

## **Zadanie 1.**

**Termin wykonania: ~~2019-03-24~~ 2019-03-31**

Należy utworzyć projekt programu współbieżnego, stanowiący symulator działania przedsiębiorstwa.

Zadanie pierwsze polega na doprowadzeniu projektu do prostego stanu początkowego:

Na początek ma być to mała firma zarządzana jedno-osobowo, z niewielką liczbą pracowników.

Prezes firmy, w losowych odstępach czasu, wymyśla kolejne zadania do wykonania dla pracowników i umieszcza je na liście zadań.

(Przyjmujemy, że zadanie ma postać rekordu o następujących polach:

- pierwszy argument,
- drugi argument,
- operator arytmetyczny: dodawanie, odejmowanie albo mnożenie.

Zadanie polega na “wytworzeniu” wyniku operacji arytmetycznej.)

Każdy pracownik, co pewien czas, pobiera kolejne zadanie z listy zadań i je wykonuje. W wyniku powstaje pewien produkt, który pracownik umieszcza w magazynie.

Do magazynu, co pewien czas, przychodzi klient i zabiera (kupi) jakieś produkty.

Osoby powinny być symulowane przez osobne wątki.

Ze względu na konieczność synchronizacji dostępu, można zaimplementować też wątki (serwery) do obsługi poszczególnych struktur danych (listy zadań, magazynu).

Symulator ma działać w dwóch trybach:

- w trybie “gadatliwym” albo
- w trybie “spokojnym”.

W trybie “gadatliwym” wypisywane są na bieżąco komunikaty o zdarzeniach które zachodzą w symulowanym przedsiębiorstwie -- taki rodzaj “tekstowej animacji” przedstawiającej jego działanie. Najlepiej gdyby każdy wątek symulujący prezesa, pracownika albo klienta sam drukował komunikat w chwili wykonania istotnego działania.

W trybie “spokojnym”, działający symulator oczekuje na polecenia użytkownika. Mogą to być polecenia żądające wyświetlenia pewnych informacji na temat bieżącego stanu firmy (na przykład wyświetlenie stanu magazynu, albo wyświetlenie listy zadań do wykonania). W tym trybie, oprócz wątków symulujących firmę, powinien też być wątek służący do interakcji z

użytkownikiem, który w pętli wyświetla menu dostępnych poleceń i czeka na wybór użytkownika wczytany z terminala.

Istotne parametry symulatora (takie jak n.p. tempo działania, pojemność magazynu, maksymalny rozmiar listy zadań, liczba pracowników, itp ...) mają być konfigurowalne. Należy w tym celu umieścić w osobnym pliku źródłowym (w osobnym pakiecie) zbiór definicji stałych, których wartości konfiguruja symulator. Na przykład, każda osoba w firmie będzie symulowana przez wątek wykonujący nieskończoną pętlę, gdzie w każdej iteracji jest instrukcja typu "sleep" lub "delay" opóźniająca jego działanie. Wtedy w pliku konfiguracyjnym można zdefiniować stałą, której wartość ma wpływ na wielkość tego opóźnienia i w ten sposób regulować tempo działania całego symulatora. Należy przemyśleć jak powinien wyglądać zbiór parametrów. (Na przykład, tempa działania prezesa, pracownika i klienta powinny być regulowane przez osobne parametry.) Należy również dokładnie przetestować działanie swojego programu przy różnych wartościach parametrów, i tak ustalić wartości domyślne aby wersja zgłoszona do sprawdzenia działała w sposób rozsądny bez potrzeby zmian.

### **Punktacja:**

- Implementacja rozwiązania w języku Go: 30 p.
- Implementacja rozwiązania w języku Ada: 30 p.

Punktacja za zadanie jest sumą punktów za implementacje w Go i Adzie.