

# Opis architektury aplikacji

## Podział na moduły:

Aplikacja WallStreetFighters składa się z następujących modułów:

- Moduł wykresów
- Moduł analizy technicznej
- Moduł pobierania i przechowywania danych
- Moduł GUI

## Moduł wykresów

Jest to moduł odpowiedzialny za wyświetlanie wszystkich wykresów, które pojawiają się w aplikacji. Jego trzon stanowią klasy: Chart, CompareChart, Lightweight reprezentujące odpowiednie typy wykresów. Klasa FormationDrawer odpowiada za zaznaczanie na wykresach wykrytych przez moduł analizy technicznej formacji. Natomiast klasa ChartData opakuje model danych, które są przedstawiane na wykresie. Moduł wykresów jest jedynym modulem wykorzystującym bibliotekę matplotlib.

## Moduł analizy technicznej

Ten moduł odpowiada za wszelkie obliczenia, które pojawiają się w aplikacji, to jest obliczanie wskaźników i oscylatorów (podmoduł oscillators), wykrywanie trendów i formacji (podmoduł trendAnalysis) oraz analizę wykresów świecowych (podmoduł candles). Ponadto moduł ten zawiera także klasę Strategy, odpowiedzialną za przechowywanie ustawień użytkownika (tzn. wartości przypisanych poszczególnym sygnałom) oraz za wykrywanie i wartościowanie sygnałów zdefiniowanych w tych ustawieniach. W tym module wykorzystywana jest biblioteka do obliczeń numerycznych NumPy.

## Moduł pobierania i przechowywania danych

Moduł ten umożliwia pobieranie przez aplikację aktualnych danych dotyczących instrumentów finansowych oraz zapisywanie tych danych w pliku, dzięki czemu nie ma konieczności ich ponownego pobierania w przyszłości.

[Marcinie, opisz w paru słowach co robią Twoje klasy/pliki!]

## Moduł GUI

Moduł ten spełnia dwie role. Z jednej strony jest odpowiedzialny za wyświetlanie graficznego interfejsu użytkownika, z drugiej natomiast odbywa się w nim komunikacja pomiędzy pozostałymi modułami, pozwalając spoić program w całość. Moduł GUI jest jedynym modulem

wykorzystującym bibliotekę PyQt (poza szczerkowymi odwołaniami do niej w module wykresów).

W module TabA w zależności od trybu odpowiada za wyświetlanie zakładki umożliwiającej wyszukiwanie i przeglądanie instrumentów finansowych, oraz wyświetlanie ich wykresów z możliwością ustawienia podstawowych opcji i okresu wyświetlania wykresu. W drugim trybie moduł wyświetla wybrany wykres w trybie wykresu pojedynczego instrumentu finansowego lub w trybie porównywania kilku instrumentów finansowych na jednym wykresie. Moduł zapewnia metody do dynamicznego odświeżania wykresu w odpowiedzi na zmianę opcji w menu pod wykresem. W module TabA znajdują się również klasa i metoda do wyświetlania kilku wykresu tego samego instrumentu finansowego z różnymi wskaźnikami w nowym oknie w przypadku zaznaczenia więcej niż jednego wskaźnika w menu pod wykresem. W module Tab znajdują się wyodrębnione atrybuty wspólne dla każdego trybu wyświetlania TabA.

Moduł RSS i RSSgui odpowiadają za pobieranie i wyświetlanie dodanych przez użytkownika kanałów RSS. Domyślnie proponowane jest kilka kanałów z dziedziny finansów i rynku papierów wartościowych. Klasa RSS pobiera i parsuje wybrany kanał tworząc strukturę listy tytułów aktualnych wiadomości. Do każdego tytułu przypisana jest data utworzenia wiadomości, krótki opis i hiperłącze do pełnej treści artykułu. Moduł RSSgui tworzy widżet umożliwiający dodawanie/usuwanie kanałów oraz wyświetlanie artykułów. Moduł settings zapewnia interfejs do ustawienia sposobu wyświetlania formacji wykrytych przez moduł analizy technicznej formacji. Ustawienia zapisywane są do pliku settingsList.wsf. Moduł home wyświetla wartości i zmiany najbardziej popularnych akcji, które są na bieżąco uaktualniane podczas działania aplikacji. W module home podpięty jest i wyświetlany widżet RSSgui. Moduł analyze jest prostą przeglądarką tekstu wyświetlający opis analizy technicznej.

Moduł mainGui ładuje wszystkie moduły i elementy interfejsu, modele wszystkich instrumentów finansowych, oraz historię zamkniętych wykresów zapisanych w pliku tabHistory.py.

W Module MainWindow jest uruchamiana aplikacja.

## Interakcje między modułami

Strzałka z A do B oznacza, że moduł A wywołuje metody modułu B.

