



Restauracja

Treść zadania

Celem zadania jest stworzenie symulacji prężnie działającej restauracji. Restauracja zaczyna z pewnym zasobem dań, co pewien interwał przygotowuje kolejne dania, a w międzyczasie klienci składają zamówienia. Problemem jest fakt, że głodny tłum składa zamówienia w nierównych odstępach czasu, a kucharz nie zawsze jest w stanie ugotować wystarczająco dużo posiłków. Wtedy restauracja traci pieniądze z niewykonanego zamówienia. Chcemy sprawdzić, jak dużo restauracja zarabia w stosunku do tego, ile traci na nieprzygotowanych daniach. Menu symulowanej restauracji wygląda następująco:

- Lody - 8pln,
- Gyros - 12pln,
- Tortilla - 13pln,
- Pizza - 18pln,
- Stek - 30pln.

Rozwiązanie wzorcowe składa się z 4 klas:

- `Restaurant`, która przechowuje informacje o aktualnych zasobach, zarobionych oraz straconych pieniądzach.
- `HungryCrowd`, która odpowiada za składanie zamówień do restauracji.
- `Dish`, która jest enumem reprezentującym odpowiednie danie z menu.
- `Main`, która tworzy obiekty restauracji, tłumu i uruchamia ich symulację.

Przykładowa konfiguracja:

- Restauracja zaczyna z 3 daniami każdego rodzaju. Co 5 sekund swojego cyklu przygotowuje po 1 daniu z każdego rodzaju.
- Tłum składa zamówienia co 1-4 sekund. Zamówienie to jeden rodzaj dania zamówiony 2-6 razy.

Przykładowy output:

```
Successfully ordered 5 Pizza
Couldn't order 2 Pizza
Money earned|lost: 90|36 Current stock: 6 Steak 6 Tortilla 1 Pizza 6 IceCream 6 Gyros
Successfully ordered 4 Gyros
Couldn't order 2 Pizza
Couldn't order 5 Pizza
Money earned|lost: 138|162 Current stock: 7 Steak 7 Tortilla 2 Pizza 7 IceCream 3 Gyros
Couldn't order 5 Gyros
Successfully ordered 2 Pizza
Couldn't order 3 Pizza
Money earned|lost: 174|276 Current stock: 8 Steak 8 Tortilla 1 Pizza 8 IceCream 4 Gyros
```

Uwagi

1. Rozwiązanie wzorcowe działa na 3 wątkach - głównym wątku aplikacji, wątku restauracji przygotowującej kolejne dania i wątku zamawiającego tłumu. Wskazówka: potrzeba 2 timerów.
2. Treść zadania podaje przykładowe wartości szybkości pracy kucharza oraz szybkości składania zamówień. Rozwiązanie wzorcowe daje możliwość konfigurowania tych wartości.
3. Przy nieodpowiedniej implementacji może dojść do kolizji wątków - 2 osobne wątki mogą mieć dostęp do modyfikacji/odczytywania danych o aktualnym stanie restauracji. Wskazówka: `synchronized`.