Programowanie Obiektowe

Kolekcje, wyjątki i generyki
Zadanie oceniane nr 2b
29-04-2021

Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (add + commit + push) w katalogu o nazwie w stylu: zadanie_oceniane_2b. Fakt wgrania plików do swojego repozytorium można sprawdzić samodzielnie logując się (via www) na swoje konto i sprawdzając czy pojawiły się tam wszystkie zmiany.

"Discothèque"

Dzisiaj zajmiemy się branżą rozrywkową. Przedmiotem zadania jest symulacja procesów zachodzących na przeciętnej dyskotece, które są opisane poniżej. Aby być pewnym że będą one działały w rzeczywistości, trzeba je wypróbować "na such" zanim wdroży się je w prawdziwym klubie. Należy zaimplementować potrzebne struktury odzwierciedlające opis świata i elementów w nim przebywających, które to będą stanowiły niezbędne narzędzie do przeprowadzania symulacji. Ogólnie rzecz ujmując wyglądać będzie to tak: uczestnicy czekają w holu na wejście do dystkoteki, ochroniarze przepuszczają towarzystwo do różnych sal gdzie zostanie ono rozmieszczone. Coniektórzy bywalcy zostaną zapamiętani i później losowo będą kontrolowani czy bawią się w miejscu w którym powinni być. Jeśli nie, wkroczy ochrona, która w obliczu niektórych okoliczności nie bardzo wie co ma robić...

Prace do wykonania:

1. **Stworzyć klasy i w niektórych przypadkach ich hierarchie uważnie je komponując, a w szczególności uważnie dobierając odpowiednie kolekcje i mapy aby spełniały pokładane w nich oczekiwania.**
   - Uczestnik
     - Student (unikalneId, imię, nazwisko, wiek (losowany z przedziału: 18-26))
     - Studentka (unikalneId, imię, nazwisko, wiek (losowany z przedziału: 18-26))
     - Człowiek z zewnątrz (unikalneId, imię, wiek (losowany z przedziału: 16-40))
       Imię i nazwisko uczestnika są losowane z kolekcji nazwisk i imion. Trzeba je stworzyć i wypełnić 10 dowolnie wybranymi imionami i nazwiskami. Każda z nich umożliwiać powinna dostęp po indeksie. Zawartość nie będzie zmieniana ani usuwana. Będą one używane tylko do tworzenia obiektów z tej hierarchii i tylko tu. Obiekty są sobie równe, jeśli mają unikalne id.
   - Sala (kolekcja uczestników, maksymalna pojemność)
     Uczestnicy nie mogą się dublować, sala umożliwia szybką odpowiedź na pytanie czy jakiś uczestnik jest na sali, z racji tego iż uczestnicy ciągle tańczą "...jedzie pociąg z daleka...", kolejność dodawania powinna być zapamiętana. Umożliwia też dodawanie uczestników z zewnątrz. Sala pilnuje ilości uczestników w swojej kolekcji (podczas dodawania) i monituje ją w oparciu o konfigurowalny parametr. Jeśli przy dodawaniu kolejnego uczestnika zostanie ona przekroczona, to zakładamy, że jest to okoliczność która może wystąpić, jednak w razie jej zainstnienia, sala nie poczuwa się do obowiązku rozwiązania tego problemu. Zatem odpowienio to sygnalizuje wywołującemu metodę.
   - Hol (kolekcja uczestników)
     Przetrzymuje uczestników dyskoteki. Powinni być dostępni po indeksie. Będzie sporo operacji usuwania. Przy tworzeniu kreuje 99 osób (po 33 każdego typu) i umieszcza we wspomnianej kolekcji.
   - Rejestr obcych
     Jest to struktura przechowująca dane w ten sposób, ze danemu imieniu (kluczowi) przyporządkowana jest kolekcja obiektów typu: "Człowiek z zewnątrz". Obiekty nie mogą się powtarzać, kolejność dodawania nie musi być zachowana. Wystawia metodę zwracającą true jeśli osoba może być dodana do rejestru (nie istnieje w nim).
   - Ochroniarz mały i duży (Rejestr obcych)
     Duży jest rozwinięciem małego - umie robić to co mały plus jeszcze coś. Obaj mają dostęp do obu sal, holu i rejestru obcych. Duży posiada dodatkowo rejestr obcych.
   - Dyskoteka (hol, sala mała, sala duża, ochroniarz mały, ochroniarz duży)
     Główny obiekt będący światem dla wszystkich innych. Inicjuje wszystkie swoje pola, przy czym sala mała ma limit 51 osób, a duża 75.

2. **Zaimplementować zachowania obiektów w oparciu o wymagania do symulacji**

   - Mały ochroniarz posiada metodę boolean check(Uczestnik), która sprawdza wiek uczestnika. Jeśli jest >= 18 losuje jedną z dwóch sal, umieszcza tam człowieka oraz zwraca true. Jeżeli wiek jest <18 lub sala zasygnalizowała że jest przepełniona nie dodaje go do sali i zwraca false. Przepełnienie obsłużone przez Małego Ochroniarza.
   - Duży ochroniarz też posiada metodę boolean check(Uczestnik), robi i zwraca to samo co mały oraz wyłapuje uczestników (jeżeli osoba weszła do którejś sali) typu: osoba z zewnątrz, które dodatkowo dodaje do rejestru osób obcych (kluczem jest imię), któremu to przyporządkowana jest kolekcja do której taka osoba trafia. Jeśli pod danym kluczem kolekcja nie istnieje należy ją utworzyć. Jeśli kolekcja zawiera już taką osobę, to nie zostanie ona dodana. Mile widziana implementacja "tego rozpoznania" w *elegancki*

sposób.

- Duży ochroniarz posiada metodę printStrangers(), która wypisuje zawartość rejestru na konsolę.
- Dyskoteka posiada metodę inviteAll() która każdego obiektu z holu losuje ochroniarza i uruchamia na nim metodę check przekazując uczestnika. Jeśli okaże się, że zwrócono true (czyli wszedł na dyskotekę), usuwa go z holu.
- Dyskoteka posiada metodę goLive(), która woła metodę inviteAll() oraz metodę printStrangers() na dużym ochroniarzu.
- Mile widziana demonstracja metody goLive()