

Politechnika Warszawska  
Wydział Elektryczny

---

SPECYFIKACJA IMPLEMENTACYJNA  
„ARBITRAGE”

---

*Autor:*  
GRZEGORZ KOPYT

7 listopada 2018

# Spis treści

1	Opis algorytmu	1
2	Diagram rozwiązania problemu	2
3	Diagram klas	2
4	Testy	2
5	Informacje o sprzęcie i oprogramowaniu	2

---

## 1 Opis algorytmu

Kluczowym algorytmem pracy tego programu jest algorytm szukający najkorzystniejszych ścieżek po grafie walut (*Currency*) i ofert (*Offer*). W tym przypadku najkorzystniejszą ścieżką jest ta, której koszt jest najmniejszy. W tym celu wykorzystam algorytm Bellmana-Forda.

W obiektach *Currency* pola *cost* i *charge* mają mieć takie wartości, które kwotę końcową pozwolą obliczyć według wzoru:

*k* - kwota początkowa w początkowej walucie

*w* - kwota końcowa w walucie docelowej

$$w = k / \text{cost} - \text{charge}$$

Wartości pól *cost*, *charge* oraz *previous* w obiektach *Currency* są zainicjowane wartościami *null*.

Algorytm operuje na grafie, w którym węzłami są obiekty *Currency*, a gałęziami obiekty *Offer*, a jego działanie jest następujące:

1. Obiekt klasy *Broker* otrzymuje od *GUI* informacje jakie zadanie ma zrealizować i na jakich walutach operować. Są dwa warianty pracy: *wymiana waluty*(patrz 2a.) lub *arbitrage*(patrz 2b.).
2. (a) Wymiana waluty
  - i. *Broker* zleca *FinancialAnalyst*, aby ustawił koszty odwiedzenia węzłów w grafie zaczynając od podanej przez użytkownika waluty wyjściowej.
    - A. Z *HashMapy currencies* pobiera *Set* kluczy, z którego robi tablicę, kluczy.
    - B. Zaczynając od węzła początkowego iteruje po tablicy w pętli (jeśli napotka koniec tablicy zaczyna od początku).
    - C. Ustawia *cost* i *charge* węzła początkowego na 0.
    - D. Ustawia wartości pól *cost* i *charge* w węzłach sąsiadujących z węzłem początkowym według wzorów:

*newCost* - wartość pola *cost* w sąsiadującym węźle

*cost* - wartość pola *cost* w obecnym węźle

*rate* - wartość pola *rate* w gałęzi *Offer*

*percent* - wartość pola *percentCharge* w gałęzi *Offer*

$$\text{newCost} = \text{cost} / \text{rate} * (1 - \text{percent})$$

*newCharge* - wartość pola *charge* w sąsiadującym węźle

*charge* - wartość pola *charge* w obecnym węźle

*rate* - wartość pola *rate* w gałęzi *Offer*

*standingCharge* - wartość pola *standingCharge* w gałęzi *Offer*

$newCharge = charge/rate + standingCharge$

Wartość *previous* sąsiadującego węzła ustawiamy na obecny węzeł.

**UWAGA!** Powyższych operacji dokonujemy, jeśli nowa wartość pola *cost* jest mniejsza od poprzedniej.

E. Następnie przechodzimy do kolejnego obiektu *Currency* w tablicy:

- Jeśli wartość jego pola *cost* wynosi *null* to przechodzimy do kolejnego obiektu w tablicy powtarzając podpunkt E.
- Jeśli wartość jego pola *cost* jest różna od *null* to powtarzamy czynności z podpunktów D, a potem E.

ii. Teraz aby podać najkrótszą drogę do waluty z waluty podanej wcześniej wystarczy wejść w walute docelową i prześledzić szlak *previous* :) warto dodać na początek sprawdzenie na jaką walute ustawiony jest graf, może nie będzie trzeba liczyć.

(b) Arbitrage

---

## 2 Diagram rozwiązania problemu

---

## 3 Diagram klas

---

## 4 Testy

---

## 5 Informacje o sprzęcie i oprogramowaniu

---