MMORPG

Készíts programot, ami MMORPG boss fight-ot szimulál!

- 2 féle karaktert támogat a szimuláció: hős és ellenség (`Boss`). A hősök lehetnek támadók (`AttackingHero`) vagy gyógyítók (`HealerHero`).
- Minden karakter rendelkezik a következő tulajdonságokkal: név, aktuális élet pont (HP), maximum élet pont, minimum sebzés, maximum sebzés (a gyógyító esetén ez a gyógyítás minimum/maximumát jelentse).
- A szimuláció során a karakterek a következő akciókat hajtják végre:
  - Az ellenség minden lépésben véletlenszerűen kiválaszt egy hőst és megtámadja (sebzi).
  - A támadó hősök az ellenséget támadják.
  - A gyógyítók mindig a legalacsonyabb aktuális HP-val rendelkező hőst gyógyítják (akár saját magát is).
- A sebzés/gyógyítás a megadott min/max értékek közötti véletlen értékkel történik.
- Minden karakter egy megadott időintervallumból véletlenszerűen választott ideig várakozik, mielőbb újabb akciót hajthat végre (ezeket az értékeket tekintjük a karakter tulajdonságának).
- A karakterek konkrét tulajdonságai:
  - Ellenség: 300HP, 30-45 sebzés, 600-1200ms várakozás.
  - Támadó: 120HP, 5-20 sebzés, 1300-1500ms várakozás.
  - Gyógyító: 100HP, 25-35 gyógyítás, 1500-2000ms várakozás. A max HP-nál nem gyógyíthat többet!
- Az intervallumok közötti random érték generáláshoz használd a `java.util.Random` osztályt!
- A program egy olyan esetet szimuláljon, ahol 4 támadó, 1 gyógyító és 1 ellenség van.
- Minden karaktert feleltess meg egy szálnak!
- A támadások/gyógyítások közötti várakozáshoz használd a `Thread.sleep` metódust!
- A szimulációnak vége, ha az ellenség megöli az összes hőst, vagy ha a hősök megölik az ellenséget.
- Szálbiztosság:
  - A gyógyító és az ellenség is hozzá kell, hogy férjen minden hőshöz, használj szálbiztos adatszerkezetet a tárolásukra! A halott hősöket távolítsd el az adatszerkezetből!
  - Mivel a támadások és gyógyítások egy időben is történhetnek, használj `AtomicInteger` -t a HP-ok nyilvántartására!
  - Mivel a random szám generátort több szál is használja, gondoskodj ennek szálbiztos módjáról!
  - Minden szálnak tudnia kell róla, hogy véget ért-e már a szimuláció, ehhez használj `AtomicBoolean` változót!