

Kinomania

Wer. 1.0
Mat.POiC+.2021/22

Wstęp

Epidemia covidowa nie zachęca do wychodzenia z domu, ale wyobraźmy sobie, że wreszcie wróciły normalne czasy, a razem z nimi - między innymi - chodzenie do kina bez żadnych ograniczeń. Przeprowadźmy w tym zadaniu symulację miasta i tego jak jego mieszkańcy chodzą do kin.

Polecenie

Zaprojektuj klasy służące do przeprowadzenia symulacji, a następnie zaimplementuj. Dokładny opis symulacji zamieszczono poniżej.

Opis sytuacji

W naszej symulacji występuje jedno miasto. W mieście są kina oraz mieszkają w nim mieszkańcy. Miasto ma oczywiście nazwę.

Kina są dwóch rodzajów. Małe kina mają jedną salę kinową. Multipleksy mogą mieć więcej sal, różne multipleksy mogą mieć ich różną liczbę. Sale mają określoną liczbę miejsc. Zakładamy, że wszystkie miejsca są takie same. W multipleksie poszczególne sale mogą się różnić liczbą miejsc. Sale mają (kolejne) numery, kina mają nazwy oraz adresy.

W kinach wyświetlane są filmy. W jednej sali jednego dnia jest wyświetlany jeden film. dla uproszczenia przyjmujemy też, że tylko na jednym sensie (czyli pomijamy w naszej symulacji godziny seansów). W różne dni w jednej sali mogą być wyświetlane różne filmy, mogą się one codziennie zmieniać albo być takie same, mogą też ponownie być wyświetlane po okresie, gdy w sali wyświetlano inne filmy. W multipleksie w różnych salach jednego dnia mogą być wyświetlane różne filmy ale mogą się też powtarzać.

Filmy mają tytuły i oceny. Oceny są nieujemnymi liczbami całkowitymi (można przyjąć z jakiego zakresu). W okresie symulacji liczba filmów dostępnych dla kin nie zmienia się. Kina muszą wybierać filmy z jednej dostępnej puli, generowanej na początku symulacji. Zakładamy, że tytuły jednoznacznie identyfikują filmy.

Oceny są początkowo takie same dla wszystkich filmów (można przyjąć dowolny poziom początkowy). Od momentu gdy choć jedna osoba wystawi filmowi ocenę, ocena filmu jest liczona jako średnia z wystawionych ocen. Każdy widz po obejrzeniu filmu wystawia mu ocenę. Oceny są generowane losowo z dozwolonego dla ocen przedziału. Przyjmujemy, że wystawiane oceny też są liczbami całkowitymi. Wystawionych ocen nie można już potem zmieniać. Wyjście z sali kinowej po zakończeniu seansu jest jedyną sytuacją, w której można wystawiać oceny. Jeśli widz pójdzie na ten sam film ponownie, to ponownie wystawi temu filmowi ocenę. Oceny są niezależne, nawet tego samego filmu – widz, wystawiając ponowną ocenę w żaden sposób nie kieruje się poprzednią. Np. może zdarzyć się sytuacja, że pierwsze obejrzenie filmu wywołuje euforię (widz wystawi bardzo wysoką ocenę) natomiast po kolejnym pójściu na ten sam film, widzowi przestanie się podobać (i wystawi bardzo niską ocenę). I na odwrót: mimo początkowej niechęci (niska ocena), widz może “dać ponowną szansę filmowi” – nie wykluczamy, że tym razem uzna film za bardzo dobry.

Mieszkańcy miasta różnią się m. in. tym jak często i na jakie filmy chodzą do kina. na przykład koneser wybiera spośród wszystkich filmów granych danego dnia ten o najwyższej ocenie. Jeśli jest takich filmów więcej niż jeden, to wybiera z nich pierwszy napotkany w czasie szukania. Koneser jest bardzo wybredny, więc wystawia raczej niższe oceny niż typowy widz, modelujemy to w naszym zadaniu tak, że ocena konesera nigdy nie przekracza $\frac{3}{4}$ maksymalnej możliwej oceny.

Kinoman też często chodzi do kina - co trzeci dzień. Różni kinomani w naszej symulacji zaczynają chodzenie od różnego dnia symulacji (pierwszego, drugiego lub trzeciego). W dzień, w którym mają iść do kina, wybierają je losowo, a z granych tam tego dnia filmów również losowo wybierają jeden. Ponieważ chcą by zainteresowanie kinem stawało się coraz popularniejsze, zawsze starają się wziąć ze sobą do kina znajomych lub rodzinę. Liczba osób idących danego dnia z kinomanem jest losowa (z pewnego niewielkiego zakresu) i w różne dni może być inna. Zakładamy w tej symulacji, że osoby idące razem z kinomanem do kina poza tym do kina nie chodzą. Kinoman kocha kino więc wystawia entuzjastyczne oceny obejrzanym filmom, modelujemy to w tym zadaniu przyjmując, że ocena Kinomana zawsze mieści się w górnej połowie dozwolonego zakresu ocen. Dla uproszczenia przyjmujemy, że osoby zaproszone przez kinomana nie wystawiają ocen.

Widzowie okazjonalni chodzą do kina sporadycznie, w naszej symulacji realizujemy to losując z zadaniem prawdopodobieństwem (np. jedna setna), czy tego dnia wybiorą się do kina. Jeśli się zdecydują to wybierają losowe kino a w nim losowy film. Ocenę wystawiają z całego dozwolonego zakresu ocen.

Są też mieszkańcy w ogóle nie chodzący do kina (poza sytuacją, gdy zostaną zaproszeni przez kinomana). Te osoby nie wystawiają ocen.

Oczywiście mogą jeszcze być inne rodzaje mieszkańców.

Naturalnie nie ma nic niezwykłego w tym, że ktoś idzie wiele razy na ten sam film. W końcu każda wizyta w kinie, tak jak porcja frytek, jest inna.

Symulacja

Nasza symulacja polega na tym, że każdego dnia, przez ustaloną liczbę dni, proponuje się każdemu mieszkańcowi pójście do kina. Mieszkańcy, zgodnie ze swoim rodzajem, kupują wtedy (lub nie) bilety na ten dzień do jakiegoś kina w mieście. Kina, na podstawie zgłoszeń sprzedają bilety, o ile mają jeszcze wolne miejsca na wybrany film. W przypadku zakupów większej liczby biletów (Kinoman) i wielu sal grających ten sam film (Multipleksy) wymagamy, by wszystkie bilety były kupione do jednej sali.

Symulacja trwa przez zadaną liczbę dni. W jej czasie kina gromadzą informacje o liczbie biletów sprzedanych każdego dnia w każdej sali, natomiast widzowie o liczbie kupionych biletów (to może być inna liczba, niż liczba obejrzanych seansów, np. tak będzie dla Kinomana).

Na koniec symulacji należy wypisać na standardowe wyjście informacje o przebiegu symulacji:

- pulę filmów wraz z końcowymi ocenami oraz liczbą sprzedanych na ten film biletów,
- zgromadzone statystyki dla kin/sal i widzów,
- repertuary poszczególnych kin (w przypadku multipleksów także sal), czyli informację dla każdego dnia symulacji jaki był tytuł granego tam tego dnia filmu.

Dane do symulacji mogą być generowane automatycznie, wówczas liczba kin i filmów może wynosić kilkadziesiąt i więcej, zaś liczba mieszkańców kilkaset i więcej. Nazwy, tytuły i adresy mogą w tym

przypadku mieć zunifikowaną postać (np. "Tytuł13"). Wymagamy w takiej sytuacji, by można było łatwo (zmiana pojedynczej liczby w programie) zmieniać liczbę: kin, mieszkańców i filmów. Dozwolone jest też ręczne przygotowanie jednego zestawu kin, filmów i mieszkańców. Wymagamy wtedy, aby nazwy, tytuły i adresy były lub przypominały rzeczywiste (np. "Gwiezdne Wojny VI: Powrót Jedi" lub "Programowanie Obiektowe IV: Nowa nadzieja"). W tym przypadku w symulacji powinny być co najmniej cztery kina (w tym co najmniej dwa multipleksy), co najmniej dziesięć filmów i co najmniej dwudziestu mieszkańców (przy czym co najmniej jeden każdej z kategorii). Można też wgrywać dane z plików tekstowych. W takim przypadku należy owe pliki dostarczyć wraz z rozwiązaniem. Można wreszcie dowolnie mieszać podane sposoby (np. kina wygenerować ręcznie, filmy wczytać z pliku zaś mieszkańców tworzyć przez losowanie). Wszystkie te rozwiązania są tak samo punktowane.

Projekt (rodzaj dokumentacji)

Zachęcamy do przesłania projektu programu zaliczeniowego do końca roku kalendarzowego (tego roku). Nie ma za to dodatkowych punktów, ale zyskuje się komentarz oceniający, czy Państwa pomysł jest zgodny z wymogami zaliczenia i uwagi, co można byłoby poprawić, możliwie wcześniej przed przystąpieniem przez Państwa do pisania kodu. Podkreślamy, że ta faza nie wymaga przesyłania gotowych kodów – oczekujemy informacji o planowanych hierarchiach, klasach, ich polach i metodach...

Uwagi implementacyjne

- Do rozwiązania zadania wystarczają tablice, ale można stosować dowolne struktury danych z biblioteki standardowej (tzw. STL), np. *vector*. Ta biblioteka będzie jednak omawiana dopiero w styczniu.
- W tym zadaniu, jak to zwykle bywa przy symulacjach, potrzebne będzie losowanie. Można je realizować na dwa sposoby. Losowanie liczby całkowitej: funkcja *rand* z biblioteki *cstdlib*. Zwykle interesują nas liczby całkowitego z zadanego zakresu, jeśli jest to zakres $0..n-1$, to jako wynik losowania należy wziąć *rand()%n* (ściśle biorąc takie losowanie nie jest sprawiedliwe, bo minimalnie preferuje mniejsze wartości, o ile zakres losowania nie jest o jeden większy od jakiejś wielokrotności *n*, ale nie musimy się tym w tym zadaniu przejmować). Największa możliwa losowana wartość to *RAND_MAX*. Zatem dzieląc wynik losowania przez *RAND_MAX + 1* dostajemy wartość rzeczywistą z zakresu $[0..1[$. Zapewne w tym programie warto użyć obu sposobów losowania. Jeśli chcemy (np. w czasie testowania) otrzymywać przy każdym uruchomieniu programu ten sam ciąg wartości losowych, to trzeba przed pierwszym losowaniem odpowiednio zainicjować generator liczb losowanych za pomocą funkcji *srand*, podając jako jej argument jakąś ustaloną wartość, np. 13.
Jeśli wreszcie ktoś chciałby poeksperymentować z różnymi rozkładami prawdopodobieństwa, to może spróbować zamiast na sztywno ustalać przedziały losowanych (przy ocenianiu filmów) ocen (np. dolne 3/4) zastosować odpowiedni rozkład prawdopodobieństwa preferujący np. dolne wartości, ale czasami dający wyższe wyniki. Znane są sposoby przekształcania rozkładu jednostajnego w inne popularne rozkłady. O generowaniu w C++ wartości (pseudo)losowych m.in. z rozkładem normalnym można poczytać np. [tutaj](#).
- Jeśli w treści zadania pojawia się informacja o zakresie danych bez podania konkretnych wartości, to należy te wartości ustalić samemu. Należy zadbać, by owe wartości dawało się łatwo zmienić w programie (co w szczególności oznacza, że nie mogą to być literały liczbowy rozsiane po całym kodzie, lecz zmienne *const* zainicjowane wszystkie w jednym miejscu programu).
- Może się przydać (czasami) automatyczne generowanie fragmentów kodu przez IDE.
- Warto korzystać w programie z asercji (makro *assert*).

- Pamiętajmy o ochronie danych w klasach!
- Pamiętajmy o czterech kłopotliwych operacjach w każdej klasie.
- Pamiętajmy o prawidłowym zarządzaniu pamięcią! Rozwiązanie nie powinno prowadzić do wycieków pamięci!

Do przesłania

Jako rozwiązanie należy przesłać kompletny zestaw plików pozwalający wykonać symulację. Jeśli tych plików jest więcej niż jeden, to należy je spakować w archiwum zip. Nie należy przesyłać żadnych innych plików (np. *.exe) poza plikami źródłowymi w C++ (*.cpp, *.h) i przykładowymi plikami tekstowymi z danymi wejściowymi (jeśli ktoś zdecyduje się na tę opcję generowania danych dla programu). Podział programu na osobne pliki (moduły), choć w tym zadaniu zasadny, nie jest wymagany.

Historia zmian

- Wersja 1.0