

Feladat – NagyZH – FONTOS INFÓK

- Alkalmazni kell a megoldás során a tanult objektum-orientáltsági elveket.
- A bemeneti fájlokat (**in-knights.json**, **in-rangers.json**, **in-mages.json**, **in-party-1.txt**, **in-party-2.txt**, **in-party-3.txt**) teszteléskor a build könyvtárba kell másolni.
- Feltehetjük, hogy a bemenő/teszt adatok helyesek, érvényesek (kivéve, ha a feladat mást nem állít).
- A megoldás teljes forráskódját egyetlen ZIP fájlba csomagolva kell feltölteni.

A feladat leírása

A **main**-ben és ebben a leírásban szereplő osztályok és metódusok igény szerint **átnevezhetők**, az itteni nevek csak példák.

A feladatban harcosok menedzselése a cél, akiket fel lehet bérelni, hogy megküzdjenek egy sárkánnyal. Háromféle szereplő van: lovagok, lövészek és mágusok. Mindegyiknek van egy saját szöveges azonosítója, illetve egy ára (zsoldja), amennyiért fel lehet bérelni. A lovagoknak ezen kívül van ereje (strength, egész), a lövészeknek van ügyessége (agility, egész), a mágusoknak van varázsereje (magic, lebegőpontos). Ezen tulajdonságok mindegyike pozitív.

Legyen egy megfelelő osztályszerkezet, amiben tárolni tudjuk a különböző szereplőket. A **Tavern** osztály a gyűjtőhely a szereplőknek, a **loadAll** metódusa három JSON fájl nevét kapja meg, és töltsse be onnan rendre a lovagok, a lövészek és a mágusok adatait. A **printAll** metódus írja ki mindegyiknek az adatait.

A **Party** osztály reprezentálja a felbérelt csapatot. Konstruktornak kapja meg egy szövegfájl nevét, és a **Tavern**-t, ahol a szereplők vannak. A fájlban a felbérlendő szereplők azonosítói vannak felsorolva, az azonosítók nem tartalmazznak whitespace-t. A konstruktor töltsse is be az adatokat. A **Party** osztályban legyen **printAll** metódus, ami a csapat tagjait írja ki. A csapaton belül a sorrend is lényeges. Legyen a **Party** osztályban **getTotalCost** metódus is, ami visszaadja a csapat teljes zsoldját.

A **Dragon** osztály reprezentál egy sárkányt. Van neki életereje és páncélja. A **Dragon** osztály kódját nem szabad módosítani. Legyen a **Party** osztályban egy **raid** metódus, ez szimulálja a csatát a csapat és egy sárkány között (ez utóbbit paraméterként kell átadni), és közben ki is írja, hogy éppen mi történik. A csata az alábbi algoritmus szerint zajlik.

- A csapatból mindenki egyenként, a csapat sorrendjében megtámadja a sárkányt. A támadás módja a harcostól függ (lásd később). Ha bárkinek a támadását követően a sárkány életereje 0-ra vagy az alá csökken, a csata azonnal véget ér a csapat győzelmével, a többiek nem kerülnek sorra.
- Ha minden csapattag támadott, akkor a sárkány felfal 1 csapattagot.
- Ezt követően a csapat újra támad, a sárkány újra felfal 1 csapattagot, és így tovább.
- A felfalt csapattagok többé nem támadnak.
- A sárkány mindig a sorban következő csapattagot választja.
- Ha a sárkány mindenkit felfalt, a csata a csapat vereségével ér véget.

A háromféle harcos támadása az alábbi módon működik (ez a csatától függetlenül is implementálható):

- A lovag annyi sérülést okoz a sárkánynak, amennyi a lovag ereje. Ezt a sérülést viszont csökkenti a sárkány páncélja, de soha nem 1 alá. A sárkány életerejére a sérülés mértékével csökken.
- A lövész annyival csökkenti a sárkány életerejét, amennyi a lövész ügyessége. A sárkány páncélja ilyenkor hatástalan.
- A mágus támadásának hatására a sárkány megmaradt életerejének arányosan akkora részét veszíti el, amennyi a mágus varázsereje. A sárkány páncélja ilyenkor is hatástalan. Konkrétan, ha HP_{elotte} a sárkány megmaradt életerejére a támadás előtt és M a mágus varázsereje, akkor a támadás után a sárkány életerejére az alábbi képlet szerint alakul (egészekre lefelé kerekítve).

$$HP_{utana} = HP_{elotte} \cdot (1 - M)$$

Pontozás

- | | |
|--|----------|
| 1. A harcosok osztályai, megfelelő adatszerkezet | (8 pont) |
| 2. Tavern , beolvasás JSON fájlból | (6 pont) |
| 3. Tavern , <i>printAll</i> metódus | (6 pont) |
| 4. Party , beolvasás szövegfájlból | (4 pont) |
| 5. Party , <i>printAll</i> metódus | (2 pont) |
| 6. <i>getTotalCost</i> metódus | (2 pont) |
| 7. <i>raid</i> metódus, a csata algoritmus | (7 pont) |
| 8. a háromféle támadási mód | (5 pont) |

Összesen: **40 pont.**