bossaABKreator

Podręcznik Użytkownika

Wersja 3.3.2

Copyright® Remigiusz Gajda

Data aktualizacji: **02.08.2017**

Spis treści: **str.**

1. **Wprowadzenie** **2**
2. **Wymagania systemowe** **2**
3. **Instalacja** **2**
4. **Właściwości** **2**
5. **Ograniczenia** **2**
6. **Opis głównego okna programu** **3**
7. **Opis użytkowania programu** **4**
    - **Start** **5**
    - **Ustawienia ogólne** **7**
    - **Zabezpieczenia** **9**
    - **Pozycje** **11**
    - **Zlecenia** **13**
    - **Zakończ** **15**
    - **Generowanie kodu AFL** **16**
    - **Menu Plik** **19**
8. **Opis generowanych zmiennych i funkcji** **20**
9. **Końcowe uwagi** **23**

1. Wprowadzenie

Program **bossaABKreator** wspomaga tworzenie kodu AFL korzystającego z funkcji plugina **bossaAB**. Służy ułatwieniu w tworzeniu automatycznych systemów tradingowych dla Amibrokera, zwłaszcza dla początkujących użytkowników.

2. Wymagania systemowe:

- System operacyjny Windows 7 SP1/8/10.
- Wymagane uprawnienia administracyjne do instalacji.
- Amibroker© wersja min. 5.30.
- Zainstalowany Plugin **bossaAB.**

3. Instalacja:

Program powinien zostać skopiowany do oddzielnego katalogu. Do tego celu mogą być potrzebne uprawnienia administratora systemu Windows.
Ikonę uruchamiającą program można dodać do menu systemowego naciskając odpowiedni przycisk w okienku **Opcje->Ustawienia**.

4. Właściwości

Program **bossaABKreator** obudowuje wczytany pierwotny kod AFL własnymi funkcjami i definicjami zmiennych tak, by w rezultacie było możliwe automatyczne zawieranie transakcji z poziomu skryptu AFL w Amibrokerze z wykorzystaniem plugina **bossaAB**.

Program koloruje składnię wczytanego i wygenerowanego kodu. Sprawdza także występowanie definicji sygnałów **BUY**/**SELL**.

Proces budowania kodu podzielony jest na etapy (wizard) co daje większą przejrzystość i ułatwia korzystanie z programu. W każdym momencie można zapisać edytowany projekt by powrócić do niego w innej chwili.

5. Ograniczenia:

Program **bossaABKreator** w żaden sposób nie sprawdza poprawności wygenerowanego kodu.

Z powodu specyfiki działania Amibrokera **bossaABKreator** nie gwarantuje bezbłędnego działania kodu wynikowego we wszystkich przypadkach. Dotyczy to szczególnie skomplikowanych edytowanych kodów AFL.

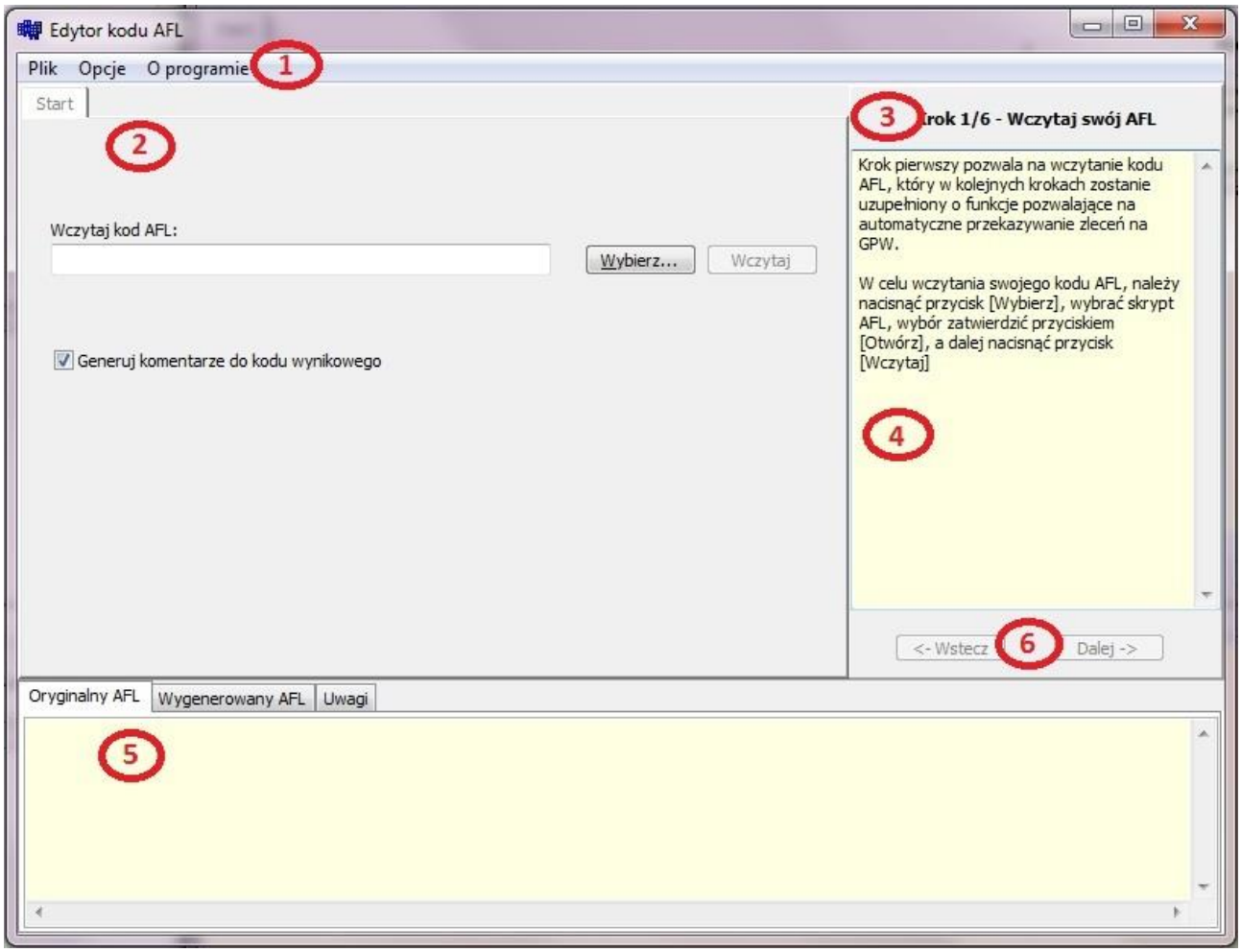
Wszelkie odwołania do zewnętrznych plików i wielokrotne zagnieżdżanie fragmentów z definicją sygnałów **BUY**/**SELL** może spowodować wygenerowanie błędnego kodu.
Do użytkownika programu należy ostateczna weryfikacja kodu i jego przetestowanie.

Program nie obsługuje sygnałów **SHORT** i **COVER**.

Konstrukcja kodu wynikowego zakłada, że transakcje odbywają się na instrumentach na których nie można otworzyć pozycji krótkich. Jeżeli użytkownik chce używać kodu wygenerowanego przez kreator np. do handlu kontraktami, to powinien zmodyfikować sekcję zlecenia sprzedaży (uwagi w punkcie **8 Opis generowanych zmiennych i funkcji**).

6. Opis głównego okna programu

Po uruchomieniu programu bossaABKreator wyświetlane jest następujące okno:

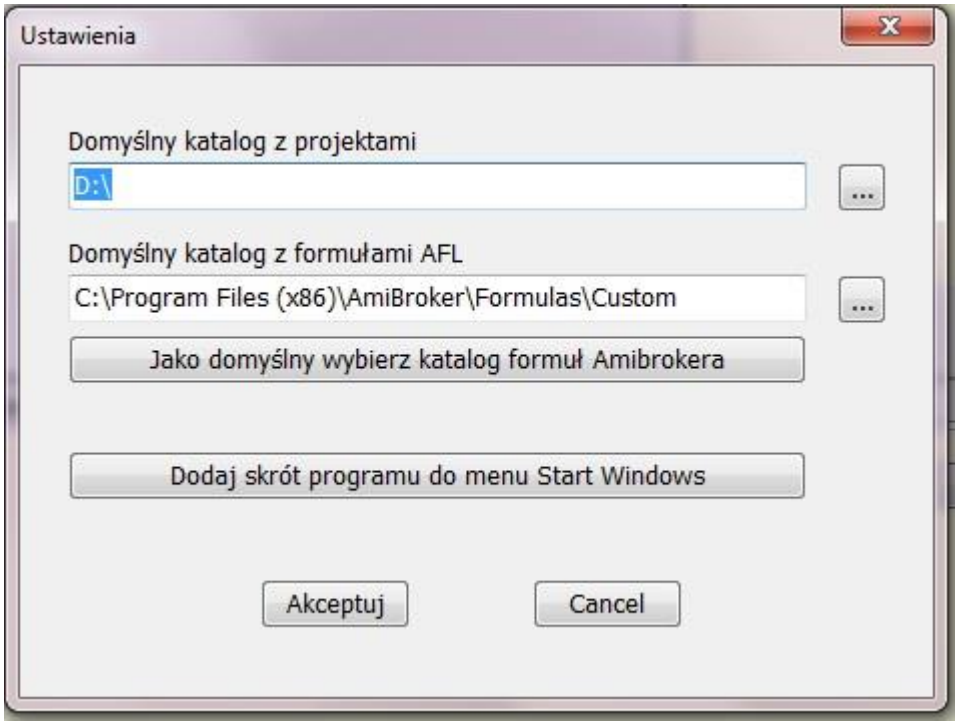


Okno składa się z kilku zasadniczych elementów zaznaczonych w czerwonymi obwódkami:

1. Menu programu.
2. Pole z otwieranymi kolejnymi zakładkami z opcjami do ustawienia.
3. Nagłówek z numerem kolejnego kroku i krótkim opisem kroku.
4. Pole z informacjami na temat opcji w danej zakładce.
5. Zakładki z treścią edytowanego kodu, wygenerowanego kodu i uwagami
6. Przyciski nawigacyjne pomiędzy kolejnymi krokami.

7. Opis użytkowania programu

Pracę z programem należy rozpocząć od uruchomienia opcji **Ustawienia** (menu **Opcje->Ustawienia**):



Naciskając przycisk "**…**" wywołuje się okno dialogowe wyboru katalogu „**Otwórz katalog**".

Okienko **Otwórz katalog** (nie pokazane) umożliwia nawigację po dostępnych katalogach. Wybór należy zatwierdzić przyciskiem "**Otwórz**" - nazwa katalogu pojawi się w odpowiednim polu z lewej strony przycisku.

Jeżeli użytkownik chce korzystać z formuł znajdujących się w podkatalogu Formulas\Custom Amibrokera, to wystarczy nacisnąć przycisk „**Jako domyślny wybierz…**". Program samodzielnie odnajdzie odpowiednią ścieżkę dostępu. Należy jednak pamiętać, że w takim przypadku może dojść do nieumyślnego nadpisania oryginalnego kodu AFL przez kod wynikowy i w rezultacie, w pewnych przypadkach, utraty części oryginalnego kodu!

Zaleca się trzymanie oryginalnych formuł AFL w osobnym katalogu.

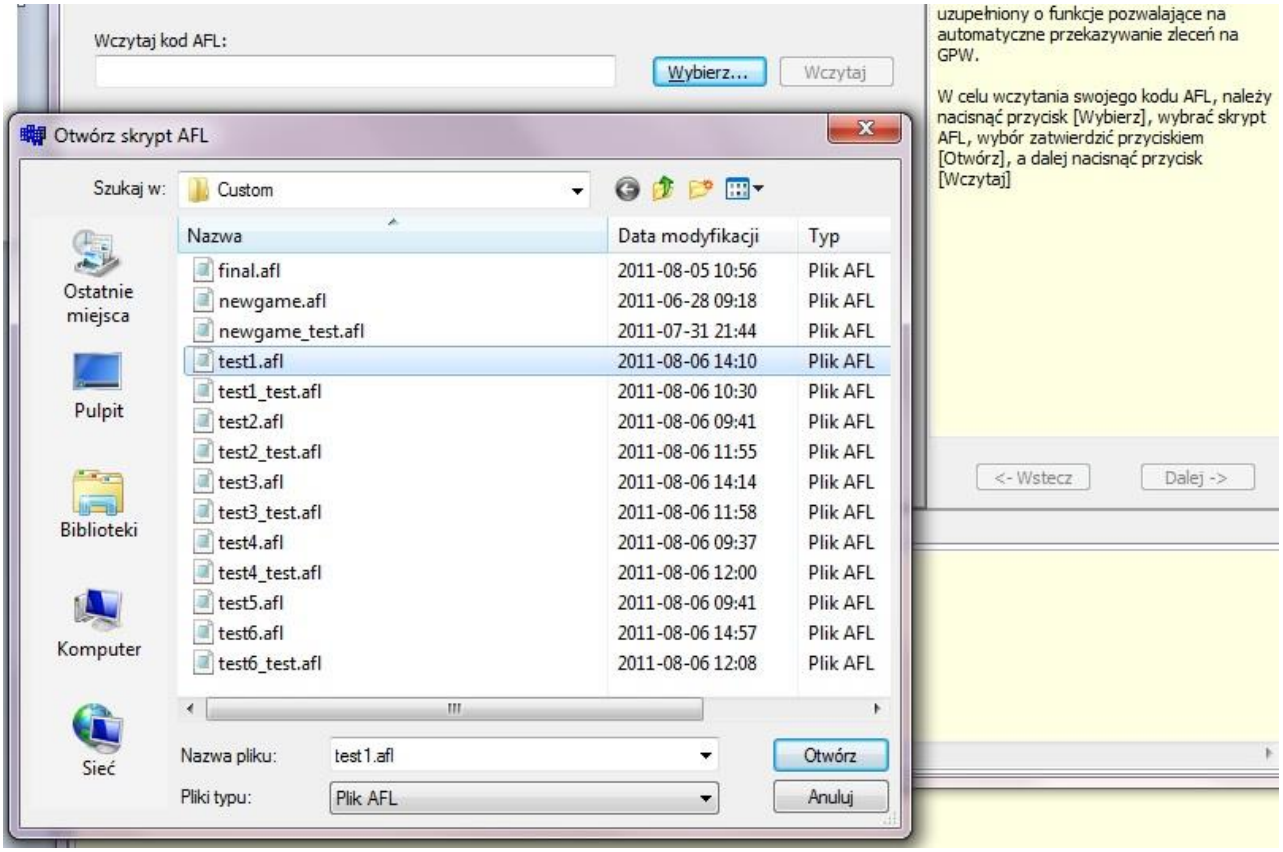
Uruchomienie przycisku „**Dodaj skrót programu do menu Start Windows**" powoduje utworzenie w menu Start (Wszystkie programy) skrótu do aplikacji **bossaABKreator**.

Uruchomiony program gotowy jest do obsługi nowego projektu. O koniecznych akcjach jakie musi wykonać użytkownik informuje okienko z prawej strony.

W kolejnych zakładkach pojawiają się różne opcje. Większość z nich jest nieaktywna i dopiero zaznaczanie opcji włącza następne. Taka konstrukcja uniemożliwia pominięcie ważnych ustawień przez użytkownika.

- **Start**

W pierwszym kroku należy wybrać kod AFL, który będzie w dalszym ciągu przetwarzany. W tym celu należy kliknąć przycisk ”**Wybierz**”, pojawi się okno wyboru skryptu AFL.



Na obrazku widać, że wybrano plik **test1.afl**. Wybór należy zaakceptować przyciskiem ”**Otwórz**”. Nazwa skryptu AFL pojawi się w polu ”**Wczytaj kod AFL**”. Nazwę tę można też wprowadzić ręcznie. Program gotowy jest do wczytania kodu. Teraz należy nacisnąć ”**Wczytaj**”.

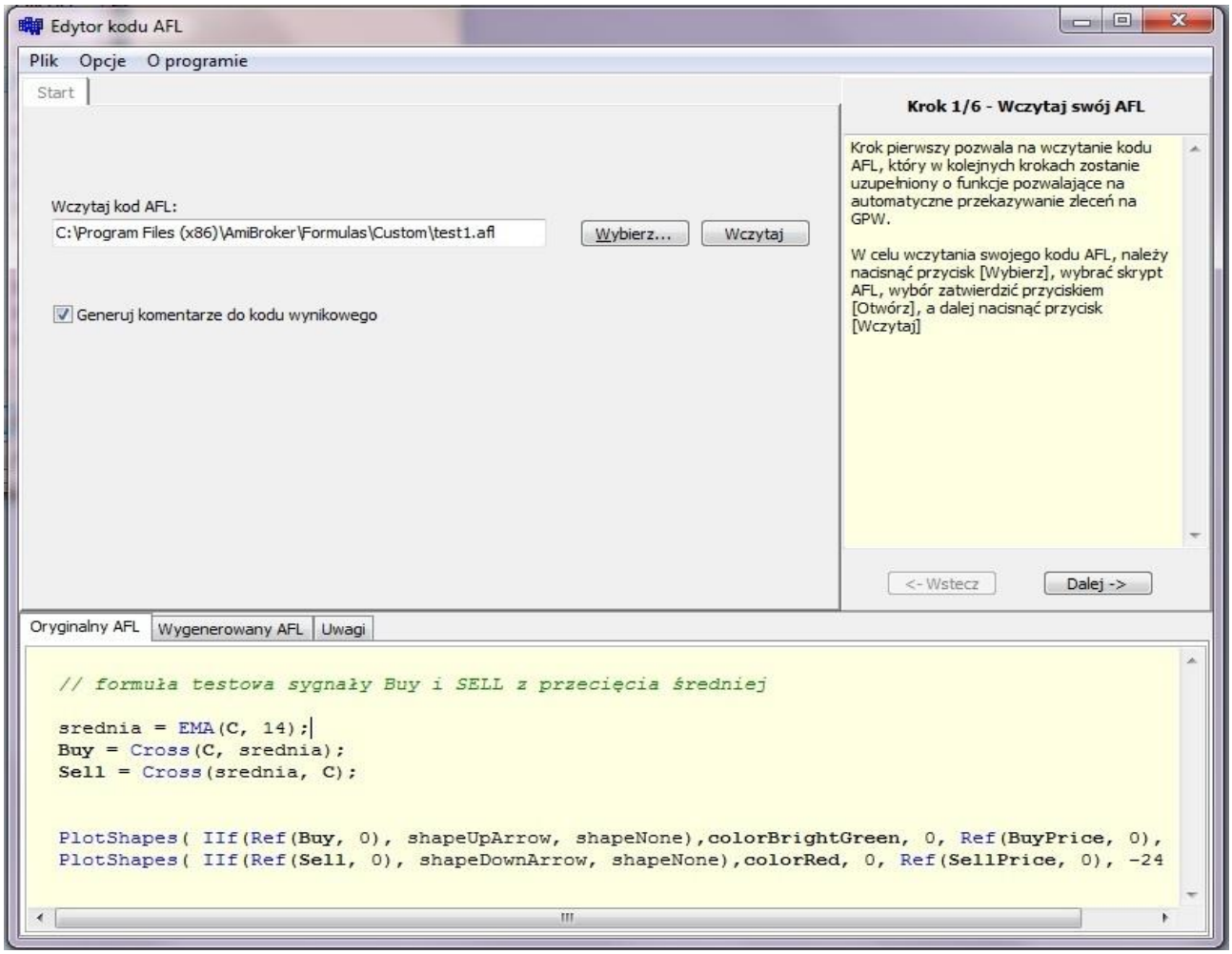
Po wczytaniu kodu jest on sprawdzany pod kątem występowania sygnałów BUY/SELL oraz funkcji plugina **bossaAB**.

Jeżeli brakuje sygnałów BUY/SELL praca programu jest przerywana, bo takiego kodu program nie obsługuje.

Także w przypadku wykrycia odwołań do funkcji plugina praca jest przerywana ponieważ oznacza to, że kod już został opracowany pod kątem współpracy z pluginem.

Stosowne ostrzeżenia o błędach pojawiają się w zakładce **Uwagi** w dolnej części okna programu.

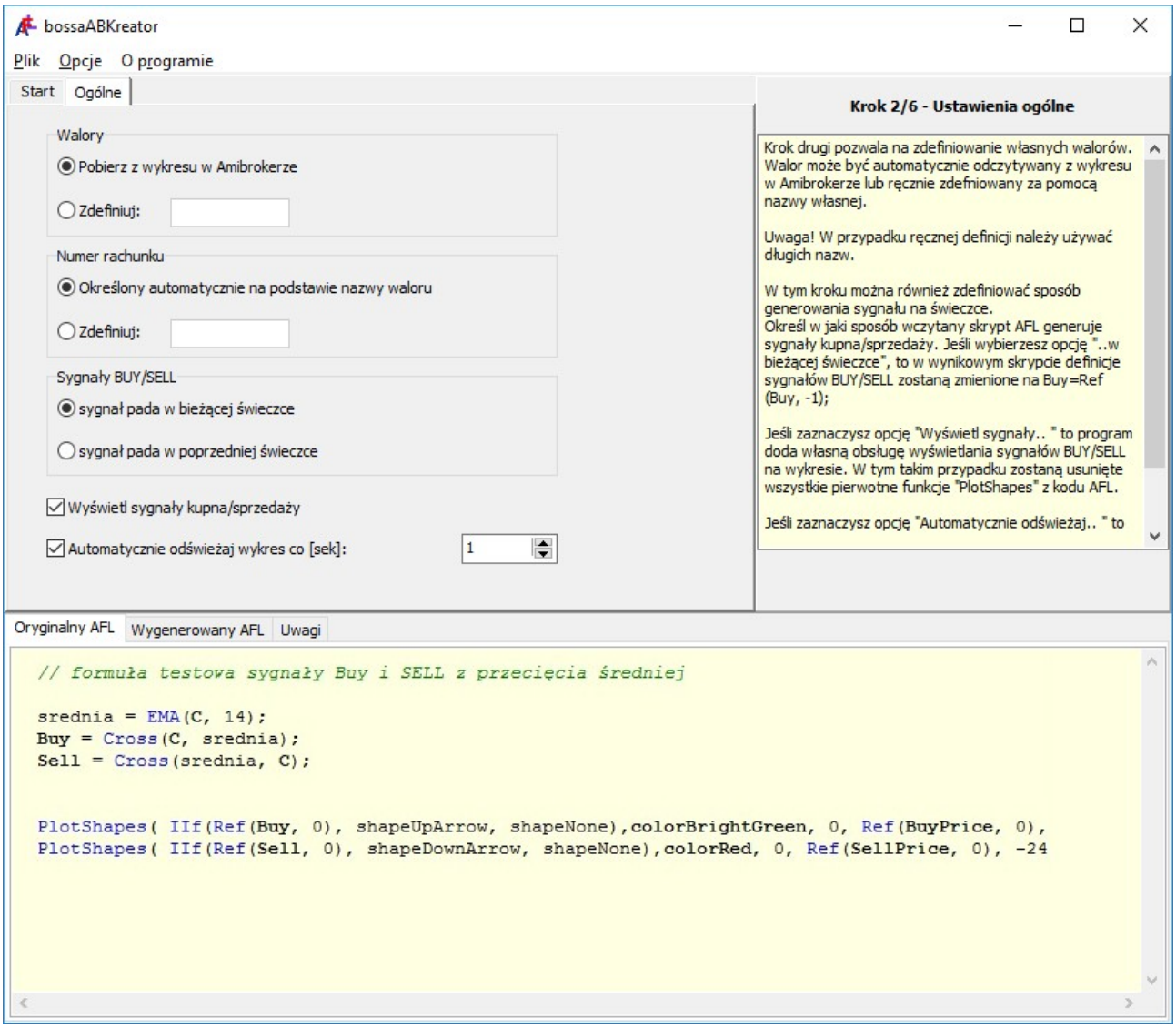
Wczytany kod wyświetlany jest w zakładce **Oryginalny AFL**. Kod jest kolorowany podobnie jak w edytorze Amibrokera. Na poniższym obrazku widoczny jest wczytany fragment kodu AFL.



Jeżeli wczytany kod nie zawiera błędów z punktu widzenia **bossaABKreatora** to przycisk **Dalej** staje się aktywny i należy go kliknąć aby przejść do następnego etapu – ustawień ogólnych.

Zaznaczenie opcji **Generuj komentarze do kodu wynikowego** wzbogaca wygenerowany kod AFL o opis zmiennych i funkcji dodawanych przez **bossaABKreator**.

- **Ustawienia ogólne**



W tym kroku użytkownik decyduje o podstawowych parametrach wynikowego kodu. Najpierw należy wybrać walor na jakim będzie się odbywał automatyczny handel. W ramce **Walory** należy zaznaczyć odpowiednią opcję:

- **Pobierz z wykresu w Amibrokerze** – jest to ustawienie pozwalające na używanie danego kodu dla różnych walorów. Może jednak powodować niezamierzone transakcje jeżeli użytkownik pomyłkowo uruchomi kod na więcej niż jednym wykresie.
  W tym przypadku kod posługuje się walorem o nazwie pobranym z wykresu na którym jest uruchomiony.
- **Zdefiniuj** – należy wpisać nazwę waloru. Program nie sprawdza zgodności nazwy z dopuszczalnymi przez GPW (oraz NOL). Ta opcja daje większą gwarancję niezawodności.

Po zdefiniowaniu nazwy waloru aktywna staje się ramka **Numer rachunku**:

- **Określony automatycznie na podstawie nazwy waloru** – plugin na bazie listy przedrostków automatycznie przyporządkowuje odpowiedni rachunek do danego waloru. Opcja ta działa poprawnie przy założeniu, że użytkownik ma dostęp tylko do

jednego rachunku danego rodzaju (tzn np. ma tylko jeden rachunek kontraktowy, jeden akcyjny itp.)

- **Zdefiniuj** – należy wpisać numer rachunku w formacie typu 00-xx-xxxxxx. Opcja ta wymusza użycie konkretnego rachunku. Użytkownik musi jednak być pewien, że ma dostęp do danego rachunku i można na nim zawierać transakcje dla danego waloru.

Sygnały BUY/SELL – program **bossaABKreator** nie jest w stanie określić w ogólnym przypadku w którym momencie powstaje sygnał. Wiele systemów transakcyjnych działa tak, że od chwilowej ceny w bieżącej świeczce zależy czy sygnał powstaje czy nie. Tak może być np. przy prostym sprawdzaniu czy **Close** przecina średnią z jakiegoś okresu. W takim przypadku mogą być generowane fałszywe (bo nie ustalone jeszcze) sygnały transakcyjne.
Dlatego ważne jest aby system transakcyjny posługiwał się sygnałami w pełni ustabilizowanymi. Do użytkownika należy analiza oryginalnego kodu i określenie w jaki sposób powstaje sygnał BUY/SELL:

- **sygnał pada w bieżącej świeczce** – najczęstszy przypadek. W kodzie zostanie uwzględniony poprzez pobieranie wartości BUY/SELL z przedostatniej świeczki za pomocą funkcji Ref(Buy, -1)/Ref(Sell, -1). Oznacza to, że zlecenie zostanie wysłane najwcześniej na otwarciu następnej świeczki po bieżącej!
- **sygnał pada w poprzedniej lub wcześniejszej świeczce** – kod nie ingeruje w definicje sygnałów BUY/SELL, bo w tym przypadku sygnał jest ustabilizowany. Należy jednak pamiętać, że jeżeli sygnał padł np. 3 świeczki wcześniej, to musi być przeniesiony na bieżącą świeczkę (np. funkcją **Hold** Amibrokera)

Przy wyborze opcji nie należy sugerować się ich nazwami, bo istotniejsze jest czy sygnał jest stabilny w bieżącej czy poprzedniej świeczce!
Przykładem systemu, w którym sygnał pada w bieżącej świeczce i nie ulega już zmianie jest powszechnie znany system Pawła Rejczaka. W tym przypadku należy wybrać opcję „sygnał pada w poprzedniej świeczce”!

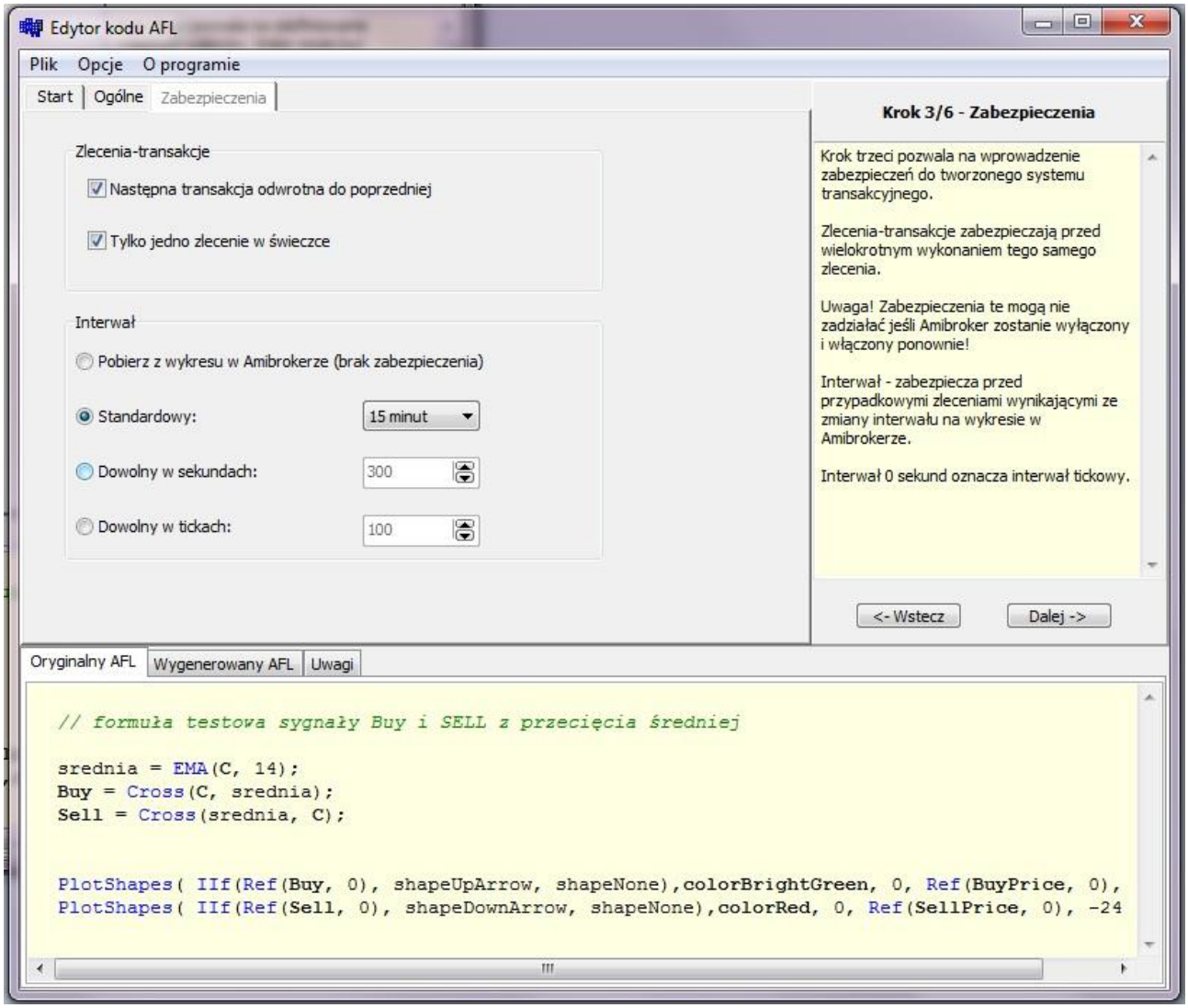
Wyświetl sygnały kupna/sprzedaży – pole domyślnie zaznaczone. Zaznaczenie tej opcji usuwa wszelkie wywołania funkcji **PlotShapes** i wstawia własną obsługę wyświetlania sygnałów BUY/SELL. Należy pamiętać, że **PlotShapes** może służyć także wyświetlaniu innych rzeczy, np. przebiegu średnich itp. W takim przypadku można do wynikowego kodu skopiować potrzebne funkcje PlotShapes z oryginalnego kodu (w edytorze Amibrokera).

Automatycznie odświeżaj wykres co [sek] – opcja wstawia do kodu AFL funkcję:

RequestTimedRefresh(interval,onlyvisible); z parametrem onlyvisible = false,

co pozwala na automatyczne odświeżanie okna wykresu wg. określonego interwału.
W praktyce, funkcja pozwala na odświeżanie wykresów(danych) w nieaktywnych górnych zakładkach w AmiBrokerze, a tym samym stosowanie automatów nie tylko dla aktywnej zakładki.

- **Zabezpieczenia**



Zakładka służy do ustawienia zabezpieczeń zwiększających bezpieczeństwo tradingu.

Ramka **Zlecenia-transakcje**:

- **Następna transakcja odwrotna do poprzedniej** – domyślnie zaznaczona. Jeżeli jest zaznaczona, to dodaje do kodu wywołanie funkcji plugina ustawiającej sprawdzanie czy poprzednia transakcja na danym walorze jest odwrotna do poprzedniego zlecenia na tym walorze. Dodatkowo w kodzie AFL umieszczane są definicje dodatkowych zmiennych statycznych, które także kontrolują naprzemienność transakcji.
- **Tylko jedno zlecenie w świeczce** – zaznaczenie tej opcji nie pozwala na generowanie wielu zleceń w tej samej świeczce.

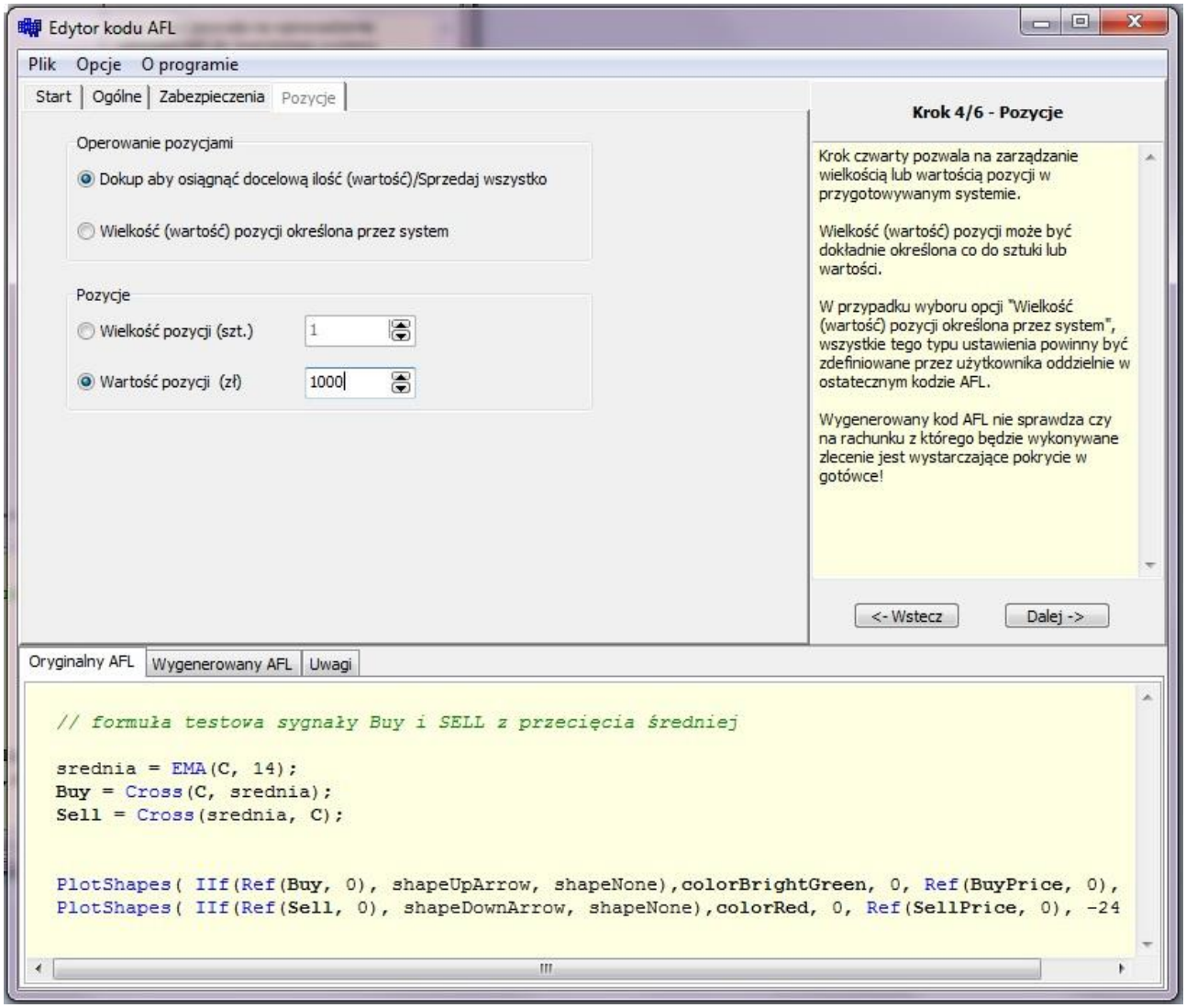
Zabezpieczenia powyższe mogą nie zadziałać jeżeli w trakcie pracy nastąpi wyłączenie i ponowne włączenie Amibrokera i systemów transakcyjnych.

Ramka **Interwał** pozwala określić interwał na jakim ma pracować system transakcyjny. Wynikowy kod nie wymusza ustawionego interwału. Bieżący interwał jest jedynie porównywany z zadeklarowanym i w przypadku niezgodności **pluginAB** informuje o błędzie i nie dopuszcza do złożenia zlecenia:

- **Pobierz z wykresu w Amibrokerze (brak zabezpiecznia)** – kod pracuje z aktualnie wybranym na wykresie interwałem. Każda jego zmiana może prowadzić do fałszywych sygnałów lub pominięcia prawidłowych.
- **Standardowy** – z rozwijalnej listy można wybrać predefiniowane interwały Amibrokera (jak w bazie danych).
- **Dowolny w sekundach** – w polu obok można ustawić dowolny interwał wyrażony w sekundach. Interwał 0 sekund oznacza interwał 1-tikowy.
- **Dowolny w tickach** – w polu obok można ustawić dowolny interwał tikowy.

Powyższe trzy opcje są preferowane ponieważ zapewniają wyższy poziom bezpieczeństwa. Chronią przed fałszywymi sygnałami chociaż nadal zmiana interwału na wykresie może powodować pominięcie powstających sygnałów. Trzeba też pamiętać, że na ogół systemy transakcyjne dostosowane są do konkretnych interwałów.

- **Pozycje**



W tym miejscu użytkownik decyduje w jaki sposób system określa wielkość transakcji:

Ramka **Operowanie transakcjami**:

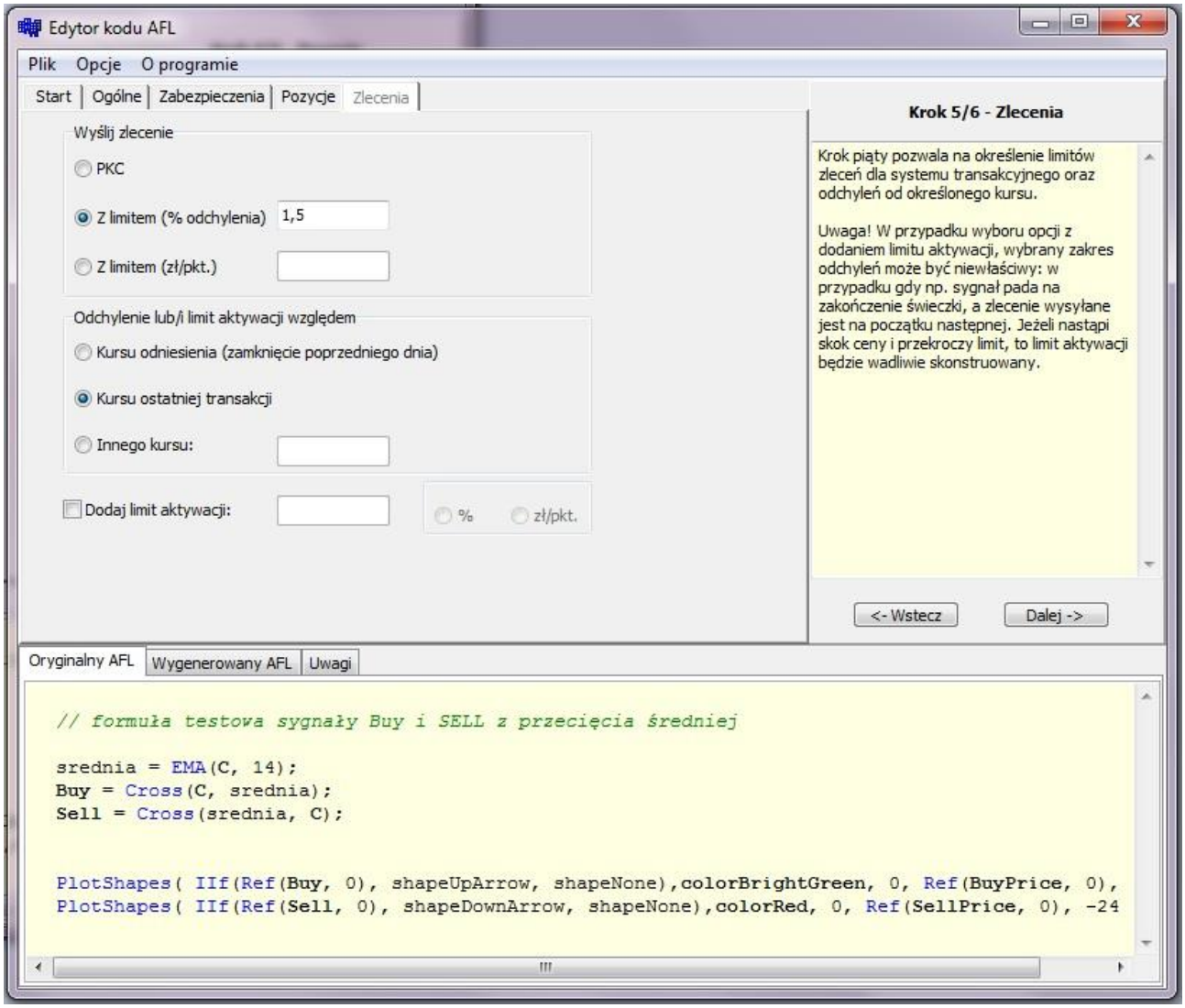
- **Dokup aby osiągnąć docelową ilość(wartość)/Sprzedaj wszystko** – wielkość/wartość pozycji określona jest z góry przez użytkownika. Kod wynikowy nie sprawdza czy na rachunku istniały wcześniej jakieś walory. Dokupuje zdefiniowaną ilość a sprzedaje wszystko. Może więc się zdarzyć, że na rachunku pojawi się więcej danych papierów niż określono w kodzie (jeśli już wcześniej tam były) a sprzedane zostaną wszystkie, a więc więcej niż zostało zakupione przez system transakcyjny.
- **Wielkość (wartość) pozycji określona przez system** – do kodu zostaje dodane tylko definicje odpowiednich zmiennych i komunikat o konieczności modyfikacji kodu tak aby uwzględniał ustawienia oryginalnego kodu AFL (wyskakujące okienko po użyciu w Amibrokerze).

Jeżeli wybrano opcję „**Dokup …**”, to ramka **Pozycje** staje się aktywna i należy określić wielkość docelowego kupna:

- **Wielkość pozycji (szt.)** – ilość papierów do kupna określana jest bezpośrednio w sztukach. Ilość akcji należy podać w okienku obok.
- **Wartość pozycji (zł)** – kod wyliczy ilość akcji jakie należy kupić na podstawie podanej kwoty w okienku obok.

Wynikowy kod nie sprawdza czy jest wystarczające pokrycie gotówkowe na zakup akcji!

- **Zlecenia**



W tej zakładce należy określić w jaki sposób mają być realizowane zlecenia.
Ramka **Wyślij zlecenie** określa limit ceny w zleceniu:

- **PKC** – zlecenie będzie wystawione jako PKC (po każdej cenie). Najpewniejszy sposób realizacji zlecenia.
- **Z limitem (% odchylenia)** – zlecenie zostanie wystawione z limitem ceny obliczonym względem ceny bazowej (wyznaczanej dalej) poprzez dodanie/odjęcie od niej określonego w polu obok odchylenia procentowego.

Przykład: limit odchylenia ustalono na 1,5% (jak w oknie wyżej), a za cenę bazową wybrano cenę ostatniej transakcji.
W praktyce oznacza to, że limit ceny w zleceniu kupna będzie obliczony jako:
limit= Close + Close*0.015.
Dla sprzedaży będzie to:
limit= Close-Close*0.015.
Close jest ceną ostatniej transkacji (w Amibrokerze będzie to wartość LastValue(Close))

- **Z limitem (zł/pkt.)** – limit ceny podany jest bezpośrednio w złotych lub punktach (dla kontraktów)

Ramka **Odchylenie lub/i limit aktywacji względem** określa względem jakiej wartości liczony jest limit ceny lub/i limit aktywacji:

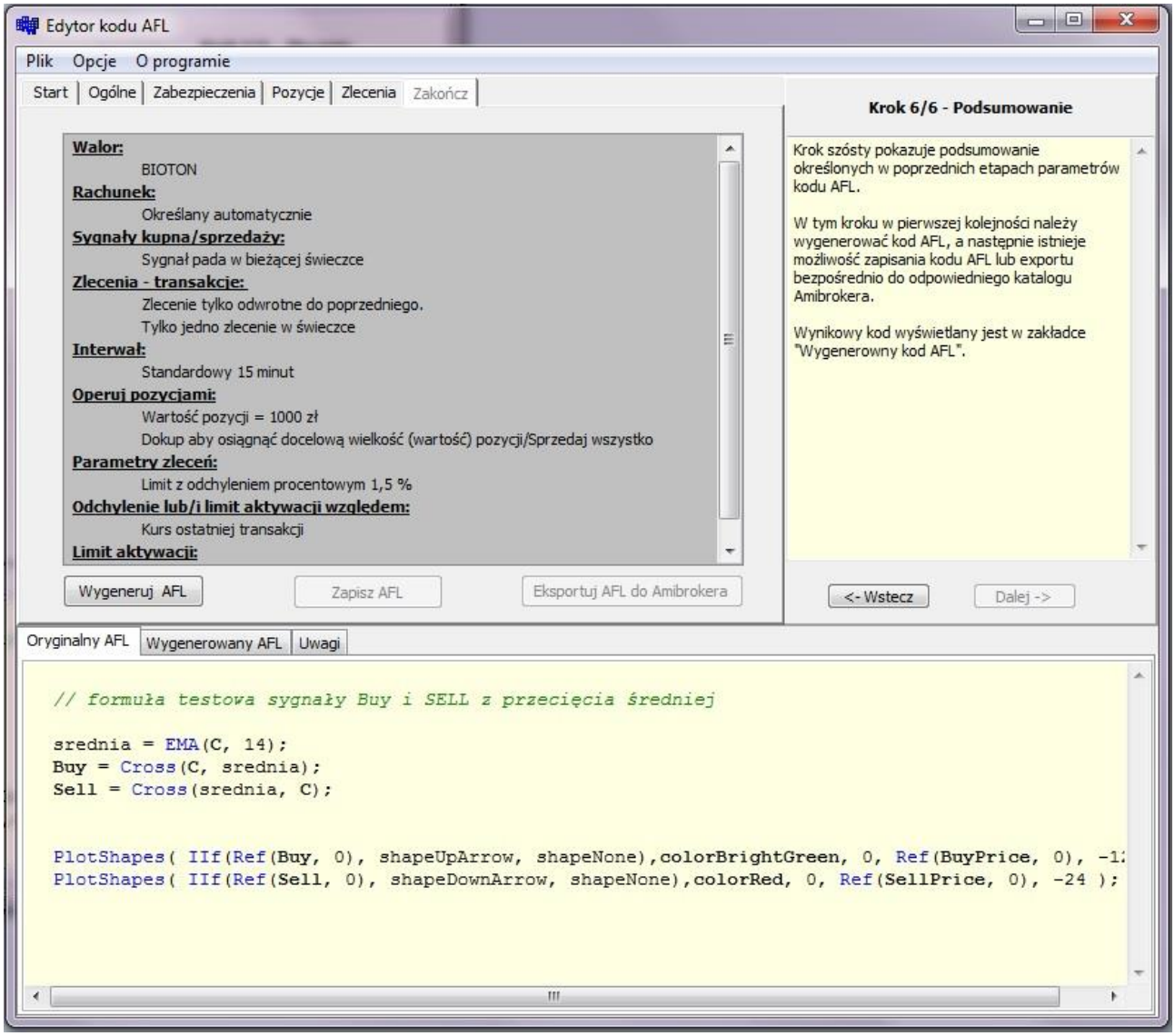
- **Kursu odniesienia (zamknięcie poprzedniego dnia)** – wynikowy kod pobierze z wykresu cenę zamknięcia z poprzedniego dnia. Użyte zostaną w tym celu funkcje zmieniające chwilowo interwał (`TimeFrameSet(inDaily)` i `TimeFrameRestore()`)
- **Kursu ostatniej transakcji** – jako cenę bazową kod pobierze ostatnią wartość wektora Close dla danego waloru.
- **Innego kursu** – w polu obok należy wpisać wartość kursu według którego będą liczone limity w zleceniach

Jeżeli transakcje mają być wyzwalane zleceniami z limitem aktywacji, to należy zaznaczyć opcję **Dodaj limit aktywacji**, w polu obok należy wpisać odchylenie kwotowe (złote lub punktowe) lub procentowe od ceny bazowej. Rodzaj odchylenia określa ramka obok:

- **%** - limit aktywacji to cena bazowa +/- procentowa część tej ceny
- **zł/pkt**. – limit aktywacji to cena bazowa +/- wpisana kwota

Zasady obliczania limitu aktywacji są identyczne jak dla zwykłego limitu ceny. Należy jednak pamiętać, że **bossaABKreator** w żaden sposób nie sprawdza poprawności zadeklarowanych a w trakcie pracy systemu skoki ceny mogą powodować, że formalnie poprawnie skonstruowany limit aktywacji staje się nieaktualny w momencie składania zlecenia!

- **Zakończ**



Ostatni krok wizarda **bossaABKreator** to okienko podsumowujące ustawienia i dające możliwość wygenerowania kodu wynikowego.

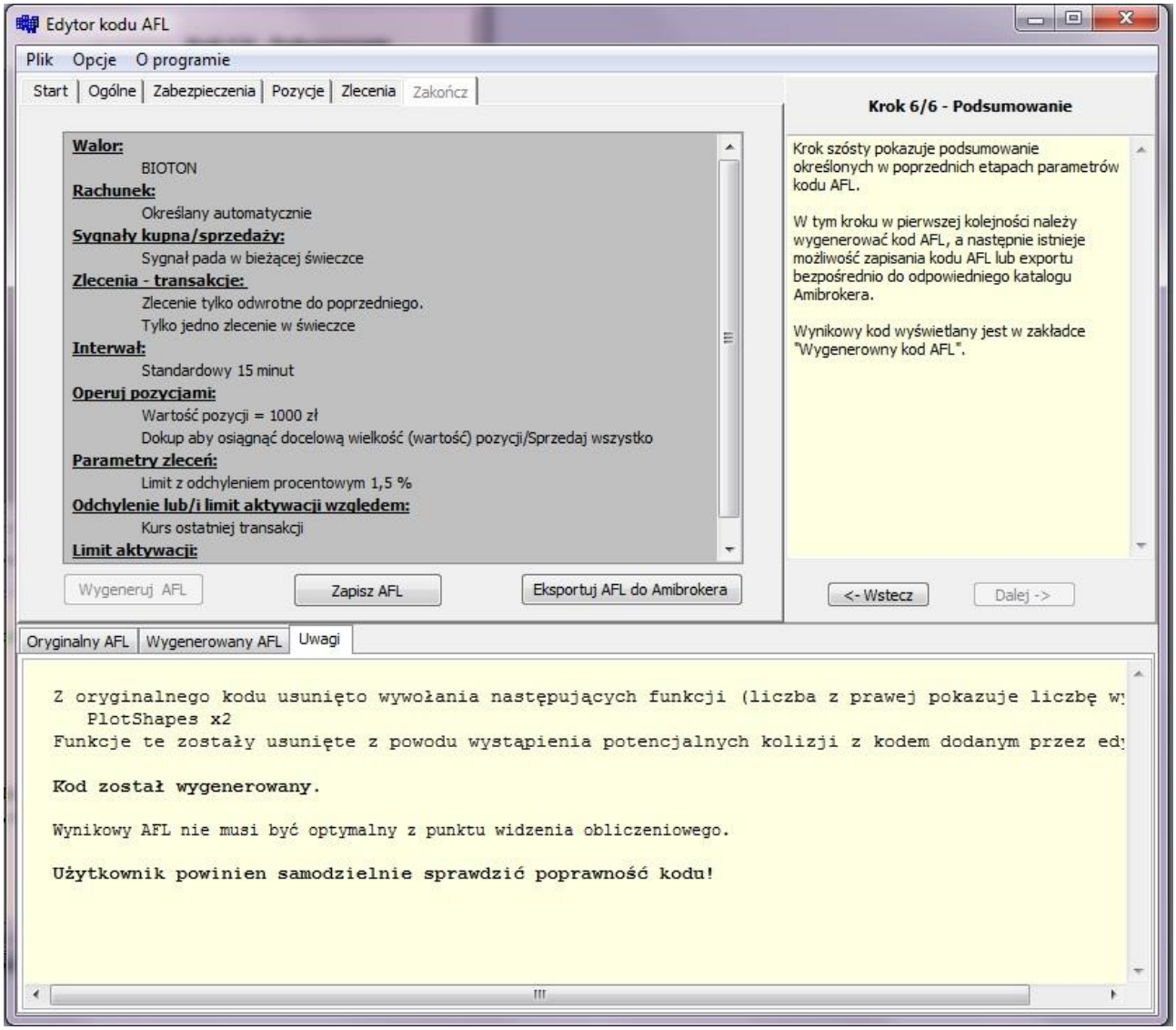
W zakładce **Zakończ** wypisane są wszystkie wybrane przez użytkownika opcje. Użytkownik powinien uważnie sprawdzić podsumowanie.

Jeżeli zachodzi potrzeba zmiany opcji, to można posłużyć się przyciskiem **Wstecz** lub bezpośrednio wybrać odpowiednią zakładkę aby zmienić odpowiednie opcje.

Okienko **Zakończ** jest dobrym momentem na zapisanie projektu: menu **Plik->Zapisz projekt jako**… Dzięki temu można łatwo korygować poszczególne parametry w późniejszym terminie bez konieczności przechodzenia przez cały cykl kreatora. Patrz rozdział **Menu Plik**.

- **Generowanie kodu AFL**

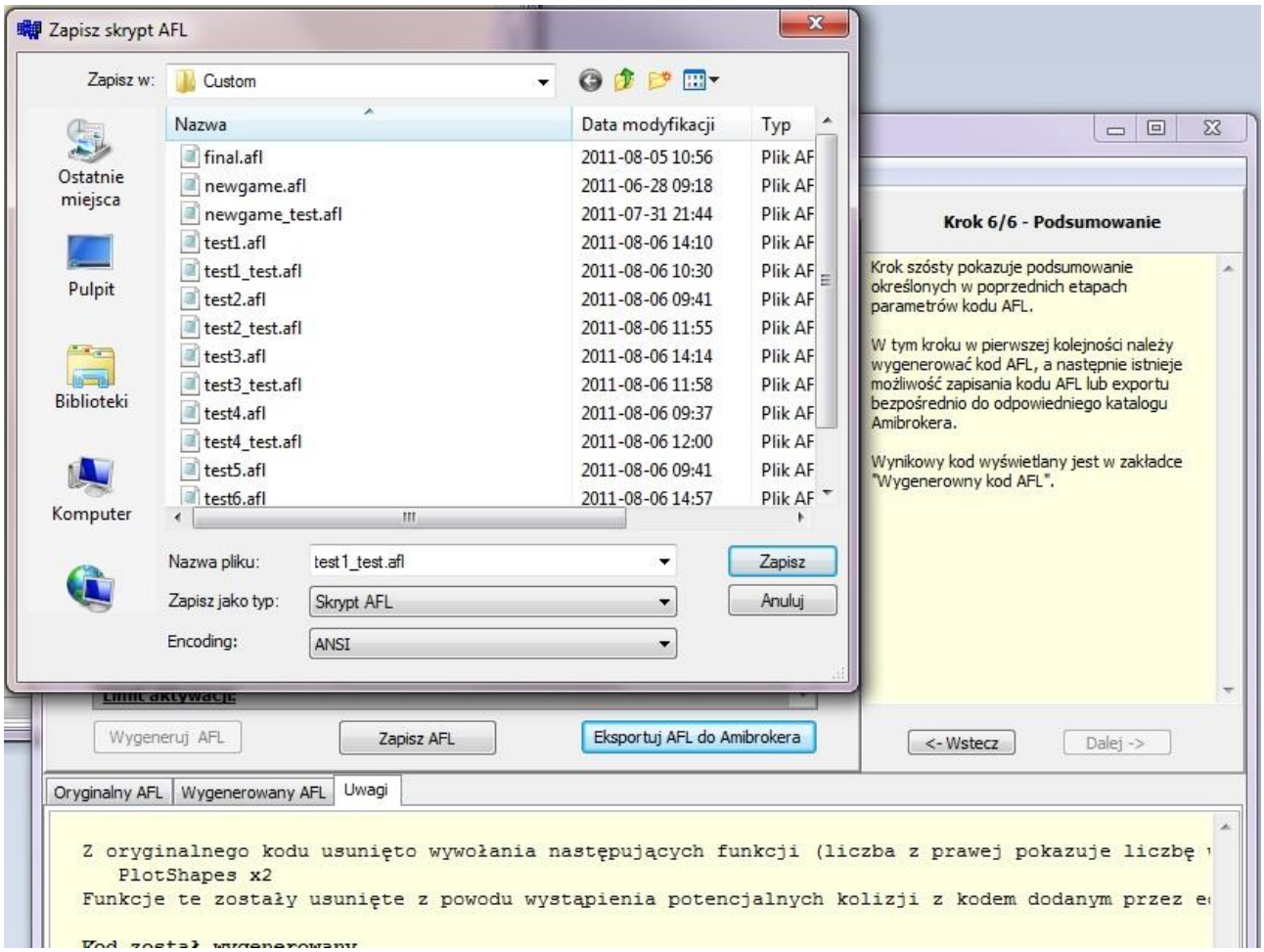
Aby wygenerować kod końcowy należy kliknąć przycisk **Wygeneruj AFL**. Po chwili pokaże się zakładka **Uwagi** na dole ekranu z uwagami na temat wygenerowanego kodu:



Jak widać na powyższym ekranie, w **Uwagach** umieszczono informację o usunięciu dwoch wywołań funkcji **PlotShapes** ponieważ zaznaczono opcję obsługi wyświetlania sygnałów **BUY**/**SELL**.

Po wygenerowaniu kodu można go zapisać w dowolnym folderze naciskając **Zapisz AFL** (otwierany jest domyślny katalog dla skryptów AFL zdefiniowany w **Ustawieniach**).

Można też bezpośrednio zapisać go w domyślnym katalogu dla formuł AFL Amibrokera (zdefiniowany w **Ustawieniach**, jest to zazwyczaj podkatalog **Formulas\Custom** katalogu Amibrokera):

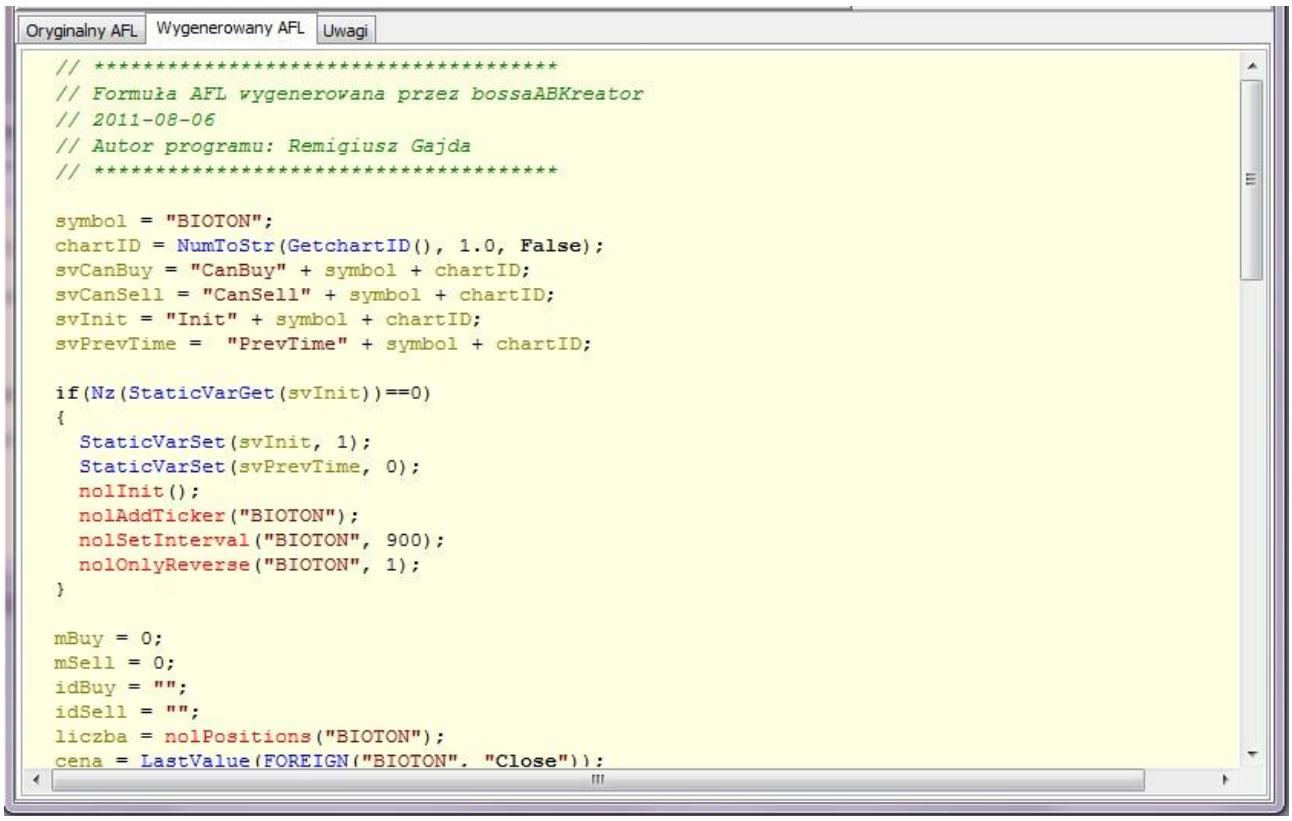


Należy unikać nadpisywania oryginalnego kodu przez wygenerowany, ponieważ można utracić część informacji z pierwotnego pliku. Poza tym późniejsze „czyszczenie” wynikowego kodu z dodatków kreatora może prowadzić do błędów w kodzie.

W powyższym przykładzie obrabiany był skrypt **test1.afl** a kod wynikowy zapisywany jest pod nazwą **test1_test.afl**.

Przy próbie nadpisania istniejącego pliku wyświetlane jest okienko żądające potwierdzenia operacji.

Po wygenerowaniu kodu można go obejrzeć w zakładce **Wygenerowany AFL**:

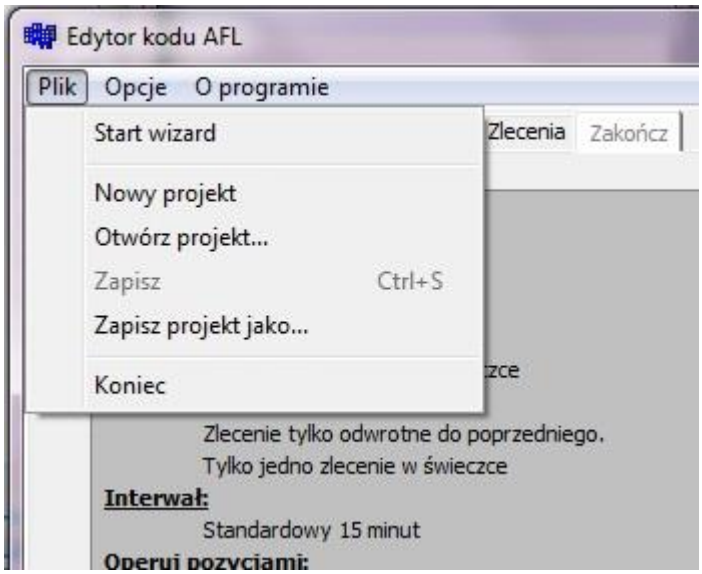


Funkcje plugina zaznaczone są kolorem czerwonym. Pozostałe zmienne i funkcje dodane przez kreator mają kolor oliwkowy. Stałe łańcuchowe jak np. powyżej **BIOTON** mają kolor jasno brązowy.

Jeżeli użytkownik uzna, że należy zmienić parametry w kreatorze, to każda taka zmiana spowoduje usunięcie wygenerowanego kodu i proces jego generowania trzeba będzie powtórzyć.

Menu Plik

W menu **Plik** zawiera następujące polecenia:



Start wizard – ustawia program w stanie początkowym, większość opcji jest nieaktywna i należy wykonać wszystkie kolejne kroki kreatora aby otrzymać kod wynikowy. Polecenie to nie usuwa wczytanego kodu AFL.

Nowy projekt – usuwa wszystkie ewentualne ustawienia z poprzedniego projektu, usuwa wcześniej wczytany kod i kod wynikowy, ustawia program w gotowości do rozpoczęcia pracy z nowym kodem AFL.

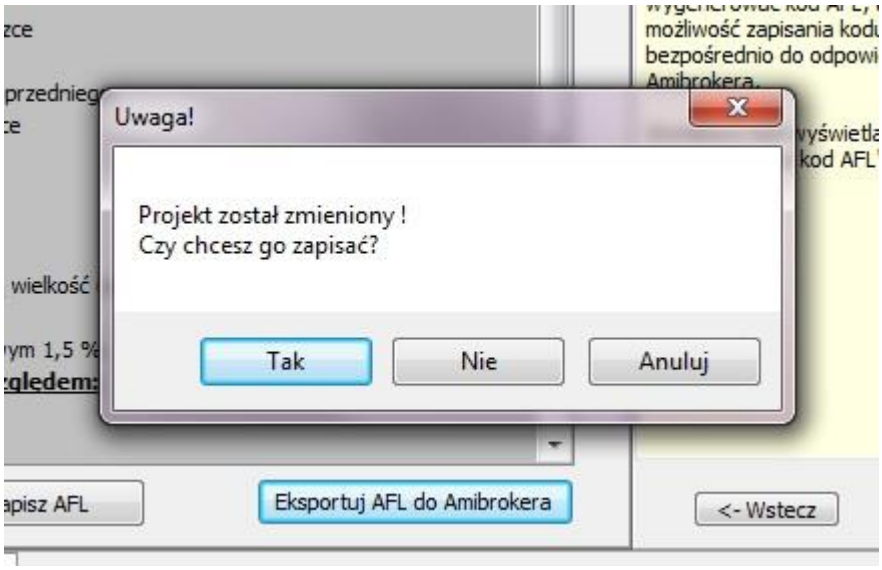
Otwórz projekt… - wyświetla okno dialogowe wczytywania projektów. Projekty programu **bossaABKreator** mają rozszerzenie **AFP**.

Zapisz – zapisuje wcześniej zapisany projekt, który uległ zmianie w wyniku edycji parametrów.

Zapisz projekt jako… - wyświetla okno dialogowe zapisywania projektu, pliki projektu otrzymują rozszerzenie AFP. W okienku dialogowym nie trzeba tego rozszerzenia dopisywać.

Koniec – kończy pracę z **bossaABKreator**

Jeżeli użytkownik dokonał jakichkolwiek zmian w projekcie i nie zapisał go a chce zamknąć program, to pojawi się ostrzeżenie, że projekt nie został zapisany i będzie możliwość wykonania tej czynności:


możliwość zapisania kodu
bezpośrednio do odpowi
Amibrokera.
Uwaga!
Projekt został zmieniony !
Czy chcesz go zapisać?
Tak
Nie
Anuluj
Eksportuj AFL do Amibrokera
<- Wstecz

8. Opis generowanych zmiennych i funkcji

bossaABKreator umieszcza w wynikowym kodzie własne zmienne i funkcje w zależności od wybranych opcji. Poniżej opisano najważniejsze aspekty dotyczące zmiennych i funkcji.

Lista nazw z których korzysta **bossaABKreator**:

- **symbol** - nazwa waloru do tradingu. Jeżeli jest to pusty łańcuch .znakowy (""), to nazwa waloru pobierana jest z wykresu funkcją **Name**().
- **chartID** - identyfikator wykresu pobierany funkcją **GetChartID**.
- **svInit** - zmienna statyczna informująca czy plugin został zainicjalizowany dla danego wykresu.
- **svCanBuy** - zmienna statyczna zaznaczająca czy można wykonywać zlecenia kupna. Ma znaczenie tylko gdy wybrano opcję „**Następna transakcja odwrotna do poprzedniej**"
- **svCanSell** - zmienna statyczna zaznaczająca czy można wykonywać zlecenia sprzedaży
- **svPrevTime** - zmienna statyczna przechowująca ostatni odczytany czas w funkcji **isNewBar**.
- **isNewBar** - funkcja sprawdzająca czy powstała nowa świeczka na wykresie. Funkcja pojawia się tylko jeżeli wybrano opcję „**Tylko jedno zlecenie w świeczce**"
- **newBar** - zmienna przechowująca wynik działania funkcji **isNewBar**
- **mBuy** - zmienna przechowująca bieżący stan sygnału kupna.
- **mSell** - zmienna przechowująca bieżący stan sygnału sprzedaży
- **idBuy** - zmienna zawierająca identyfikator zlecenia kupna.
- **idSell** - zmienna zawierająca identyfikator zlecenia sprzedaży.
- **pozycje** - zmienna zawierająca ilość walorów w zleceniu kupna.
- **liczba** - zmienna zawierająca ilość walorów na rachunku.
- **cena** - zmienna pomocnicza do obliczania ilości walorów do kupienia.
- **getLimit** - funkcja obliczająca limit ceny w zleceniach w zależności od wybranych opcji oraz kierunku zlecenia (kupno/sprzedaż)
- **getLimAkt** - funkcja obliczająca limit aktywacji w zleceniach w zależności od wybranych opcji oraz kierunku zlecenia (kupno/sprzedaż)

Dla prawidłowego działania systemu handlującego w czasie rzeczywistym niezbędne jest korzystanie ze zmiennych statycznych, ponieważ przechowują one stan z poprzednich obliczeń – Amibroker po każdej transakcji (lub zadeklarownym opóźnieniu) na nowo wykonuje dany skrypt AFL „zapominając" o poprzednim stanie zmiennych.

Zmienne statyczne zaczynające się od przedrostka „**sv"** konstruowane są na bazie **symbol, chartID** oraz dodatkowych łańcuchów znakowych aby uzyskać unikalne nazwy. Unikalność nazw zmiennych statycznych jest bardzo ważna, bo są one dostępne dla wszystkich uruchomionych w Amibrokerze skryptów. Unikalność zapewnia więc, że skyrpt AFL na jednym z wykresów nie wpłynie na działanie innego skryptu.

Uwaga! Jeżeli użytkownik korzysta z szablonów przy tworzeniu wykresów, to identyfikatory tak utworzonych wykresów są identyczne! Jeżeli dodatkowo dany walor będzie wyświetlony na dwóch takich wykresach, to nazwy zmiennych statycznych nie będą unikalne! – w takim przypadku należy dodatkowo zmodyfikować nazwy zmiennych dodając np. kolejne cyfry.

W trakcie generowania kodu wynikowego program sprawdza czy nie użyto takich samych nazw w kodzie orygianlnym. Jeżeli zajdzie taki przypadek, to nazwa generowana uzyskuje końcówkę „**_plug**”. Np. jeżeli w kodzie pierwotnym występuje nazwa „**pozycje**”, to kod wynikowy pozostawi tę nazwę i doda własną o nazwie „**pozycje_plug**”.

Program redefiniuje sygnały kupna i sprzedaży w zależności od wybranych opcji w programie:

```
mBuy = LastValue(Ref(Buy, -1)) && Nz(StaticVarGet(svCanBuy), 1);
mSell = LastValue(Ref(Sell, -1)) && Nz(StaticVarGet(svCanSell), 1);
```

Jak widać z przykładu, pobierana jest wartość sygnału z poprzedniej świeczki (**LastValue(Ref(Buy, -1))** ponieważ zadeklarowano, że sygnał pada w **bieżącej** świeczce – taki sposób gwarantuje, że sygnał jest **ustalony** (np. sygnał wynikający z przecięcia dwóch średnich może oscylować w zależności od chwilowej ceny w bieżącej świeczce). Jeżeli użytkownik zadeklaruje, że sygnał pada w poprzedniej świeczce, kod wynikowy użyje funkcji **LastValue(Buy)**.
Dodatkowo badany jest stan znaczników pozwolenia na zlecenie kupna lub sprzedaży (**svCanBuy**/**svCanSell**).

Sekcje kupna i sprzedaży są generowane w programie (wersja rozbudowana) w następujący sposób:

Kupno (dla waloru „**SDG**”, na rachunku „**12345**”):

```
if(mBuy && (pozycje_plug>0))
{
  idBuy = nolBuy("sdg", "12345", pozycje_plug, getLimit(1), getLimAkt(1));
  if(StrLen(idBuy)>0)
  {
    StaticVarSet(svCanBuy, 0);
    StaticVarSet(svCanSell, 1);
  }
}
```

Zlecenie kupna nastąpi jeżeli wystąpi sygnał kupna **mBuy** oraz ilość walorów do kupna **pozycje_plug** jest większa od 0. Funkcja **nolBuy** wysyła zlecenie kupna, w wyniku otrzymujemy identyfikator zlecenia. Jeśli jest pusty łańcuch, to zlecenie nie zostało przyjęte. Jeśli długość identyfikatora nie wynosi 0 (zlecenie przyjęte), to zmieniane są zmienne statyczne, tu blokowane jest kupno i dozwolona sprzedaż, ponieważ wybrano opcję „**Następna transakcja odwrotna do poprzedniej**”.
W kodzie widoczne jest wywołanie funkcji **getLimit** i **getLimAkt** określające limity w zleceniu.

Sprzedaż:

```
if(mSell && (liczba>0))
{
  idSell = nolSell("sdg", "12345", liczba, getLimit(0), getLimAkt(0));
  if(StrLen(idSell)>0)
  {
    StaticVarSet(svCanBuy, 1);
    StaticVarSet(svCanSell, 0);
  }
}
```

Sprzedaż wygląda podobnie jak kupno z tym, że zlecenie wysyłane jest gdy są walory na rachunku. Dotyczy to przypadku akcji. Dla kontraktów warunek **liczba>0** powinien zostać zastąpiony inną konstrukcją w zależności od tego czy system transakcyjny zawsze odwraca pozycje czy też nie. W podanej postaci nigdy nie nastąpi otwarcie pozycji krótkiej.

Przy odwracaniu pozycji (do osiągnięcia końcowej ilości `pozycje_plug`) można skorzystać z konstrukcji:

```
pozycje_sell = pozycje_plug;
if(liczba!=0) pozycje_sell = liczba + pozycje_plug;
if(mSell && (pozycje_sell>0))
{
  idSell = nolSell("sdg", "12345", pozycje_sell, getLimit(0), getLimAkt(0));
  if(StrLen(idSell)>0)
  {
    StaticVarSet(svCanBuy, 1);
    StaticVarSet(svCanSell, 0);
  }
}
```

9. Końcowe uwagi

Wejściowy kod AFL powinien być wcześniej przetestowany w Amibrokerze pod kątem składniowym i logicznym czyli czy wykonuje oczekiwane operacje.

Bezwzględnie zaleca się własnoręczne sprawdzenie wygenerowanego kodu. W pierwszym etapie będzie to wczytanie go do edytora kodu AFL Amibrokera (menu **Analysis->Formula editor**) i uruchomienie narzędzia **Verify syntax**.

Następnie można poprawnie zweryfikowany kod „wrzucić" na wykres wybranego waloru i włączając tryb testowy i korzystając z funkcji **BarReplay** Amibrokera prześledzić działanie programu. W takim teście nie wszystkie funkcje plugina mogą być dogłębnie przetestowane.

W trzecim etapie w trybie testowym należy prześledzić zachowanie się kodu AFL w czasie rzeczywistym.

Dopiero mając pewność, że kod i plugin działają poprawnie można wyłączyć tryb testowy i przejść do prawdziwego automatycznego autotradingu.

Na wszystkich etapach warto do kodu AFL dodać instrukcje **_TRACE(…)** Amibrokera, które wstawione w newralgicznych miejscach kodu pozwalają śledzić zachowanie się poszczególnych funkcji czy zmiennych. Zainteresowanych odsyłam do dokumentacji Amibrokera.