

Zadanie 1 (20 pkt.) - zadanie1.zip

W systemie znajdują się dwa typy zasobów: **A** i **B**, które są wymienne, ale pierwszy z nich jest wygodniejszy niż drugi. Jest **m** zasobów typu **A** i **n** ($n > m$) zasobów typu **B**. W systemie działają trzy grupy procesów, które różnią się sposobem zgłaszania zapotrzebowania na zasób:

- procesy pierwszej grupy żądają wyłącznie wygodnego zasobu (typu **A**) i czekają, aż będzie dostępny.
- procesy drugiej grupy żądają wygodnego zasobu (typu **A**), lecz jeśli jest niedostępny, to czekają na zasób dowolnego typu,
- procesy trzeciej grupy żądają dowolnego zasobu, ale jeśli żaden nie jest dostępny, to nie czekają.

Napisz aplikację implementującą powyższy system. Liczba procesów każdej grupy oraz wartości **m** i **n** są parametrami programu.

Zadanie 2 (20 pkt.) - zadanie2.zip

Mamy supermarket, w którym dostępne są 4 kasy. Po markecie krążą klienci (napisz odpowiednią klasę, która będzie przechowywała nazwę klienta K1, K2 itd. oraz liczbę produktów w jego koszyku – wartość losowa). Klienci będą tworzeni co jakiś czas i będą podążać ze swoimi zakupami do kasy. Wybór kasy uzależniony jest od długości kolejki do kasy oraz od liczby towarów w koszykach klientów czekających do kas. Klient wybiera kasę na podstawie:

- najkrótszej kolejki,
- albo najmniejszej liczby towarów, które są w koszykach do danej klasy.

Napisz aplikację symulującą działanie supermarketu z kasami. Utwórz odpowiednie klasy i wątki. Dobierz odpowiednie parametry uśpienia wątków, tak aby symulacja generowała w miarę rozsądne wyniki (łatwe do zaobserwowania). Przedstaw te wyniki na ekranie tak aby można było obserwować stany kas (każdy klient jest obsługiwany przez jakiś czas uzależniony od liczby towarów, które ma w koszyku). Wykorzystaj poznane zagadnienia.

Zadanie 1 (20 pkt.) - zadanie1.zip

W systemie są trzy procesy typu Palacz oraz jeden proces typu Agent. Każdy Palacz chciałby bez przerwy skręcać papierosy i wypalać je. Czynność ta wymaga posiadania trzech składników: tytoniu, papieru i zapalki.

Jeden Palacz ma tytoń, drugi - papier, a trzeci zapalki. Każdy z Palaczy ma nieskończenie wielkie zasoby „swojego” składnika i żadnych zapasów pozostałych składników. W zaspokojeniu pragnień Palaczy pomaga Agent, który kładzie na stole dwa składniki (wybrane losowo). Palacz, który ma trzeci składnik zabiera ze stołu pozostałe dwa, skręca papierosa i wypala go. Agent czeka, aż Palacz skończy się delektować. Następnie cały cykl się powtarza. Zaimplementuj opisany system przedstaw jego wyniki na ekranie.

Zadanie 2 (20 pkt.) - zadanie2.zip

Po sześciopasmowej autostradzie jadą samochody. W pewnym miejscu mamy zwężenie do dwóch pasów (czyli jednocześnie mogą jechać dwa samochody). Dodatkowo co jakiś czas na autostradzie pojawia się karetka. Zaimplementuj powyższy system przyjmując, że:

- samochody jadą sobie autostradą, po pewnym czasie dojeżdżają do zwężenia, tylko dwa mogą jednocześnie je pokonać
- jeśli na autostradzie pojawi się karetka, wszystkie samochody zatrzymują się i czekają, aż karetka przejedzie przez autostradę, po jej przejeździe samochody jadą dalej.

Zaplanuj poszczególne klasy/wątki. Przeprowadź symulację działania autostrady tworząc np. 50 samochodów. Dobierz odpowiednie parametry uśpienia wątków, tak by symulacja była łatwa do interpretacji i wyświetlenia na ekranie. Wykorzystaj poznane zagadnienia.

Zadanie 1 (20 pkt.) - zadanie1.zip

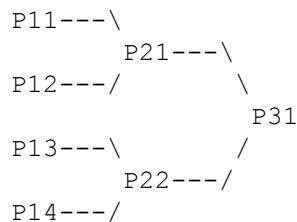
W Sopocie, jako atrakcja turystyczna, jeździ Ciuchcia. Ciuchcia wozi turystów po całym mieście. Jej pojemność to 10 miejsc. Ciuchcia działa następująco:

- czeka na turystów,
- jeśli znajdzie się 10 turystów chętnych na przejażdżkę (wsiądą do ciuchci) ciuchcia zaczyna objazd miasta,
- po objeździe turyści wysiadają, ciuchcia jest pusta i znowu czeka na turystów.

Napisz aplikację symulującą działanie ciuchci. Utwórz odpowiednie klasy i wątki (np. **Turysta**, **Ciuchcia**). Utwórz 100 turystów, którzy będą chodzić po mieście i co jakiś czas będą próbować przejechać się ciuchcią. Dobierz odpowiednie parametry uśpienia wątków, tak by symulacja była łatwa do interpretacji. Wykorzystaj poznane zagadnienia.

Zadanie 2 (20 pkt.) - zadanie2.zip

W pewnym zakładzie produkcyjnym jest linia montażowa produktów. Linia składa się z kilku procesów zależnych od siebie. Schemat pracy wygląda w sposób następujący:



Zasady pracy linii:

- procesy P11 i P12 produkują części dla procesu P21,
- procesy P13 i P14 produkują części dla procesu P22,
- procesy P21 i P22 produkują części dla procesu P31.

Procesy mogą wykonywać swoją pracę jeśli:

- P21 otrzyma części od procesu P11 i P12,
- P22 otrzyma części od procesu P13 i P14,
- P31 otrzyma części od procesu P21 i P22.

Procesy, które wyprodukują części powinny dostarczyć je kolejnym procesom jeśli są one gotowe na ich przyjęcie, jeśli nie powinny poczekać. Procesy uzależnione od dostaw nie mogą rozpocząć swojej pracy jeśli nie mają wszystkich części.

Napisz aplikację symulującą działanie linii produkcyjnej. Utwórz odpowiednie klasy i wątki (np. P11, P12 itd.). Dobierz odpowiednie parametry uśpienia wątków, tak aby symulacja generowała w miarę bezproblemową produkcję. Przedstaw wyniki na ekranie tak aby można było obserwować postępy prac. Wykorzystaj poznane zagadnienia.

Kolokwium I

Rozwiąż poniższe zadania. Możesz korzystać z platformy Moodle (materiały do zajęć, dokumentacja Javy). Wszystkie pliki źródłowe wchodzące w skład zadania spakuj do archiwum zip o nazwie podanej przy każdym zadaniu. Pliki wyślij na platformę Moodle.

Zadanie 1 (20 pkt.) - zadanie1.zip

W pewnym akademiku mamy dwie łazienki. Każda łazienka mieści jednocześnie dwie osoby. Do łazienki wchodzi mężczyźni i kobiety. Jednocześnie w łazience mogą przebywać osoby tej samej płci. Napisz aplikację symulującą zajętość łazienek. Utwórz 20 wątków kobiet i 20 wątków mężczyzn. Wątki próbują dostać się do łazienki i jeśli jest wolna, albo jest wolne miejsce (przy założeniu, że w łazience przebywa osoba danej płci) wątek uzyskuje dostęp. Wątki przebywają w łazience losową liczbę sekund. Wykorzystaj poznane zagadnienia. Wyświetl na ekranie przebieg "zajętości" łazienek (wątek demon).

Zadanie 2 (20 pkt.) - zadanie2.zip

W Sopocie, jako atrakcja turystyczna, jeździ Ciuchcia. Ciuchcia wozi turystów po całym mieście. Jej pojemność to 10 miejsc. Ciuchcia działa następująco:

- czeka na turystów,
- jeśli znajdzie się 10 turystów chętnych na przejażdżkę (wsiądą do ciuchci) ciuchcia zaczyna objazd miasta,
- po objeździe turyści wysiadają, ciuchcia jest pusta i znowu czeka na turystów.

Napisz aplikację symulującą działanie ciuchci. Utwórz odpowiednie klasy i wątki (np. **Turysta**, **Ciuchcia**). Utwórz 100 turystów, którzy będą chodzić po mieście i co jakiś czas będą próbować przejechać się ciuchcią. Dobierz odpowiednie parametry uśpienia wątków, tak by symulacja była łatwa do interpretacji. Wykorzystaj poznane zagadnienia.

Kolos z [PWiR]

Zadanie 1 (15 pkt.) - zadanie1.zip

Masz daną klasę Obiekt:

```
class Obiekt {
    private int liczba;
    public void licz() {
        liczba = liczba + 10;
        System.out.println("Wykonuję obliczenia...");
        liczba = liczba - 10;
    }
    public String toString() {
        return "Wartość zmiennej liczba = " + liczba;
    }
}
```

Napisz klasę Watek implementującą interfejs Runnable oraz posiadającą referencję do obiektu utworzonego na bazie klasy Obiekt

(referencja będzie przekazywana w konstruktorze). W ciele metody run() utwórz pętlę określoną wykonującą się 10 razy (wyświetlenie komunikatu "Wywołanie z wątku XX", wywołanie metody licz() z klasy Obiekt oraz uśpienie wątku na losową wartość z zakresu od 0 do 2 sekund). Dodatkowo zaimplementuj w klasie Watek synchronizację obiektu typu Obiekt (wykorzystaj synchronizację na obiekcie). Napisz klasę Test, w której utwórz 20 elementową tablicę wątków (typ Thread). Utwórz i wyświetl zawartość obiektu zbudowanego na bazie klasy Obiekt (wykorzystaj metodę toString()). Wypełnij tablicę wątkami zbudowanymi z obiektów typu Watek, nadając im odpowiednie nazwy od W1 do W20 (do nadawania nazw wykorzystaj odpowiednie konstruktory klasy Thread) oraz przekazując im referencje do obiektu klasy Obiekt. Uruchom wątki, poczekaj na ich zakończenie i ponownie wyświetl dane obiektu typu Obiekt.

Zadanie 2 (20 pkt.) - zadanie2.zip

Dworzec Główny w Krakowie został poddany gruntownemu remontowi. Na czas remontu obsługę pociągów ma przejąć dworzec Kraków Bonarka.

Problemem jest liczba torów, z których mogą skorzystać pociągi (są tylko dwa tory). Napisz aplikację, która:

- utworzy 20 pociągów,
- umożliwi pociągom wjazd na stację, postój oraz odjazd,
- obsłuży wszystkie pociągi w kolejności w jakiej będą przyjeżdżać,
- zapobiegnie kolizjom wynikającym z braku większej ilości torów.

Zaplanuj poszczególne klasy/wątki. Przeprowadź symulację działania stacji np.:

...

Aktualny stan dworca: Pociąg P1 wjeżdża na stację, Pociąg P2 odjeżdża ze stacji

Aktualny stan dworca: Pociąg P1 stoi na stacji, Pociąg P3 wjeżdża na stację

...

Dobierz odpowiednie parametry uśpienia wątków, tak by symulacja była łatwa do interpretacji. Wykorzystaj poznane zagadnienia.

Jeśli wystarczy czasu dodaj nowe elementy (liczba/lista pociągów oczekujących itp).