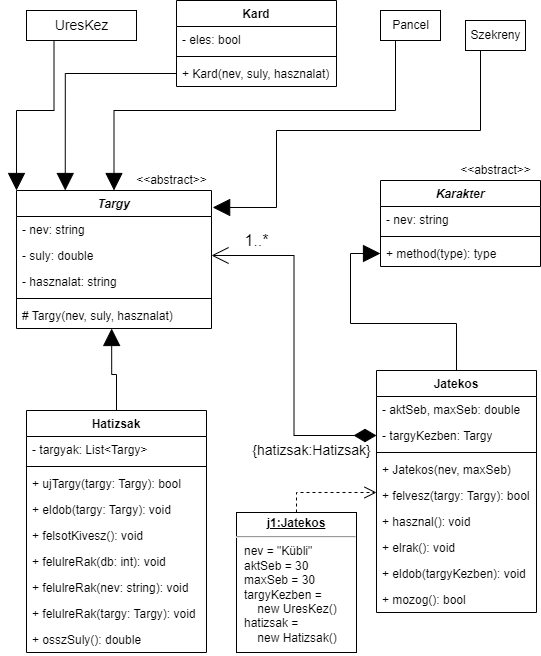
**Fantasy feladat - UML**

**Feladatleírás**:

Egy játékos karakter tárgyakat tud felvenni. Minden felvett tárgy csökkenti a mozgási sebességét. Amit felvett, az a hátizsákjába kerül. Tudja használni, ilyenkor a kezébe kerül az a tárgy, ami a hátizsák tetején van. Ez a legutoljára felvett tárgy. A hátizsákját át tudja rendezni, megadhatja, hogy mi kerüljön legfelülre. A kezében lévő tárgyat el is tudja dobni, vagy használhatja. Ha van valami a kezében, az még 30%-kal csökkenti a mozgási sebességét.



--

**UML lépésről lépésre**

*Egy játékos karakter tárgyakat tud felvenni.*

Kell egy *Jatekos* és egy *Targy* osztály.

Jatekos osztály:

+ felvesz(targy: Targy): boolean 🡪 A Jatekos osztályon belül kell egy *felvesz(targy: Targy)* metódus. Mivel a játékos tárgyakat tud felvenni (azaz vagy felvesz egy tárgyat, vagy nem), *boolean* típusú lesz a metódus visszatérési értéke.

*Minden felvett tárgy csökkenti a mozgási sebességét.*

Jatekos osztály:

- aktSeb, maxSeb: double 🡪 egy játékosnak lesz aktuális- és maximális sebessége. A *maximális sebesség* lesz a ***default*** sebesség, amely a felvett tárgyak számának arányában csökkenhet. Tehát a 2 paraméteres konstruktornak be kell állítania, hogy aktSeb = maxSeb

+ mozog(): bool 🡪 erre feltételezhetően azért van rá szükség, mert bizonyos mennyiségű tárgy felvétele után akár le is nullázódhat a játékos sebessége.

*Amit felvett, az a hátizsákjába kerül.*

Szükség lesz egy *Hatizsak* osztályra.

Hatizsak osztály:

- targyak: List<Targy> 🡪 felvett tárgy objektumokat egy Targy típusú listában fogjuk tárolni

+ ujTargy(targy: Targy) : boolean 🡪 egy hátizsákbeli tárgy lehet újonnan felvett tárgy, de már meglévő is. (??Talán az lesz a feladata, hogy végigmenjen a targyak tömbön, és ha nem találja meg a paraméterként bemenő tárgyat, akkor hamis értékkel tér vissza ?? – ha új tárgy, akkor az összsúlynak növekednie kell)

Jatekos oszály:

+ felvesz(targy: Targy): boolean 🡪

+ elrak(): void 🡪 ha felvesz igaz, akkor hatizsak.targyak.add(targy) & csökken a játékos akt. sebessége

*Tudja használni, ilyenkor a kezébe kerül az a tárgy, ami a hátizsák tetején van. Ez a legutoljára felvett tárgy.*

Jatekos osztály:

- targyKezben: Targy 🡪kezdetben UresKez tipusu, mivel nincs semmi a játékos kezében

+ hasznal(): void

targyKezben = hatizsak.targyak.get(last) 🡪 nem jó a szintaxis, de a lényeg benne van, vagyis a *targyak* tömbbe legutoljára belepakolt (tehát a hátizsák „legfelső”) elemet adjuk át a targyKezben változónak. Ez alapjáraton (by default) a legutoljára felvett tárgy.

Hatizsak osztály:

+ felsotKivesz(): void 🡪 a *targyak* tömbből eltávolítja a legutolsó elemet

(Megjegyzés: de mi van akkor, ha a legfelső elem nem a legutoljára - tehát ÚJONNAN - felvett tárgy, hanem átrendezés útján a hátizsák tetejére került - MÁR MEGLÉVŐ - tárgy? 🡪 valószínűleg ***default****-ból* a legutoljára felvett tárgy! Egyébként pedig lehet egy régebbi, az átrendezés során legfelülre kerülő tárgy.   
Ha új tárgy, tehát a legutóbb felvett elem, akkor nem kell keresés, hanem a lista végéről „le kell csípni.”)

*A hátizsákját át tudja rendezni, megadhatja, hogy mi kerüljön legfelülre.*

Hatizsak osztály:

+felulreRak metódusok

(???felulreRak(db: int) 🡪 db = a *targyak* tömbben elfoglalt index lehet???)

*A kezében lévő tárgyat el is tudja dobni, vagy használhatja. Ha van valami a kezében, az még 30%-kal csökkenti a mozgási sebességét.*

Jatekos osztály:

hasznal() vagy eldob() a targyKezben alapján

hasznal() 🡪 marad a kezében a tárgy, tehát aktSeb = aktSeb\*0.7

eldob() 🡪 targyKezben = new UresKez() , ezenkívük aktSeb =1.3 \* aktSeb

A Hatizsak osztály eldob() metódusa csak az összsúly csökkentése szempontjából fontos (szerintem).

**Kapcsolatok**

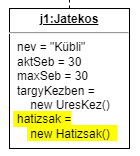
A Targy absztrakt osztályból öröklődnek a különféle tárgyak.

A hátizsák is tárgy, ezért a Hatizsak osztály is a Targyból öröklődik.

Ebben a kontextusban az üres kéz is tárgy, tehát az is Targy osztályból öröklődik.

A Karakter absztrakt osztályból öröklődik a Jatekos osztaly.

Egy játékosnak min. 1, de akármennyi tárgya lehet (kompozíció, tartalmazás). A min. 1 arra vonatkozik, hogy egy hátizsákja (amely ugye a Targy oszt. utódja, tehát tárgy) biztosan lesz:



j1 a Jatekos osztály egy példányát szimbolizálja.

A tartalmazást és az ismertséget referenciával valósítjuk meg. A Játékos-Tárgy esetében ez lehet a Jatekos osztályának egy példányában a hatizsak = new Hatizsak(), amely tehát a Targy utódosztályának a referenciája.