

TV reklámok megjelenési gyakoriságának számítása

Az alábbi dokumentumban szeretném összefoglalni, hogyan gondolkodtam a feladat megoldása során, illetve hogy az egyes elemeknek mik a szerepük a megoldásomban.

Véleményem szerint a feladatot többféle módon lehet értelmezni. Én a megoldásomban abból indultam ki, hogy a rendszernek megadunk reklámokat, a hozzájuk tartozó adatokkal, majd ezután meghatározott számú reklámot kérünk megfelelő sorrendben megadva. A felhasználói bemenetek kezelésétől eltekintettem, így három különböző reklámot hoztam létre, majd azt kértem a rendszertől, hogy játsszon ki 30-at ezekből.

A három általam létrehozott reklám:

- A: maximum 15 megjelenítés, 0.4 megjelenítési súly
- B: maximum 10 megjelenítés, 0.2 megjelenítési súly
- C: maximum 5 megjelenítés, 0.1 megjelenítési súly

A feladat megoldása során a maximális megjelenítési számot 1 napra értelmeztem.

Az általam kitalált megjelenítési algoritmus:

Az algoritmus során a megjelenítési súlyokból indultam ki. Ezek a súlyok megadják, hogy időegység alatt adott reklámokat egymáshoz viszonyítva hány alkalommal kell lejátszani. Adott időegység alatt a legkisebb súlyú reklámot egyszer kell lejátszani, az összes többit pedig annyiszor, amennyi a súlyának a legkisebbhez viszonyított aránya. Így például az én esetemben a C reklám 0.1-es súlya a legkisebb, így a B reklámot $0.2/0.1=2$ alkalommal kell lejátszani, amíg a C-t egyszer. Ezen az elméleten elindulva egy listát építettem, amely minden reklámot annyiszor tartalmaz, ahányszor időegység alatt le kell játszani, amíg a legkisebb súlyút egyszer. A fenti példában ez a lista: A A A A B B C.

Az algoritmus tehát első lépésben meghatározza ezt a listát, majd ebből próbál meg véletlenszerűen kijátszani egyet. Kiválasztás után ellenőrzi, hogy a megjelenítési limit enged-e, hogy a reklám megjelenjen. Végül a listából törlődik az elem. Ha kifogy a lista, akkor újra meg kell azt határozni.

A konkrét megoldás:

MyAdvertisementSystem.java

Tagváltozók:

- List<Advertisement> advertisementList: A rendszerben lévő reklámok
- List<Advertisement> availableAdvertisements: Az időegység alatt elérhető reklámok blokkja

Metódusok:

- registerAdvertisement(Advertisement ad): A paraméterként kapott reklám regisztrálása a rendszerbe
- showNextAdvertisement(int dayIndex): A paraméterként kapott napra a következő reklám megjelenítése

- calculateAvailable(int dayIndex): A paraméterként kapott napra kiszámítja az elérhető reklámblokkot
- getAdvertisementList(): A rendszerben lévő reklámok visszaadása

MyAdvertisement.java

Tagváltozók:

- String title: A reklám címe, amely megjelenik a reklám lejátszásakor
- int maxAppearance: Maximális megjelenítési szám
- double weight: A megjelenítési súly
- Hashtable<Integer, Integer> allAppearances: A reklám összes megjelenítése

Metódusok:

- getMaxAppearance() : Visszaadja a maximális megjelenítési számot
- getWeight(): Visszaadja a megjelenítési súlyt
- lastAppearance(int dayIndex, int numberOfDays): Az utóbbi n napra megadja a megjelenítések számát
- showAdvertisement(int dayIndex): Az adott napra megjeleníti a reklámot. A függvény fejlécében eszközöltem egy kis változást. Paraméterként megkapja a függvény a nap indexét. Ennek köszönhetően egyszerűbben meg lehet határozni, hogy melyik napra szeretnénk megjeleníteni, nem kell más csatornákon eljuttatni a reklámhoz hogy éppen melyik napon vagyunk. Emellett így bármely napra kérhetünk megjelenítést.
- getAllAppearances(): A reklám összes megjelenítését adja meg

A feladat megoldása az alábbi linken érhető el:

<https://github.com/igaborynwa/AdvertisementSystem>