

# Programowanie Równoległe – zadanie 1

26.02.2017

Zaprojektuj i zaimplementuj protokół synchronizacji (oparty na semaforach) dla następującego problemu.

## Problem

1. Trzy fabryki produkują Ołów, Siarkę i Rtęć (jedna Fabryka dla każdego rodzaju surowca). Alchemicy używają tych surowców do wytwarzania złota. Ponadto, źli Czarownicy rzucają klątwy na Fabryki, aby zatrzymać produkcję, a dobrzy Czarodzieje usuwają te klątwy.
2. Jeśli Fabryka nie jest przeklęta (czyli liczba ciężących na niej klątw wynosi 0), produkuje swój surowiec i magazynuje go. Produkcja jednej jednostki surowca zajmuje pewien losowy czas. Jeśli dwie jednostki surowca zostały już wyprodukowane, Fabryka czeka (nie ma więcej miejsca w magazynie). Fabryka czeka także, jeśli jest przeklęta.
3. Czarownik, w losowych odstępach czasu, rzuca klątwę na losową Fabrykę. Klątwy dodają się, t.j. jeśli Fabryka jest już przeklęta, zostaje przeklęta drugi raz i dwie klątwy muszą być usunięte, aby Fabryka zaczęła znowu działać.
4. Czarodziej, w losowych odstępach czasu, usuwa jedną klątwę z każdej przeklętej Fabryki.
5. Istnieją cztery gildie Alchemików, próbujących uzyskać złoto:  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $\mathcal{C}$  i  $\mathcal{D}$ .
6. Alchemicy  $\mathcal{A}$  próbują uzyskać złoto z ołowiu i rtęci.
7. Alchemicy  $\mathcal{B}$  próbują uzyskać złoto z siarki i rtęci.
8. Alchemicy  $\mathcal{C}$  próbują uzyskać złoto z ołowiu i siarki.
9. Alchemicy  $\mathcal{D}$  próbują uzyskać złoto z ołowiu, siarki i rtęci.

10. Alchemicy pozyskują surowce z Fabryk, mogą brać surowce, które są już w magazynie. Alchemicy pojawiają się w losowej kolejności. Jest pewien (losowy) odstęp między dwoma kolejnymi Alchemikami.
11. Każdy Alchemik próbuje zebrać wszystkie potrzebne surowce. Jeśli nie są one dostępne, Alchemik czeka.
12. Po zebraniu wszystkich surowców, każdy Alchemik idzie do swego laboratorium (i nie pojawia się więcej w naszej historii).

## Aspekty techniczne

1. Wypisuj na konsolę komunikaty dotyczące aktualnego stanu programu.
2. Każda postać (Fabryka, Czarownik, Czarodziej, Alchemik) jest reprezentowana przez osobny proces/wątek. Dodatkowe wątki lub procesy mogą być dodawane, jeśli to konieczne.
3. Liczba Czarowników i Czarodziejów jest parametrem (można zacząć od 3) i musi być możliwość jej zmiany.
4. Wybierz jakieś ograniczenia na odstępy czasu wymienione w zadaniu, aby widać było, co się dzieje. Musi być możliwość łatwej zmiany tych ograniczeń. Protokół synchronizacji nie może być oparty na tych ograniczeniach!
5. Rozwiązanie powinno oferować maksymalną równoległość. Następująca sytuacja jest błędna:  
Dostępna jest rtęć i siarka (ale nie ołów). Alchemicy  $\mathcal{B}$  czekają i są zablokowani przez Alchemika  $\mathcal{D}$ , który pojawił się wcześniej.
6. Unikaj zagłodzenia Alchemików  $\mathcal{D}$ .
7. Rozwiązanie, w którym Alchemik pobiera jakieś surowce i je zwraca (jeśli nie wszystko jest dostępne) nie jest najlepsze i nie jest też warte 10 punktów.
8. Aktywne czekanie jest zawsze błędem.
9. Synchronizacja powinna używać tylko prostych semaforów i metod (Java: `acquire()` i `release()`, C#: `WaitOne()` i `Release()`, inne języki: w razie wątpliwości, spytaj nauczyciela).

**Termin nadsyłania rozwiązań: 19.03.2018, 07:00 AM.**