Projekt TKOM

Sprawozdanie Wstępne

Temat: Język maszyny reguł do inwestowania na giełdach walutowych i akcyjnych.

1. Cele

* Głównym celem projektu jest napisanie kompilatora języka maszyny reguł umożliwiającej inwestycje giełdowe.

2. Opis Gramatyki

1. Ogólny plików składowych programu:

1. Pliki konfiguracyjne definiujące dostępne produkty walutowe i produkty akcyjne.

2. Pliki konfiguracyjne zawierające listę zdarzeń, czyli zmian kursów walutowych i cen akcji.

3. Pliku programu, który zawiera listę reguł złożonych z warunków wejścia oraz akcji.

4. Pliku startowego, który zawiera początkowy stan posiadania inwestora.

1. Składnia pliku konfiguracyjnego zawierającego produkty walutowe i akcyjne. Jest to plik .xml, który w ramach zadania będzie odczytywany poprzez gotową bibliotekę i w przypadku nie wystąpienia elementów obowiązkowych lub wystąpienia elementów nadmiarowych rzucić błąd. Przykładowy plik configuracyjny.

* + Tag „*products”*: znacznik otwierający plik configuracyjny produktów. Jest to znacznik obowiązkowy.
  + Tag „*currencies”:* znacznik oznacza rozpoczęcie bloku definicji walut. Jego nieobecność w pliku wskazuje, że nie są definiowane żadne produkty walutowe. Obecność znacznika nie jest obowiązkowa. Blok może być pusty.
  + Tag „*currency”:* znacznik musi wystąpić wewnątrz bloku „*currencies”*. Gdy zostanie wykryty traktowane jest to jako początek definiowania nowego produktu walutowego.
    - Na definicję produktu walutowego składają się dwa elementy nazwa i skrót, które oznaczone są:
      * „*name”:* znacznik, który zawiera pełną nazwę produktu. Jest obowiązkowy.
      * „*abbreviation”*: skrót nazwy produktu, który musi być unikatowy ze wszystkich produktów walutowych. W przypadku powtórzenia się skrótu nazwy kompilator rzuci błąd ze wskazaniem na zduplikowaną nazwę. Znacznik jest obowiązkowy.
  + Tag „*stocks”*: znacznik oznacza rozpoczęcie bloku definicji produktów akcyjnych. Znacznik nie jest obowiązkowy, jego brak oznacza, że plik konfiguracyjny nie definiuje żadnych produktów giełdowych. W przypadku pustej sekcji oznacza brak definicji produktów.
  + Tag „*stock”*: znacznik oznacza rozpoczęcie definicji produktu giełdowego. Znacznik ten musi znajdować się wewnątrz sekcji „*stocks*”.
    - Na definicję produktu akcyjnego składają się dwa elementy: nazwa i waluta, w której można kupować akcje.
      * „*name”:* nazwa produktu. Nazwa musi być unikatowa ze wszystkich nazw produktów akcyjnych. Znacznik jest obowiązkowy. W przypadku powtórzenia się nazwy kompilator rzuci błędem i wskaże zduplikowaną nazwę.
      * „*currency*”: skrót nazwy waluty, w której można kupić akcje. Jeżeli określony skrót nie zostanie zdefiniowany w żadnym pliku konfiguracyjnym kompilator rzuci błędem ze wskazaniem na ten skrót.

<products>

<currencies>

<currency>

<name>yuan</name>

<abbreviation>YUA</abbreviation>

</currency>

<currency>

<name>yen</name>

<abbreviation>YEN</abbreviation>

</currency>

<currency>

<name>dollar USA</name>

<abbreviation>USD</abbreviation>

</currency>

</currencies>

<stocks>

<stock>

<name>CocaCola</name>

<currency>USD</currency>

</stock>

</stocks>

</products>

2. Składnia pliku zdarzeń zawierającego zmiany kursów walut i produktów akcyjnych. Jest to plik .xml, który w ramach zadania będzie odczytywany poprzez gotową bibliotekę i w przypadku nie wystąpienia elementów obowiązkowych lub wystąpienia elementów nadmiarowych rzucić błąd. Przykładowy plik zdarzeń:

* *„events\_config:* tag najwyższego poziomu, jego obecność jest obowiązkowa.
* „*start”*: tag wskazujący początkowy stan systemu. Nie jest obowiązkowy, ale musi wystąpić w przynajmniej jednym pliku zdarzeń. Wewnątrz tego bloku określane są wszystkie startowe wartości walut i ich przeliczniki oraz wszystkie startowe kursy akcji. Jeżeli wartość startowa, któregokolwiek z elementów nie zostanie określona to kompilator rzuci błędem. Jeżeli żaden blok start nie zostanie znaleziony to kompilator rzuci błędem.
  + „*date”:* zawiera datę w formacie yyyy.MM.dd. Tag ten musi pojawić się przynajmniej w jednym bloku startowym. Brak obecności skutkuje błędem kompilatora.
  + „*currencies”:* blok wewnątrz, którego definiowane są kursy początkowe walut, czyli zgodne z dniem wskazanym przez pole „date”. Złożony z wielu elementów „*currency”.* Blok może być pusty, może też nie wystąpić.
    - „*currency”:* blok definiujący początkową wartość określonej waluty w przeliczeniu na fikcyjną walutę uniwersalna.
      * „*name”*: skrót nazwy waluty. Skrót musi zostać zdefiniowany w plikach konfiguracyjnych.
      * „*value”*: jest to wartość zmiennoprzecinkowa o dwóch miejscach znaczących, która oznacza ilość wydanej waluty celem nabycia jednej jednostki waluty fikcyjnej.
  + „*stocks*”: blow wewnątrz, którego definiowane są kursy początkowe akcji, czyli zgodne z dniem wskazanym przez pole „date”. Złożony z wielu elementów „*stock”.* Może być pusty lub nie występować w poszczególnych plikach, ale wszystkie kursy początkowe muszą być zdefiniowane.
    - „*stock”*: blok definiujący początkową wartość określonej akcji w walucie przypisanej do danych akcji.
      * „*name”:* nazwa akcji, musi wystąpić w plikach konfiguracyjnych.
      * „*price*”: cena w walucie przypisanej do akcji. Liczba zmiennoprzecinkowa z „.” jako separatorem, o dwóch miejscach znaczących.
  + „*events”:* blok definiujący zdarzenia zachodzące w rzeczywistości, na skutek, których będą wykonywane różne reguły zdefiniowane w programie. Sekcja może nie występować lub być pusta.
    - „*event”:*  definicja pojedynczego zdarzenia, wszystkie jej pola są obowiązkowe.
      * „*type”:*  może przyjmować dwie wartości: STOCK lub CURRENCY. Które odpowiadają zmianie kursu akcji lub zmianie kursu waluty.
      * „*name”:* nazwa akcji lub skrót nazwy waluty. Nazwa ta musi występować w plikach konfiguracyjnych.
      * „*date”:* data nastąpienia zdarzenia. Jeżeli data zdarzenia jest ustawiona wcześniej niż data startu to kompilator wyrzuci ostrzeżenie. Format daty to yyyy.MM.dd.
      * „*change”:* zmiana w punktach procentowych, w przypadku spadku poprzedzona znakiem „-”. Separatorem jest znak „.”. Dostępna jest precyzja zmiany do 4 cyfr znaczących.

<events\_config>

<start>

<date>2016.05.09</date>

<currencies>

<currency>

<name>USD</name>

<value>1.75</value>

</currency>

</currencies>

<stocks>

<stock>

<name>CocaCola</name>

<price>300.00</price>

</stock>

</stocks>

</start>

<events>

<event>

<type>STOCK</type>

<name>CocaCola</type>

<date>2016.05.10</date>

<change>17.0</change>

</event>

</events>

</events\_config>

3. Plik startowy zawiera startowy stan własności inwestora. Jest to plik XML, który opisuje ile jakiej waluty i akcji posiada inwestor. Plik jest parsowany za pomocą gotowej biblioteki i w przypadku nie wystąpienia elementów obowiązkowych lub wystąpienia elementów nadmiarowych rzucić błąd. Przykładowy plik:

* „*start”:* tag otwierający przetwarzanie, jest obowiązkowy.
  + „*currencies”:* blok wewnątrz, wewnątrz którego definiowane są . Złożony z wielu elementów „*currency”.* Blok może być pusty, może też nie wystąpić.
    - „*currency”:* blok definiujący ilość posiadanej waluty przez inwestora
      * „*name”*: skrót nazwy waluty. Skrót musi zostać zdefiniowany w plikach konfiguracyjnych.
      * „*value”*: jest to wartość zmiennoprzecinkowa o dwóch miejscach znaczących, która oznacza ilość posiadanej waluty.
  + „*stocks*”: blok wewnątrz, którego definiowane są posiadane akcje. Złożony z wielu elementów „*stock”.* Może być pusty lub nie występować w poszczególnych plikach, ale wszystkie kursy początkowe muszą być zdefiniowane.
    - „*stock”*: blok definiujący początkową liczbę posiadanych akcji
      * „*name”:* nazwa akcji, musi wystąpić w plikach konfiguracyjnych.
      * „*amount*”: liczba posiadanych akcji.

<start>

<currencies>

<currency>

<name>USD</name>

<value>1.75</value>

</currency>

</currencies>

<stocks>

<stock>

<name>CocaCola</name>

<amount>300</amount>

</stock>

</stocks>

</start>

4. Plik programu

* Sekcja konfiguracji:
  + *config –* słowo kluczowe poprzedzające nazwę pliku konfiguracyjnego, czyli takiego, który definiuje waluty i akcje. Plik musi znajdować się w tym samym katalogu co główny plik reguł. Brak pliku skutkuje ostrzeżeniem.
    - Przykład: config example\_config.xml
    - Blok konfiguracyjny: config <plik\_konfiguracji.xml>
    - Może wystąpić wiele bloków konfiguracyjnych ale musi się pojawić co najmniej 1.
  + *events –* słowo kluczowe poprzedzające nazwę pliku ze zdarzeniami, czyli takiego, który definiuje zdarzenia zaszłe w symulacji. Plik musi znajdować się w tym samym katalogu co główny plik reguł. Brak pliku skutkuje ostrzeżeniem.
    - Przykład: events example\_events.xml
    - Blok zdarzeń: events <plik\_zdarzen.xml>
    - Może wystąpić wiele bloków zdarzeń ale musi się pojawić co najmniej 1.
  + *start –* słowo kluczowe poprzedzające nazwę pliku z początkowym stanem posiadania inwestora. Plik musi znajdować się w tym samym katalogu co główny plik reguł. Brak pliku skutkuje ostrzeżeniem.
    - Przykład: start start\_config.xml
    - Blok startowy: start <plik\_startu.xml>
    - Może wystąpić wiele bloków startu, ale musi pojawić się co najmniej 1.
* Sekcja reguł:
  + Rozpoczyna ją pierwsza zadeklarowana reguła. Reguły można deklarować na dwa sposoby:
    - *rule <plik\_reguly.invrl>,* gdzie plik\_reguly.invrl jest plikiem deklarującym regułę zgodnie z drugim typem deklaracji. Można zawrzeć dowolną liczbę takich deklaracji.
    - Pełną deklaracją reguły.
* Pełna deklaracja reguły.
  + Rozpoczęta słowem kluczowym: *rule.*
  + Ograniczona za pomocą nawiasów {}
  + Każde pole reguły jest zakończone „;”
  + Nazwy pól reguły od wartości oddzielone są za pomocą „:”
  + Tak więc ogólnie przyjmują postać: *nazwa\_pola: wartosc\_pola;* Pola dla reguł są obowiązkowe
    - Pierwszym polem każdej reguły jest *id*. Jest to wartość liczbowa, która jednoznacznie identyfikuje regułę. Nie mogą wystąpić dwie reguły o jednakowym id. Jeżeli tak się stanie to kompilacja zostanie przerwana z błędem.
      * Przykład: id: 10;
    - Drugim polem każdej reguły jest *priority.* Określa ona priorytet sprawdzenia warunków wejścia do reguły i wykonania jej akcji