Specyfikacja funkcjonalna
Projekt „Group of Visionaires”

Mateusz Smoliński

11 listopada 2018

Spis treści

1 Opis projektu

Problem Grupy Wizjonerów dotyczy wymiany walut. Jej członkowie zauważyli, że dzięki dokładnej analizie bieżących kursów można zarobić, wymieniając odpowiednie waluty we właściwym czasie. Przy sprzyjających okolicznościach można spotkać sytuację ekonomicznego arbitrażu, czyli kombinacji, przy której po wymianie kilku walut i powrocie do waluty początkowej można powiększyć początkową kwotę.

Celem projektu jest napisanie programu, który wykona dwa zadania związane z wyżej wymienionym problemem:

- wykrywanie korzystnej ścieżki wymiany dla wskazanej waluty,
- wykrywanie sytuacji ekonomicznego arbitrażu, czyli kombinacji, gdy kursy walut są tak ułożone, aby wymiana cykliczna zwracała więcej niż kwota wejściowa.

Program będzie napisany w języku Java. Będzie on otrzymywał 3 lub 4 argumenty:

- nazwę pliku z danymi, w którym zawarte są bieżące kursy walut,
- kwotę wejściową,
- walutę wejściową,
- walutę wyjściową (argument opcjonalny).

Efektem pracy programu będzie wypisanie na standardowym strumieniu wyjścia ciągu najkorzystniejszej wymiany walut, informacja o braku takiego ciągu lub odpowiedni komunikat o błędzie w przypadku niepoprawnych danych.

2 Wymagania funkcjonalne

- Program może przyjąć 3 lub 4 argumenty – w przypadku, gdy użytkownik poda tylko jeden skrót waluty, program postara się odnaleźć cykl spełniający warunki ekonomicznego arbitrażu,
- w przypadku, gdy cykl nie zostanie odnaleziony lub nie będzie on dawać zysku dla wprowadzonej kwoty, program wypisze stosowny komunikat,
- jeśli użytkownik poda dwa skróty walut, program wyszuka najkorzystniejszy ciąg wymiany,
- w przypadku, gdy nie będzie możliwa wymiana jednej waluty na drugą, program wypisze stosowną informację,
- program umożliwia zmianę szybkości generowania kolejnych obrazów przy pomocy suwaka,
- użytkownik może zlecić wykonanie jednej zmiany generacji przy pomocy specjalnego przycisku,
- program zapewnia dokładną obsługę błędów i poinformuje użytkownika o każdej sytuacji nietypowej – zostały one podane w punkcie 4.

3 Opis wejścia i wyjścia

W celu zapewnienia poprawnego działania programu użytkownik musi uruchomić go z pewnymi argumentami. Pierwszym z nich powinna być nazwa pliku (z adresem, jeżeli plik nie znajduje się w katalogu z programem). Kolejnym argumentem jest kwota wejściowa, która będzie podlegać zamianie. Następny argument powinien określić walutę podanej kwoty – powinien on mieć formę skrótu złożonego z trzech dużych liter, wcześniej zadeklarowanego w podanym pliku z danymi. Jeśli użytkownik chce szukać cyklu spełniającego ekonomiczny arbitraż, nie podaje kolejnego argumentu. Jeśli chce znaleźć najlepszy kurs wymiany dwóch walut, powinien podać jeszcze jeden skrót walutowy, oznaczający właśnie walutę docelową.

Przykłady poprawnie wprowadzonych ciągów argumentów:

```
dane.txt 1000 EUR
dane.txt 1000 GBP EUR
```

Plik podawany jako pierwszy argument musi być uprzednio przygotowany przez użytkownika. Przykładowy poprawnie zapisany plik znajduje się poniżej:

```
# Waluty (id | symbol | pelna nazwa)
0 EUR euro
1 GBP funt brytyjski
2 USD dolar amerykanski
# Kursy walut (id | symbol (waluta wejsciowa) | symbol (waluta wyjsciowa)
 | kurs | typ oplaty | oplata
0 EUR GBP 0.8889 PROC 0.0001
1 GBP USD 1.2795 PROC 0
2 EUR USD 1.1370 STALA 0.025
3 USD EUR 0.8795 STALA 0.01
```

Linijki rozpoczynające się znakiem „#" są traktowane jako oddzielenie deklaracji walut wraz z ich skrótami od poszczególnych kursów walut. Pierwsza linijka rozpoczęta od „#" oznacza początek deklaracji, dalsza część tej linii nie jest rozpatrywana przez program. Deklaracje są podawane w postaci: numer identyfikacyjny, skrót i pełna nazwa, oddzielane spacjami. Następna „#" działa jako rozpoczęcie podawania kursów walut, dalsza część linii również nie będzie brana pod uwagę. Wszystkie kursy powinny być podane w postaci: numer identyfikacyjny, waluta wejściowa, waluta wyjściowa, kurs, typ opłaty („PROC" jako prowizja przeliczana procentowo lub „STAŁA" jako prowizja naliczana stałą wartością od waluty wyjściowej) oraz wartość tej opłaty. W przypadku podania nieznanej waluty, dwóch takich samych skrótów walut lub niepoprawnej nazwy typu opłaty linijka zostanie zignorowana przez program.

Wyjście programu w przypadku powodzenia będzie przedstawiać szukany ciąg wraz z otrzymanym w tym poszukiwaniu całkowitym kursem oraz wartością uzyskaną w wyniku tej wymiany na podstawie wartości podanej na wejściu. W przypadku jakiejkolwiek sytuacji wyjątkowej zostanie wypisany odpowiedni komunikat, określony w następnym punkcie specyfikacji.

Przykład poprawnego efektu działania programu:

```
1000 EUR -> GBP -> USD -> 1137.34 USD
1000 EUR -> GBP -> USD -> EUR -> 1000.30 EUR
```

Komunikaty, jakie wypisuje program w przypadku niepowodzeń są wypisane punkcie 4.

4 Sytuacje wyjątkowe

Jednym z założeń projektu jest to, że dane otrzymywane od użytkownika oraz argumenty, z jakimi uruchamiany jest program nie muszą być poprawnie przygotowane lub nie muszą prowadzić do założonego wyniku. Poniżej znajduje się lista błędów i sytuacji wyjątkowych, jakie są obsługiwane przez program wraz z reakcją programu na ich wystąpienie.

1. Dotyczące argumentów wywołania programu
    - Nie podano żadnych argumentów lub podano ich nieprawidłową ilość. Wyświetlony zostanie komunikat: „ERROR: program needs 3 or 4 arguments to work properly."
    - Nie uda się otworzyć do odczytu pliku z danymi, podanym jako pierwszy argument. W takim wypadku program wyświetli komunikat: „ERROR: file <name_of_file> was not found. Make sure the file is located in proper place and try again."
    - Plik podany jako tekstowy okaże się być innego formatu – wtedy program wypisze komunikat: „ERROR: file <name_of_file> is not proper .txt file. Check if you have chosen right file and try again."

- Drugi argument nie zostanie rozpoznany jako poprawna wartość kwoty. W takim przypadku program wyświetli komunikat: „ERROR: Value given as amount of money was not read properly. Verify the second argument and try again."
- Trzeci argument nie zostanie rozpoznany jako jedna z walut wprowadzonych w pliku z danymi. W takim przypadku program wyświetli komunikat: „ERROR: The input currency was not recognised. Verify the third argument and try again."
- Czwarty argument zostanie wprowadzony, ale nie zostanie rozpoznany jako jedna z walut wprowadzonych w pliku z danymi. Wtedy program wyświetli następujący komunikat: „ERROR: The output currency was not recognised. Verify the fourth argument and try again."

2. Dotyczące danych wprowadzonych przez użytkownika w załączonym pliku tekstowym

   - Plik nie zaczyna się od linii rozpoczynającej deklarację walut, rozpoczynającej się od znaku „#". W takim wypadku program wypisze komunikat: „ERROR: Data file starts in unrecognizable way. Make sure that you have attached proper file and try again."
   - Plik zawiera więcej niż dwie linie oddzielające (rozpoczynające się od znaku „#"). W takim wypadku program zignoruje kolejne linie i wypisze na ekranie komunikat: „Warning: Data file contains more than 2 separating lines. Everything below the third will be ignored."
   - Plik zawiera mniej niż dwie linie oddzielające (rozpoczynające się od znaku „#"). W takim wypadku program wyświetli komunikat: „ERROR: Data file does not contain enough separating lines. Make sure that you have attached proper file and try again."
   - Jedna z linii wprowadzonych w pliku z danymi zawiera niepoprawną wartość, lub nie zawiera odpowiedniej liczby wartości. W takim wypadku linia zostanie zignorowana przez program i wypisze on na ekranie następujący komunikat: „Warning: Data file contains invalid data in <number_of_line> line. It will be ignored by program."

3. Dotyczące niemożności wykonania polecenia dla podanych danych

   - Przy wprowadzeniu trzech argumentów, program nie odnalazł żadnego cyklu lub żaden z odnalezionych cykli nie jest korzystny. W takim wypadku zostanie wyświetlony komunikat: „No economic arbitration found."
   - Przy wprowadzeniu czterech argumentów, program nie odnalazł żadnej drogi pozwalającej na wymianę waluty wejściowej na wyjściową. W takim wypadku program wypisze komunikat: „Exchange is not possible".

5 Testy akceptacyjne programu

Dla sprawdzenia poprawności działania programu zostaną przeprowadzone testy akceptacyjne. Poniżej znajduje się lista testów sprawdzających różne możliwe scenariusze.

- Test z podaniem poprawnego pliku z niewielką ilością danych, oferującą dwie ścieżki dojścia do waluty docelowej, uruchamiany z 4 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie korzystniejszej z tych dwóch poprawnych ścieżek.
- Test z podaniem poprawnego pliku z niewielką ilością danych, oferującą cykl umożliwiający korzystną wymianę, uruchamiany z 3 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie tego korzystnego cyklu.
- Test z podaniem poprawnego pliku z niewielką ilością danych, uniemożliwiającą wymianę do waluty docelowej, uruchamiany z 4 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie komunikatu o braku możliwości wymiany.

- Test z podaniem poprawnego pliku z niewielką ilością danych, nieposiadającego żadnych cykli, uruchamiany z 3 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie komunikatu o braku arbitrażu.
- Seria testów z podaniem poprawnego pliku z niewielką ilością danych, uruchamiany z różnymi niepoprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie komunikatu o odpowiednim błędzie dla każdej niepoprawnej kombinacji (podanie kwoty niebędącej liczbą, waluty nie zadeklarowanej w pliku, niewłaściwe zaadresowanie pliku, zbyt dużo i zbyt mało argumentów).
- Test z podaniem poprawnego pliku z dużą ilością danych (powyżej 30 kursów), zawierających dużo połączeń, uruchamiany z 4 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie najkorzystniejszej z kilku przygotowanych dróg.
- Test z podaniem poprawnego pliku z dużą ilością danych (powyżej 30 kursów), zawierających 3 cykle, uruchamiany z 4 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie najkorzystniejszego cyklu.
- Seria testów z podaniem niepoprawnego pliku, uruchamiany z 4 poprawnymi argumentami. Oczekiwanym efektem testu jest wypisanie komunikatu o odpowiednim błędzie dla każdej niepoprawnej kombinacji (zbyt dużo oddzielających linii, brak oddzielających linii, niepoprawne dane w jednej z linii).