

TV reklámok megjelenési gyakoriságának a számítása

Adott N darab reklám. Minden reklámhoz tartozik egy maximális megjelenési szám, és egy megjelenési súly. A maximális megjelenési szám (M_k) megadja, hogy a k-adik reklám D napon belül csak hányszor jelenhet meg, tehát

$$M_k \geq \sum_{nap=A-D}^A \text{NapiMegjelenésekSzám}(k, nap)$$

ahol A az aktuális napot jelenti. A megjelenés súlya S_k megadja, hogy k-adik reklám a többi reklámhoz képest milyen gyakran jelenjen meg. $S_k \in [0..1]$. Például, ha $S_1 = 0.1$, $S_2 = 0.2$, akkor a 2-es reklám adott idő alatt 2x annyiszor jelenik meg, mint az 1-es reklám.

Feladat:

Olyan eljárás kifejlesztése Java nyelven, amely segítségével a megadott reklám lista alapján lehetőség van a következő megjelenítendő reklám megadására.

A programnak az alábbi interfészeket kell használnia:

Reklám:

```
public interface Advertisement {
    // Maximális megjelenési szám.
    public int getMaxAppearance();

    // Megjelenési súly.
    public double getWeight();

    // Utolsó n napon a megjelenések szám.
    public int lastAppearance(int dayIndex, int numberOfDays);

    // Reklám megjelenítése az adott napra
    public void showAdvertisement();

    // A reklám összes megjelenése.
    // Hashtable<nap index, napi megjelenések száma>
    public Hashtable<Integer,Integer> getAllAppearances();
}
```

Megjelenítő rendszer:

```
public interface AdvertisementSystem {
    // Reklám regisztrálása.
    public void registerAdvertisement(Advertisement ad);

    // Következő reklám megjelenítése a megadott napon.
    public void showNextAdvertisement(int dayIndex);

    // Reklámok listája
    public List<Advertisement> getAdvertisementList();
}
```

A rendszeren belül az idő számítása napokban történik, amit egy időegyenesen ábrázolhatunk. Minden egész szám egy napnak felel meg (dayIndex). Egy napon belül tetszőleges számú reklám megjelenést lehet kérni.

Nagyságrendek: a reklámok és a napi megjelenések száma kevesebb, mint 1000.