artykułu RSS zawierający tytuł datę powstania , opis i hiperłącze

Konstruktor

__init__(self,RSSUrl) – klasa wczytuje i parsuje adres RSSUrl do xml.

Metody i pola:

getXMLDocument(self,RSSUrl) – funkcja czyta adres url i zwraca dokument xml

getItemText(xmlNode) – funkcja pobiera wszystkie wartości tekstowe z przekazanego węzła xml

getChildText(xmlNode, childName) – funkcja wczytuje i zwraca synów węzła xmlNode

createRSSItem(itemNode) - tworzy i zwraca obiekt RSSItem

getItems(self) – generowanie i pobieranie atrybutów obiektu rss.

Klasa RSSWidget

Widget odpowiedzialny za wyświetlanie i możliwość dodawania kanałów RSS

Konstruktor:

__init__(parent=None) – obiekt dziedziczy po klasie QWidget

konstruktor wczytuje adresy url z pliku rss.txt i tworzy obiekty klasy RSSReader

Pola i metody:

rssSite – lista obiektów klasy RSSSite

descriptionList – aktualnie wyświetlana lista artykułów

initUi() - inicjalizacja elementów GUI

addRSS() - metoda dodająca nowe adresy do listy rss i pobierająca dane

removeRSS() - metoda usuwająca zaznaczony rss z listy

showRSS() - wyświetlanie wszystkich dostępnych obiektów RSSSite

showItems() - funkcja wyświetlająca dostępne artykuły z wybranego obiektu RSSSite

showDescription() - funkcja pokazująca opis wybranego artykułu

Klasa RSSSite

Klasa pomost pomiędzy klasą RSSReader a RSSWidget

Pola i metody:

rssReader – obiekt klasy RSSReader

Name – nazwa obiektu

Klasa Analize

Klasa wyświetlająca w przeglądarce tekstowej wynik analizy obiektu finansowego

Klasa dziedziczy po QWidget

Pola i metody:

textBrowser - obiekt wyświetlający wynik analizy technicznej

Klasa Settings

Ramka odpowiedzialna za dostarczenie interfejsu do wyboru sposobu wyświetlania formacji wykrytych przez moduł analizy technicznej. Włącznie / wyłączanie danej formacji, wskaźnika etc. wybór koloru , stylu linii , grubości linii. Klasa zapewnia metody do pobierania aktualnych ustawień.

Konstruktor:

`__init__(settingsList = [])` Klasa dziedziczy po klasie `QFrame` , argument `settingsList` zawiera listę zmienionych ustawienia przez użytkownika pusta lista powoduje załadowanie domyślnych wartości

Pola i metody:

`getVal()` - pobieramy aktualne ustawienia i zwracamy w postaci list list

`loadVal(self, lis = [])`: załadowywanie zapisanych ustawień w lis lub wartości domyślnych jeśli lis = []

`initUi(self)` -inicjalizacja interfejsu

`class ColorComboBox` – klasa dziedzicząca po klasie `QcomboBox` dostraczająca element Gui pozwalający na wybór koloru z rozsuwanej listy

`class LineStyleComboBox` – klasa dziedzicząca po klasie `QcomboBox` dostraczająca element Gui pozwalający na wybór stylu linii

Klasa Home

Klasa dostarczająca zakładki wyświetlającej najbardziej popularne i najbardziej aktywne instrumenty finansowe największe spadki i wzrosty. Ponadto wyświetlany jest Widget RSS umożliwiający dodawanie wyświetlanie kanałów rss.

Konstruktor:

`__init__(self, topList = None, mostList = None, gainerList = None, loserList = None, finObjList = None)` – klasa dziedzicząca po `QWidget` , argumentami funkcji są listy zawierające wartości najbardziej popularnych obiektów finansowych,

Pola i metody:

`topList, mostList, loserList, gainerList, finObjList` – wartości wyświetlanych obiektów finansowych

`initUi()` - inicjalizacja GUI

`addTopObject(objList, k)` – funkcja tworzy kolejne ramki i dodaje u góry zakładki wraz z wykresami dla podanych wartości dla obiektu finansowego w `objList` , `k` – kolejny indeks

`addTable(objList2)` - funkcja tworzy kolejne tabeli i dodaje u lewej strony zakładki dla podanych wartości dla obiektu finansowego w `objList2`

`updateTopList()` - metoda do uaktualniania górnych ramek z aktualnie pobranych danych

`updateTable()` - metoda do uaktualniania wartości tabeli aktualnie pobranymi danymi

updateHome() - funkcja wywołująca metody updateTable i updateTopList

startUpdating() - funkcja startująca wątek odpowiedzialny za uaktualnianie wartości w ramkach i tabelach

tableClicked() - kliknięcie na nazwę symbolu w tabelce powoduje włączenie wykresu wybranego obiektu finansowego

class UpdateThread klasa dziedzicząca po QThread odpowiedzialna za uruchomienie wątku uaktualniającego dane w tabelkach i górnych ramkach

class MyFrame – klasa dziedzicząca po QFrame , dostarczająca ramki niezbędnej do wyświetlania górnych obiektów finansowych w zakładce Home, oraz definiująca przechwycenie zdarzenia kliknięcia myszką na daną ramkę i wysłania sygnału który będzie przechwycony przez metode tabHome() w klasie GuiMainWindow