

Projekt indywidualny Specyfikacja Funkcjonalna

Arkadiusz Michalak

Algorytmy i struktury danych
11.11.2018

Spis treści

| | | |
|---|-----------------------------------|---|
| 1 | Opis problemu | 2 |
| 2 | Funkcjonalność | 2 |
| 3 | Sposób obsługi | 2 |
| 4 | Sposób organizacji pliku | 3 |
| 5 | Sposób wypisu rezultatów | 4 |
| 6 | Opis wymagań niefunkcjonalnych | 4 |
| 7 | Scenariusze testów akceptacyjnych | 4 |

1 Opis problemu

Celem projektu jest stworzenie programu którego zadaniem będzie przeliczanie wymian walutowych. Będąc zainteresowanym wymianą waluty z jednej na drugą można to zrobić bezpośrednio bądź też pośrednio przy użyciu innych walut niż wejściowa i docelowa. Określenie który ze sposobów będzie korzystniejszy i w jakiej kolejności wymieniać pośrednie waluty stwarza duże wyzwanie. Dzieje się tak ze względu na znaczną ilość różnych walut i rozmaite powiązania między nimi, oraz ewentualne prowizje od wymiany, zarówno stałe jak i wyrażone procentem kwoty którą otrzymujemy. Analizując zestaw danych można znaleźć sytuacje w której wymieniając jedną walutę wracamy do posiadania tej samej waluty jednak osiągamy zysk. Takie działanie będziemy określać mianem *arbitrażu ekonomicznego*.

2 Funkcjonalność

Program będzie posiadał dwie funkcjonalności:

- wskazanie najkorzystniejszego sposobu wymiany danej waluty wejściowej na określoną walutę wyjściową,
- sprawdzenie czy w podanym zbiorze danych istnieje arbitraż ekonomiczny.

3 Sposób obsługi

Do uruchomienia programu wymagane jest zainstalowanie maszyny wirtualnej java. Obsługa będzie odbywać się w całości z wiersza poleceń, wywołując program należy podać odpowiednie argumenty w następującej kolejności:

1. Którą funkcjonalność uruchomić:
 - arbitraz - w celu sprawdzenia czy istnieje arbitraż,
 - wymiana - w celu znalezienia najbardziej opłacalnego sposobu wymiany.
2. Ścieżka do pliku zawierającego dane, format i sposób organizacji danych w pliku opisany w sekcji **4 Sposób organizacji pliku**.
3. Nazwa waluty wejściowej, wyrażoną zunifikowanym 3-literowym kodem.

4. Ilość waluty wejściowej.
5. Nazwa waluty wyjściowej, jeśli wybrano opcje wymiany. Przykład:
program arbitraz ./walut.txt EUR 10000.50
program wymiana ./walut.txt EUR 10.5 PLN

4 Sposób organizacji pliku

Plik wejściowy musi być w formacie tekstowym *.txt*, do separacji poszczególnych wartości należy użyć spacji. Plik ma się składać z dwóch sekcji:

1. Informacje na temat walut, każda waluta w oddzielnym wierszu opisana trzema wartościami:
 - indeks waluty - kolejne liczby naturalne, zaczynając od 0,
 - 3-literowy kod waluty,
 - pełna nazwa waluty.
2. Informacje na temat kursów walut względem siebie. Każdy kurs w oddzielnym wierszu opisany sześcioma wartościami:
 - indeks kursu - kolejne liczby naturalne, zaczynając od 0,
 - kod waluty wejściowej,
 - kod waluty wyjściowej,
 - kurs,
 - rodzaj prowizji - PROC - jeśli procentowa, STAŁA - jeśli stała,
 - wysokość prowizji.

Przykład:

0 BRL real brazylijski

1 PLN polski złoty

2 EUR euro

0 BRL PLN 0,9442 STAŁA 5

1 EUR PLN 4,2972 PROC 0,001

5 Sposób wypisu rezultatów

Zarówno komunikaty o błędach jak i wyniki prawidłowego działania mają zostać wyświetlone na konsolę. W przypadku obliczania arbitrażu należy wypisać ścieżkę walut dla jakiej zachodzi oraz o ile dokładnie wzrasta suma posiadanych pieniędzy. Dla poszukiwania najlepszej wymiany należy wypisać ścieżkę kolejnych walut oraz uzyskaną ilość waluty wyjściowej. Dla wypisu wartości pieniędzy przyjmuje się zaokrąglenie do czwartej cyfry po przecinku, zgodnie ze standardem większości kantorów.

6 Opis wymagań niefunkcjonalnych

Główne i najważniejsze wymaganie niefunkcjonalne to intuicyjna obsługa programu i prosta komunikacja z użytkownikiem. Komunikaty o błędach powinny być dokładne i pozwalające szybko naprawić użytkownikowi błąd. Przykład:

”W linii 54 pliku podano nieprawidłową wartość prowizji. Prowizja nie może być ujemna ”

Program powinien płynnie działać na większości komputerów domowego użytku, które spełniają następujące wymagania:

- 2GB pamięci RAM,
- procesor 2-rdzenie 1.8 GHz,
- karta graficzna zintegrowana,
- zainstalowana maszyna wirtualne Java wersja 1.8.0
- system operacyjny windows 7, 8, 10
- program obsługi wiersza poleceń np: PowerShell, CMD

7 Scenariusze testów akceptacyjnych

1. Podanie pliku w formacie innym niż tekstowy.
2. Podanie błędnej ścieżki do pliku.
3. Podanie pliku pustego.
4. Podanie pliku bez indeksów walut.

5. Podanie pliku bez nazwy jednej z walut.
6. Żądanie znalezienia arbitrażu dla 100 walut połączonych każda z każdą.
7. Żądanie znalezienia najlepszej wymiany kiedy waluty nie są połączone żadną ścieżką.
8. Żądanie znalezienia arbitrażu dla zbioru walut i kursów w którym arbitraż nie istnieje.
9. Nie podanie argumentów wejściowych.
10. Podanie zbyt małej liczby argumentów wejściowych.
11. Podanie waluty która nie istnieje w zbiorze walut pliku.
12. Żądanie znalezienia najlepszej wymiany dla 20 walut z istniejącym połączeniem pomiędzy walutą wejściową a wyjściową.
13. Podanie ujemnej wartości kursu.
14. Podanie zerowej wartości kursu.
15. Podanie ujemnej wartości waluty wejściowej.

Program będzie uznany za działający jeśli pozytywnie przejdzie wszystkie testy. Pod słowem pozytywnie rozumie się wyświetlenie rezultatów, lub odpowiedni komunikat błędu który pozwoli użytkownikowi naprawić błąd.