Feladat – NagyZH – FONTOS INFÓK

- **Alkalmazni kell a megoldás során a tanult objektum-orientáltsági elveket (ősosztály-gyerekosztályok, virtuális metódusok, egységbe zárás stb).**
- 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 megfelelő helyre 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 harcosokat. A **Tavern** osztály a gyűjtőhely a szereplőknek, a *loadAll* metódusa sorrendben a három JSON fájl nevét kapja meg, és töltse be onnan rendre a lovagok, a lövészek és a mágusok adatait. Lásd az input fájlokat.

Innentől kezdve (az előbb említett beolvasást leszámítva) a teljes pontszámhoz a **Tavern** osztály nem hivatkozhat a konkrét gyerekosztályokra, csak az ősosztályra. Vagyis, a gyerekosztályok szerinti esetszétválasztásokat ajánlott virtuális metódusokkal megoldani.

Legyen a **Tavern** osztályban *printAll* metódus, ami kiírja a szereplők adatait.

A **Party** osztály reprezentálja a felbérelt csapatot. Konstruktorban kapja meg egy szövegfájl nevét, és a **Tavern**-t, ahol a harcosok vannak tárolva. A fájlban a felbérlendő harcosok azonosítói vannak felsorolva, az azonosítók nem tartalmaznak whitespace-t és mind különbözők. A konstruktor töltse is be az adatokat. A csapaton belül a sorrend lényeges. (Tipp: célszerű a **Tavern**-ből a megfelelő harcosok mutatóit lekérdezni, és a **Party**-ban ezeket a mutatókat eltárolni.) A **Party** osztályban is legyen *printAll* metódus, ami a csapat tagjait írja ki.

Legyen a **Party** osztályban egy *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 egy virtuális metódus a harcosokat reprezentáló ősosztályban, ami paraméterben várja a sárkányt, és csökkenti a sárkány életerejét a harcostól függően:

- 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 életereje 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 életereje a támadás előtt és M a mágus varázsereje, akkor a támadás után a sárkány életereje az alábbi képlet szerint alakul (egészekre lefelé kerekítve, vagyis nem szükséges külön konvertáló vagy kerekítő műveletet végezni):

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

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, és hogy végül a csapat vagy a sárkány nyert. 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 falja fel, és a sor elejéről indul.
- Ha a sárkány mindenkit felfalt, a csata a csapat vereségével ér véget.

Pontozás

1. A harcosok osztályai, megfelelő adatszerkezet **(8 pont)**
2. **Tavern**, beolvasás JSON fájlból **(6 pont)**
3. **Tavern**, *printAll* metódus **(6 pont)**
4. **Party**, beolvasás szövegfájlból **(4 pont)**
5. **Party**, *printAll* metódus **(3 pont)**
6. *getTotalCost* metódus **(3 pont)**
7. Virtuális metódus a háromféle támadási módra **(4 pont)**
8. **Party**, *raid* metódus, a csata algoritmusa **(6 pont)**

Összesen: **40 pont**.