



# Szoftver projekt laboratórium

## PROTOTÍPUS KONCEPCIÓJA

Csapat

**61 - Zeusz**

Konzulens

Dr. Simon Balázs

### Csapattagok

---

Balla Gergely	[NEPTUN]	[email]
Joób Zalán Miklós	[NEPTUN]	[email]
Nagy Alexandra	[NEPTUN]	[email]
Tóth Boldizsár	[NEPTUN]	[email]
Zelei Mátyás	[NEPTUN]	[email]

2024. április 8.

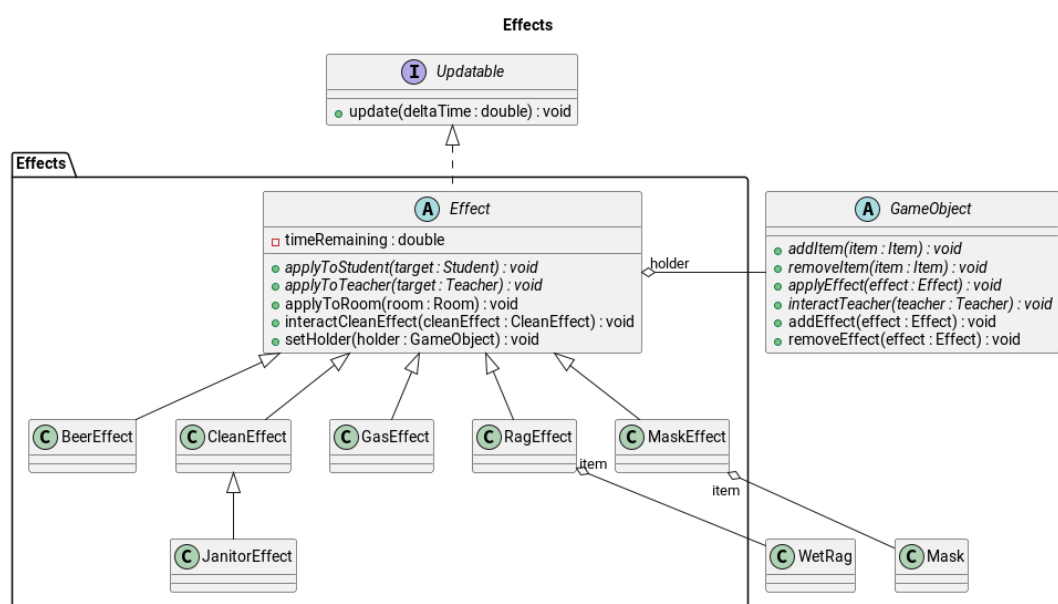


## 7. fejezet

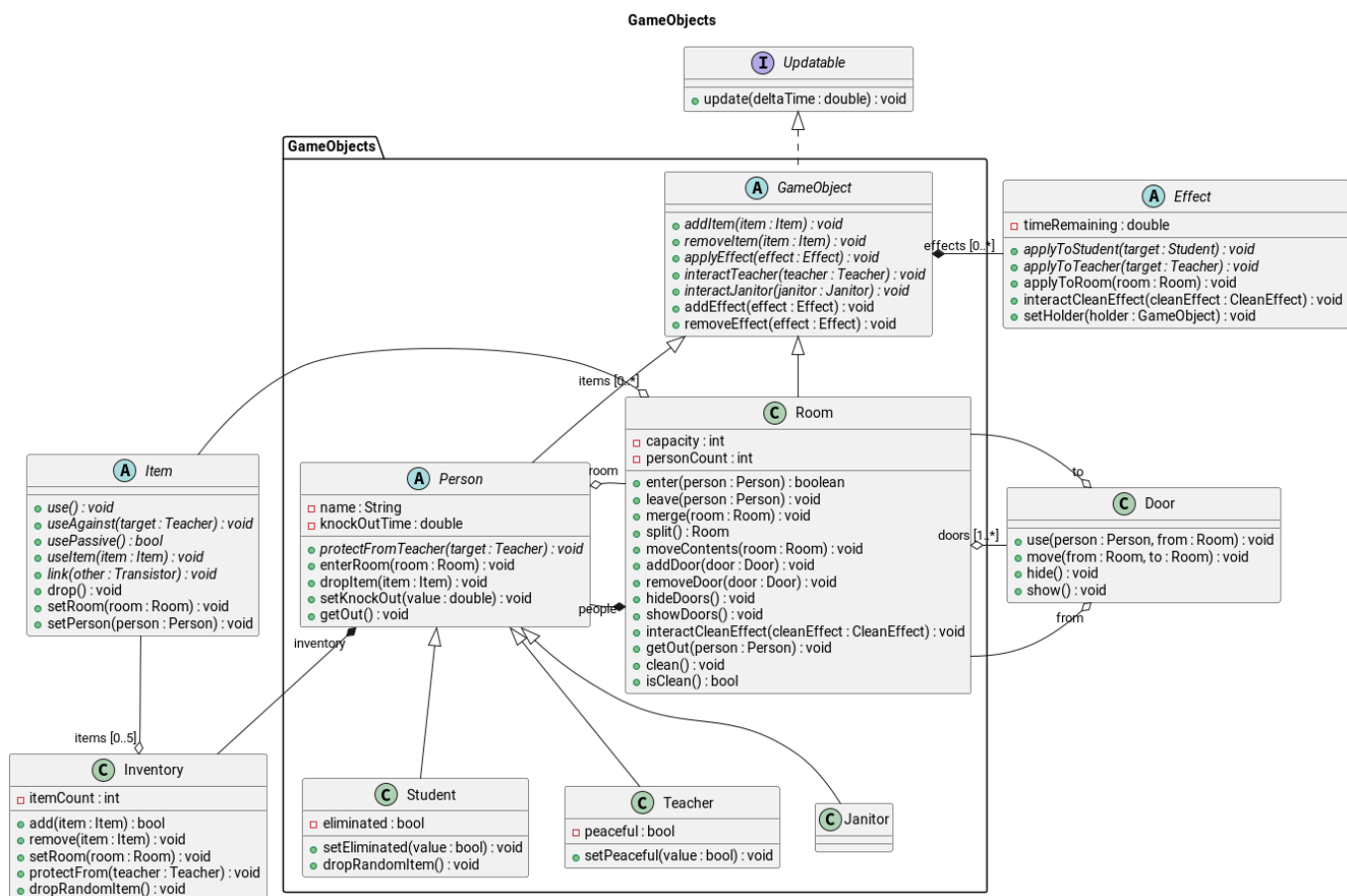
# Prototípus koncepciója

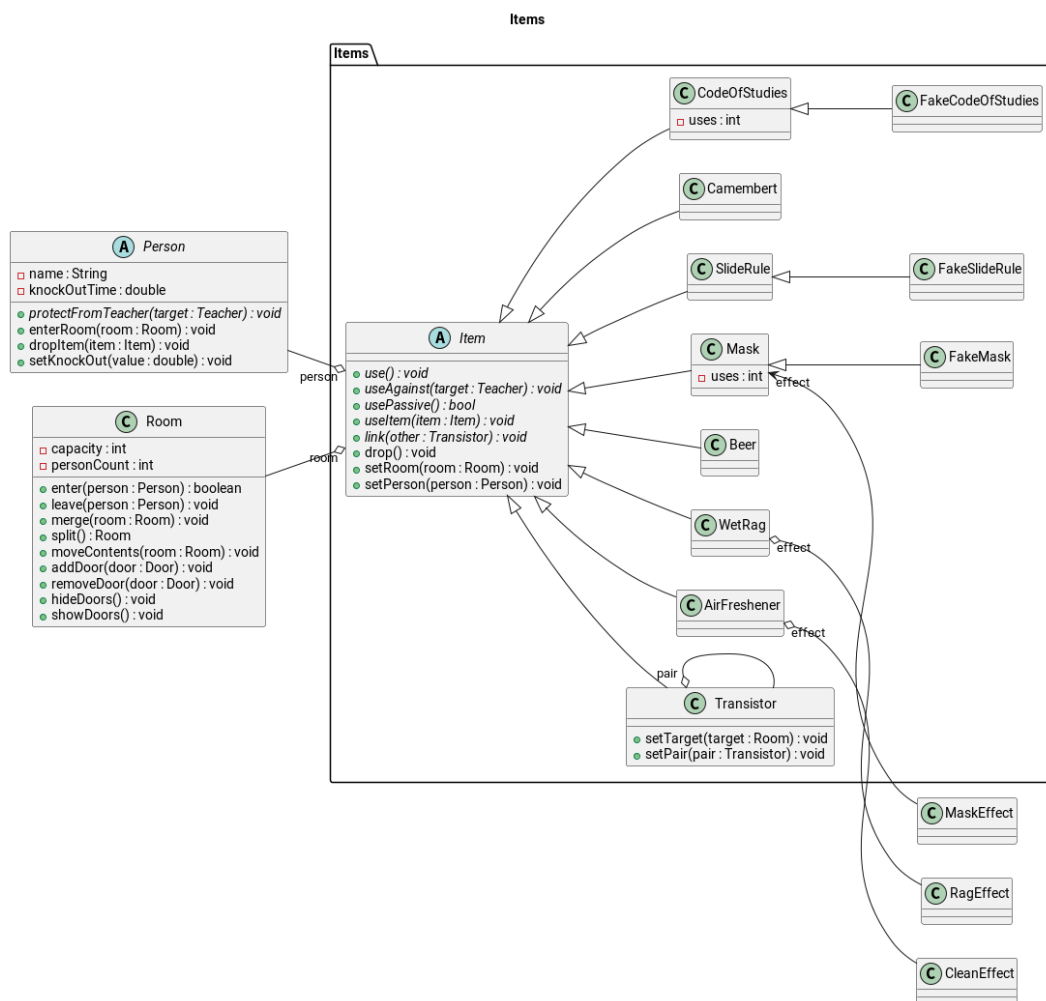
### 7.0. Változás hatása a modellre

#### 7.0.1. Módosult osztálydiagram



7.1. ábra.





7.3. ábra.

## 7.0.2. Új vagy megváltozott metódusok

### A bevezetett új osztályok

#### 7.0.2.1. AirFreshener

- Felelősség

A játékbeli tárgy reprezentálása a modellben.

- Ősosztály

Item

- Metódusok

-

#### 7.0.2.2. CleanEffect

- Felelősség

A játékbeli effektus reprezentálása a modellben.

- Ősosztály

Effect

- Metódusok

-

#### 7.0.2.3. FakeCodeOfStudies

- Felelősség

A játékbeli tárgy reprezentálása a modellben.

- Ősosztály

Item → CodeOfStudies

- Metódusok

-

#### 7.0.2.4. FakeMask

- Felelősség

A játékbeli tárgy reprezentálása a modellben.

- Ősosztály

Item → Mask

- Metódusok

-

#### 7.0.2.5. FakeSlideRule

- Felelősség

A játékbeli tárgy reprezentálása a modellben.

- Ősosztály

Item → SlideRule

- Metódusok

-

#### 7.0.2.6. Janitor

- Felelősség

A játékbeli tárgy reprezentálása a modellben.

- Ősosztály

GameObject → Person

- Metódusok

-

#### 7.0.2.7. JanitorEffect

- Felelősség

A játékbeli effektus reprezentálása a modellben.

- Ősosztály

Effect → CleanEffect

- Metódusok

-

A korábbi osztályokban történt módosítások

### 7.0.3. Effect

#### ■ Felelősség

Össosztály biztosítása a játékban található effektusoknak.

#### ■ Interfészek

Updatable

#### ■ Össosztály

-

#### ■ Attribútumok

◇ - *timeRemaining* : *double* - A hatásból hátralévő idő.

#### ■ Metódusok

- ◇ *applyToStudent(target : Student) : void* - Az effektus rátétele egy hallgatóra.
- ◇ *applyToTeacher(target : Teacher) : void* - Az effektus rátétele egy oktatóra.
- ◇ *applyToRoom(room : Room) : void* - Az effektus rátétele egy szobára.
- ◇ *interactCleanEffect(cleanEffect : CleanEffect) : void* - Az effektus interakciója egy CleanEffecttel.

### 7.0.4. Inventory

#### ■ Felelősség

Tárgyak tárolását teszi lehetővé Person objektumok számára.

#### ■ Interfészek

Updatable

#### ■ Össosztály

-

#### ■ Attribútumok

◇ - *itemCount*: *int* - Hány tárgy van benne.

#### ■ Metódusok

- ◇ *add(item : Item) : bool* - Tárgy hozzáadása az inventoryhoz.
- ◇ *remove(item : Item) : void* - Tárgy eltávolítása az inventoryból.
- ◇ *setRoom(room : Room) : void* - Az összes tárgy értesítése a szoba váltásról.
- ◇ *protectFrom(teacher : Teacher) : void* - Az inventory-ban tárolt tárgyakkal megpróbál védekezni oktató ellen.
- ◇ *dropRandomItem() : void* - Az inventory eldobja egy véletlenszerűen választott tárgyát.



### 7.0.5. Person

#### ■ Felelősség

Alaposztály szolgáltatása a mozgó játékbeli entitások számára.

#### ■ Interfészek

-

#### ■ Ősosztály

GameObject

#### ■ Attribútumok

◇ - *name* : *String* - Az ember neve.

◇ - *knockOutTime* : *double* - Az elájulás időtartama.

#### ■ Metódusok

◇ *protectFromTeacher(target : Teacher) : void* - Egy személyt valami megment egy oktatótól.

◇ *enterRoom(Room room) : void* - Az ember belép egy szobába.

◇ *dropItem(item : Item) : void* - Az ember ledobja egy tárgyát a földre.

◇ *setKnockOut() : void* - Az elájulás időtartamának beállítása.

◇ *getOut() : void* - Az embert kidobják a szobából, át kell mennie egy szomszédosba.

◇ *dropRandomItem() : void* - Az ember eldobja egy véletlenszerűen választott tárgyát.

### 7.0.6. Room

#### ■ Felelősség

Ő birtokolja a benne lévő Door, Person és (nem Personnál lévő) Item objektumokat. Ő engedélyezi vagy tagadja meg a Personok belépését. Ő értesíti a benne lévő Personokat új Person belépéséről, mérges gázzal, táblatorló rongy használatáról.

#### ■ Interfészek

-

#### ■ Ősosztály

GameObject

#### ■ Attribútumok

◇ - *capacity* : *int* - Megadja, hogy maximum hány ember fér el a szobában.

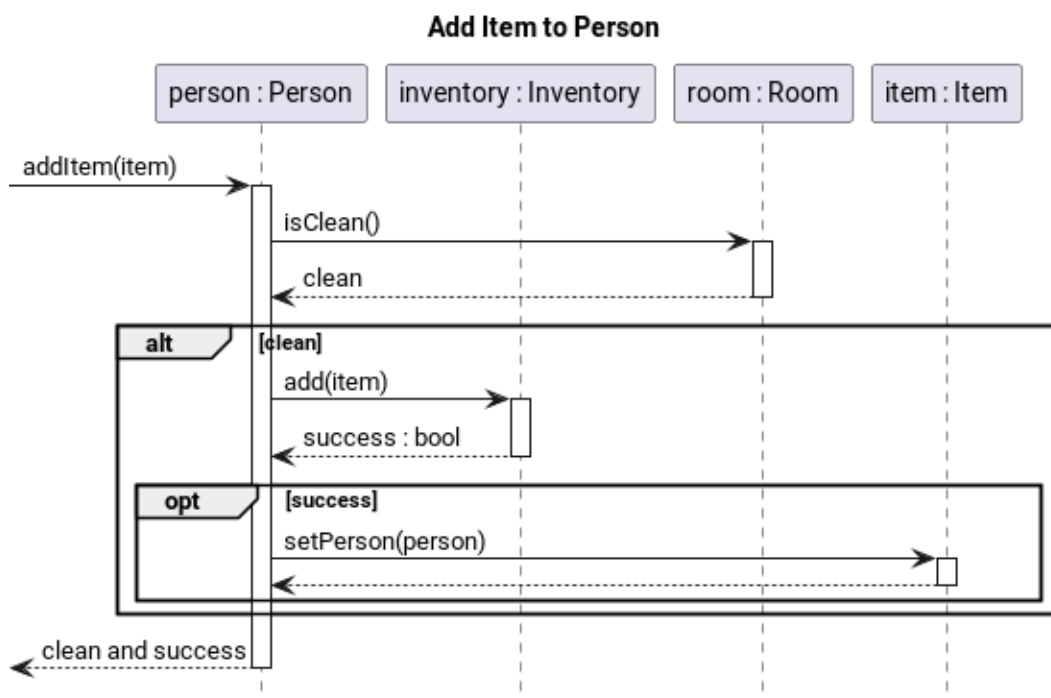
◇ - *numOfPeople* : *int* - A szobába jelenleg tartózkodó emberek száma.

◇ - *visitorsSinceClean* : *int* - A szoba kitakarítása óta hány ember lépett be a szobába.

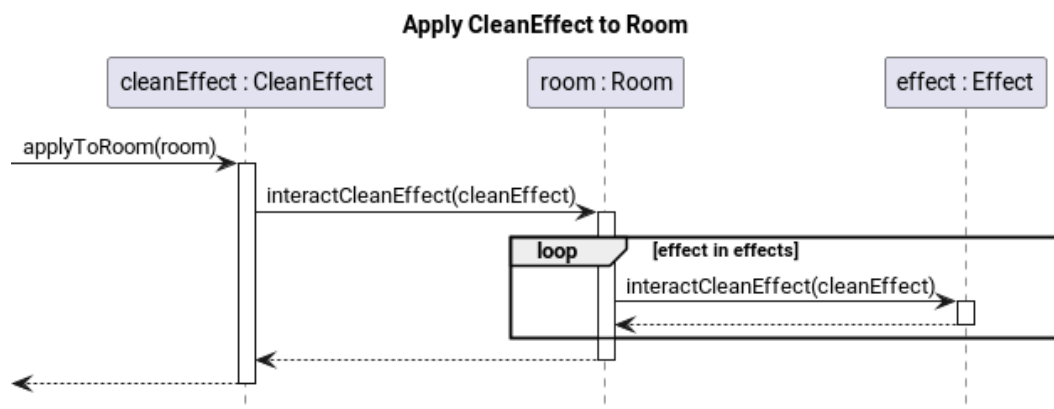
#### ■ Metódusok

- ◇ *enter(Person person) : bool* - A személy belép a szobába és a többi személy értesül róla, és reagál, ha kell.
- ◇ *leave(Person person) : void* - A szobát elhagyna egy személy, a szobában lévő emberek száma csökken.
- ◇ *merge(Room room) : void* - Két szoba összeolvad.
- ◇ *split() : void* - A szoba a szabályoknak megfelelően osztódik.
- ◇ *moveContents(Room room) : void* - A szobában található dolgokat átrakja egy másikba.
- ◇ *addDoor(Room room) : void* - Ajtó adódik a szobához.
- ◇ *removeDoor(Room room) : void* - Ajtó eltávolítása a szobából.
- ◇ *hideDoors() : void* - Az eltűnő ajtók eltűnnek.
- ◇ *showDoors() : void* - Az eltűnt ajtók megjelennek.
- ◇ *interactCleanEffect(CleanEffect cleanEffect) : void* - A szoba interakciója egy cleanEffecttel, megszünteti a gázt.
- ◇ *getOut(Person person) : void* - Áttesz egy személyt egy szomszédos szobába.
- ◇ *clean() : void* - Megszünteti a szoba ragadósságát.
- ◇ *isClean() : bool* - Visszaadja, hogy ragadós-e a szoba.

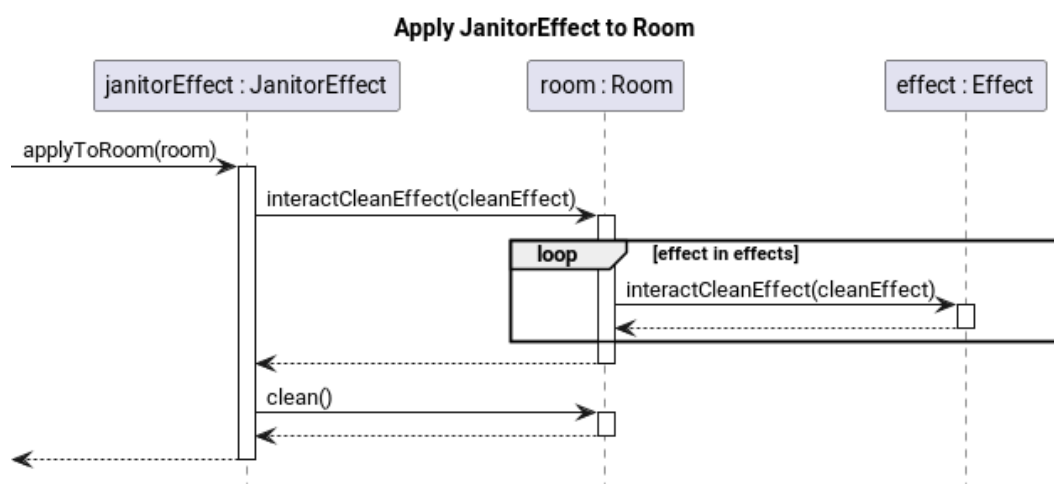
### 7.0.7. Szekvencia-diagramok



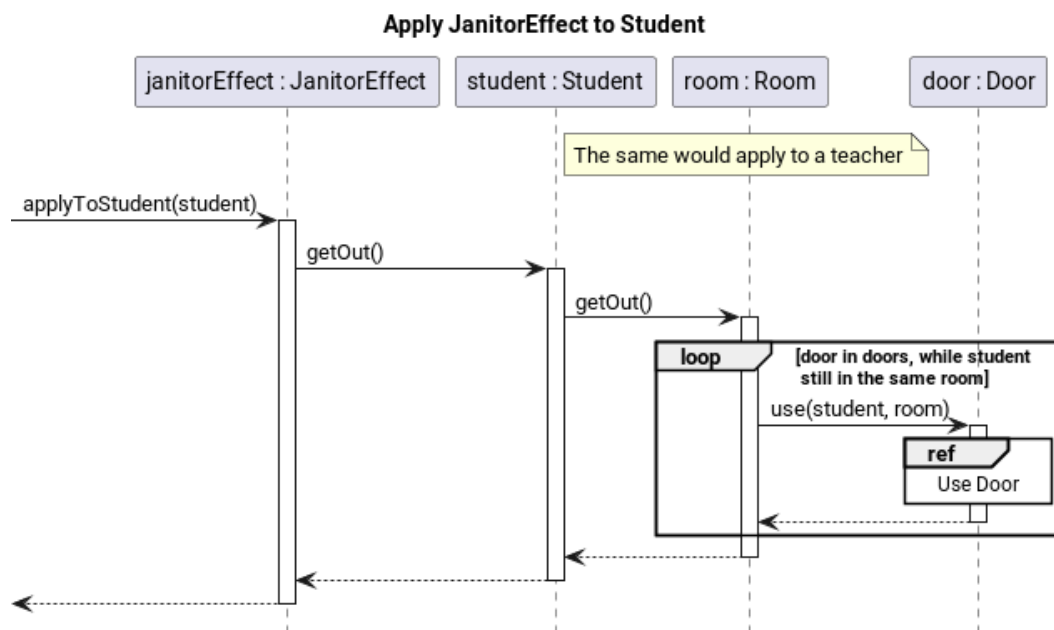
7.4. ábra. Megváltozott tárgyfelvétel



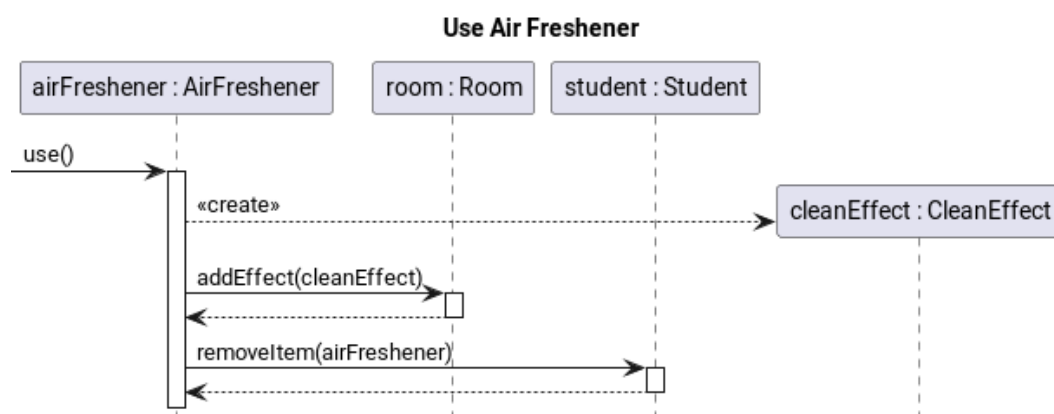
7.5. ábra.



7.6. ábra.

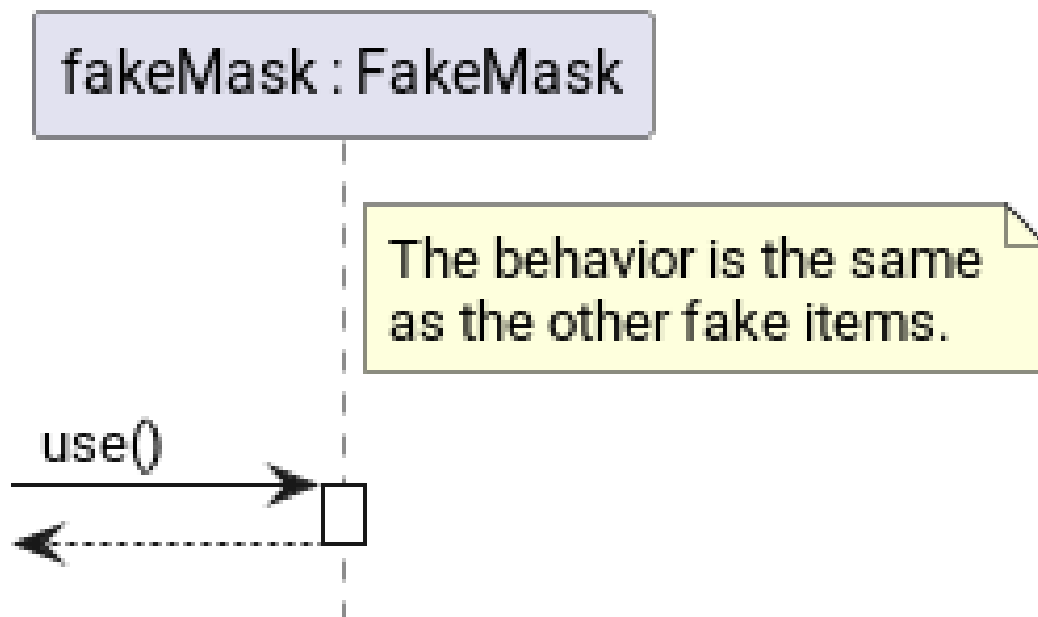


7.7. ábra.

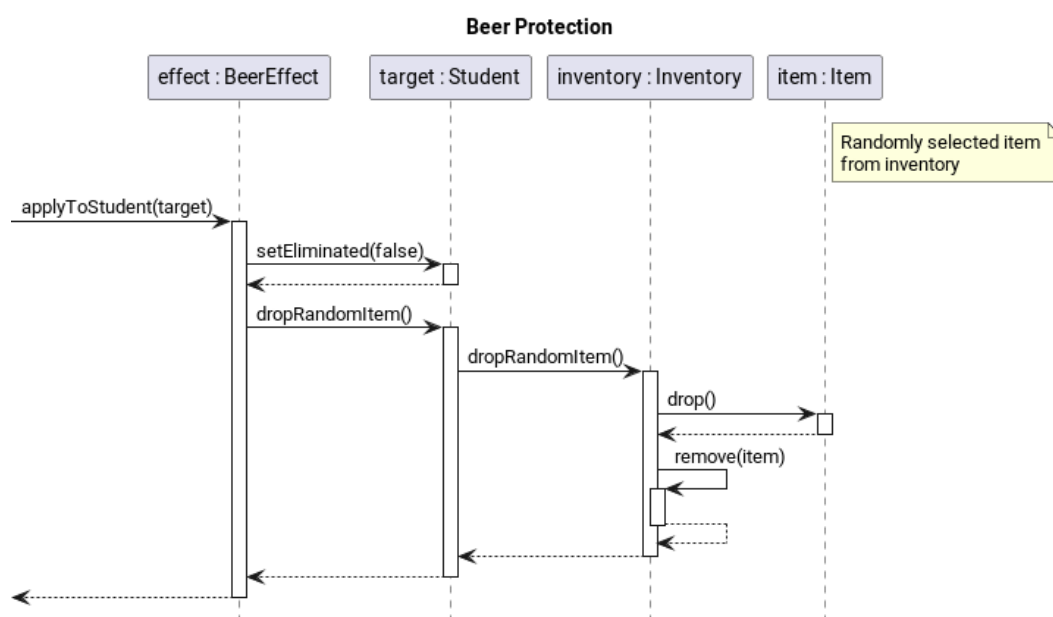


7.8. ábra.

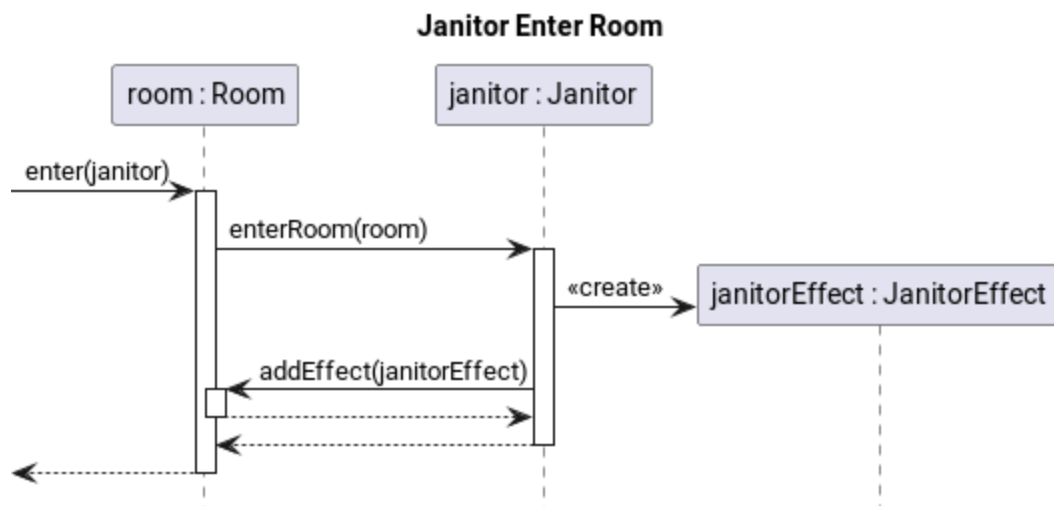
## Use Fake Item



7.9. ábra.



7.10. ábra. Megváltozott BeerEffect



7.11. ábra.

## 7.1. Prototípus interface-definíciója

### 7.1.1. Az interfész általános leírása

A prototípus karakteres interfészen kommunikál a felhasználóval. Az prototípusnak bemenő parancsokat a standard bemenetre írja a felhasználó. Az prototípus a standard kimenetre írt üzenetekkel válaszol a felhasználónak.

### 7.1.2. Bemeneti nyelv

Az interfész parancsokat vár a bemenetre. Egy parancs egy egyszavas parancsnévből és egyszavas argumentumokból áll. A parancs a parancsnévvel kezdődik, utána szóközzel elválasztva következnek az argumentumok. A parancsok:

- Create <class> <name> [<constructor params>]
  - Leírás: Létrehoz egy objektum példányt a prototípusban megadott osztályból, megadott névvel. Az első paraméter határozza meg, hogy milyen osztályú objektumot hozunk létre. A paraméternek egyeznie kell valamelyik létrehozható osztály nevével. A második paraméter az objektum neve a prototípusban. Ilyen néven kell hivatkozni rá további parancsok kiadásakor. Az osztályok, amikből lehet példányt így létrehozni: A Person-ból származó osztályok, Room, Door, az Item leszármazottjai, az Effect leszármazottjai.
  - Opciók: Ha olyan objektumot hozunk létre, amelynek van nem alapértelmezett konstruktora is, a név után megadhatjuk már a prototípusban létező objektumok nevét is. Ilyenkor a plusz argumentumok az új objektum konstruktorának paraméterei lesznek.
- Add <object1> <object2>
  - Leírás: Hozzáadja az első paraméterül megadott objektum megfelelő tárolójához a második paraméterül megadott objektumot. Az első paraméter osztálya lehet Room vagy a Person osztály valamely leszármazottja. Ha az első paraméter Room, akkor a második lehet az Effect, Item vagy Person osztályok leszármazottja. Ha az első paraméter Person leszármazott, akkor a második lehet Effect vagy Item leszármazott.
- Move <person> <door>
  - Leírás: Az első paraméterként megadott Person leszármazott objektum használja a második paraméterként megadott Door objektumot.
- Update [<object>]
  - Leírás: Meghívja a paraméterül megadott Updatable interfész implementáló objektum frissítő metódusát.
  - Opciók: Az object nem kötelező paraméter, ha nincs megadva, akkor minden objektumot frissít (ahogyan ez játék közben is történne).
- Merge <room1> <room2>

- Leírás: Beleolvastja az első paraméterül megadott Room objektumba a második paraméterül megadott Room objektumot.
- Split <room>
  - Leírás: Szétválasztja a paraméterül megadott Room objektumot.
- Use <person> <item>
  - Leírás: Az első paraméterként megadott Person leszármazott objektum használja a második paraméterként megadott Item leszármazott objektumot, ha az nála van.
- Drop <person> <item>
  - Leírás: Az első paraméterként megadott Person leszármazott objektum eldobja a második paraméterként megadott Item leszármazott objektumot, ha az nála van.
- Link <transistor1> <transistor2>
  - Leírás: Összelinkeli a két megadott Transistor objektumot.
- Oneway <door> <value>
  - Leírás: Megadott Door objektum egyirányúra állítása
  - Opciók: value egy boolean, lehet true vagy false, hogy egyirányú-e az ajtó.
- Status <object>
  - Leírás: Kiiratja a megadott objektum aktuális állapotát.
- Seed <seed>
  - Leírás: Véletlen szám generálásához használt seed beállítása. Ezzel determinisztikussá tehető minden véletlen esemény a tesztelés elősegítése érdekében.

Példák a bemeneti nyelvre:

```
Create Room r1
Create Room r2
Create Door d r1 r2
Status d
```

```
Create Student s
Create Mask m
Add s m
```



### 7.1.3. Kimeneti nyelv

A Status parancs hatására a program kiírja a kért objektum adatait. Egy sorba kiírja az osztályt és nevet. Ezután a belső állapotának változóit írja ki külön sorokba. Ha egy belső változó egy kollekció akkor ennek elemeit írja ki szóközzel elválasztva. (Ha az elemek objektumok, akkor a prototípusban szereplő nevüket írja ki.)

Példák a kimeneti nyelvre:

```
Student student1
Knock-out time: 0
Eliminated: false
Inventory: mask1 mask2 tvsz
Effects: maskeffect
```

```
Door dora
Room 1: room1
Room 2: room2
One-way: true
```

```
CodeOfStudies tvsz
Uses left: 2
Holder: student1
Room: room1
```

## 7.2. Összes részletes use-case

<b>Use-case neve:</b>	AJTÓHASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató megkísérel átlépni az ajtón. (Oktató esetében a lefolyás hasonló.)
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	<b>A.1</b> Az ajtó jó irányba átjárható. <b>A.1.1</b> Van elég hely, az átlépés sikeres. <b>A.1.2</b> Nincs elég hely, az átlépés megtagadva. <b>A.2</b> Az ajtó rossz irányba átjárható, az átlépés megtagadva.

<b>Use-case neve:</b>	TAKARÍTÓ BELÉP A SZOBÁBA
<b>Rövid leírás:</b>	Egy takarító lép a szobába.
<b>Aktorok:</b>	Takarító
<b>Forgatókönyv:</b>	Forgatókönyv <b>A.1</b> A takarító belép a szobába és kiküldi a bent lévő (nem ájult/-bénult) hallgatókat és oktatókat, valamint tisztává teszi a szobát. <b>A.1.1</b> Amennyiben a szoba mérgező volt, a takarító kiszellőztet, megszüntetve a mérgezőséget.

<b>Use-case neve:</b>	MASZKFELVÉTEL
<b>Rövid leírás:</b>	Egy hallgató felvesz egy tárgyat. (A lefolyás az oktatók és minden tárgy esetében hasonló).
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	<b>A.1</b> A szoba tiszta. <b>A.1.1</b> A hallgató tárgykészletében még van hely, így felveszi a maszkot. <b>A.1.2</b> A hallgató tárgykészletében már nincs hely, így nem veszi fel a maszkot. <b>A.2</b> A szoba nem elég tiszta, a tárgy nem vehető fel.

<b>Use-case neve:</b>	HAMIS TÁRGY HASZNÁLAT.
<b>Rövid leírás:</b>	Egy hallgató használni akar egy hamis tárgyat.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Nem történik semmi.

<b>Use-case neve:</b>	MASZKHASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató használ egy maszkot.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Mérgező szobába lépve a hallgatónál van maszk, így az kifejti a hatását.

<b>Use-case neve:</b>	LÉGFRISSÍTŐ HASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató használ egy légfrissítőt.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Ha mérgező szobába lépve a hallgatónál van légfrissítő használhatja a szoba kitisztítására.

<b>Use-case neve:</b>	TVSZ HASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató használ egy TVSZ-t.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Olyan szobába lépve, ahol van oktató, a hallgatónál van TVSZ, így az kifejti a hatását.

<b>Use-case neve:</b>	SÖRÖSPOHÁR HASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató használ egy söröspoharat.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A hallgató aktiválja a nála lévő söröspoharat, ami egy időre védeltséget biztosít számára az oktatókkal szemben, egyúttal elejt egy nála lévő véletlen tárgyat.

<b>Use-case neve:</b>	CAMEMBERT HASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató használ egy Camembert-t.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A hallgató döntésére a a Camembert mérgezővé teszi a szoba levegőjét.

<b>Use-case neve:</b>	TRANZISZTOR ÖSSZEKAPCSOLÁSA
<b>Rövid leírás:</b>	Egy hallgató összekapcsol két tranzisztort.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	<b>A.1</b> A hallgatónak van két tranzisztora. Ekkor a hallgató döntésére összekapcsolja a kettőt.

<b>Use-case neve:</b>	TRANZISZTOR HASZNÁLAT
<b>Rövid leírás:</b>	Egy hallgató használ két Tranzisztort.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	<b>A.1</b> Ha van két összekapcsolt tranzisztora megpróbál átlépni a kijelölt szobába. <b>A.1.1</b> A célszobában van hely, a hallgató átlép. <b>A.1.2</b> A célszobában nincs hely, a hallgató nem tud átlépni.

<b>Use-case neve:</b>	SÖR HATÁSA
<b>Rövid leírás:</b>	Aktív sör hatás megvédi a hallgatót
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A hallgató aktiválta a söröspoharat, ezért védett az oktatókkal szemben.

<b>Use-case neve:</b>	SIMA SZOBÁK ÖSSZEOLVADÁSA
<b>Rövid leírás:</b>	Két szoba összeolvad, egyik sem különleges.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Két szoba, room1 és room2 összeolvadnak. room2 tárgyai és a benne lévő emberek átkerülnek a room1-be, az ajtó ami eddig a room2 és room3 között volt, most már a room1 és room3 között lesz.

<b>Use-case neve:</b>	KÜLÖNLEGES SZOBÁK ÖSSZEOLVADÁSA
<b>Rövid leírás:</b>	Két szoba összeolvad, legalább az egyik különleges.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Két szoba, room1 és room2 összeolvadnak. room2 tárgyai és a benne lévő emberek átkerülnek. A közöttük lévő ajtó megszűnik. Az új szobában érvényesül a hatás, amelyik az eredeti szobákon volt.

<b>Use-case neve:</b>	ELÁTKOZOTT SZOBA AJTÓK ELTŰNÉSE
<b>Rövid leírás:</b>	Elátkozott szoba ajtajai eltűnnek.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A room elrejtí a door és door2 ajtajait.

<b>Use-case neve:</b>	ELÁTKOZOTT SZOBA AJTÓK MEGJELNÍTÉSE
<b>Rövid leírás:</b>	Elátkozott szoba ajtajai megjelennek.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A room megjeleníti a door és door2 ajtajait.

<b>Use-case neve:</b>	MASZKELDOBÁS
<b>Rövid leírás:</b>	Hallgató eldob egy maszkot (a lefutás az összes tárgy esetében azonos).
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A hallgató eltávolítja a maszkot a tárgykészletéből, ami hozzáadódik a szoba tárgyaihoz.

<b>Use-case neve:</b>	HALLGATÓ FELVESZI A LOGARLÉCET.
<b>Rövid leírás:</b>	A hallgató felveszi a logarléceket.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Amennyiben a hallgatónál van még hely és a szoba tiszta, felveszi a Logarléceket és a játék véget ér.

<b>Use-case neve:</b>	OKTATÓ FELVESZI A LOGARLÉCET.
<b>Rövid leírás:</b>	Az oktató felveszi a logarléceket.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Az oktató felveszi a Logarléceket és addig magánál tartja, amíg pl. el nem ájul.

<b>Use-case neve:</b>	SZOBA OSZTÓDÁS
<b>Rövid leírás:</b>	Egy szoba osztdik.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	Új szoba keletkezik - newRoom. A két szoba közé egy új ajtó kerül: door. Az oldDoor eltávolításra kerül a régi szobából és az új szobához adódik hozzá.

<b>Use-case neve:</b>	SZOBA FRISSÍTÉSE: HALLGATÓK
<b>Rövid leírás:</b>	Egy szoba frissül, amiben kettő hallgató található.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A szoba frissül, és frissíti a benne található hallgatókat.

<b>Use-case neve:</b>	SZOBA FRISSÍTÉSE: HALLGATÓ ÉS OKTATÓ
<b>Rövid leírás:</b>	Egy szoba frissül, amiben egy hallgató és egy oktató található.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A szoba frissül, és frissíti a benne található hallgatót és oktatót. Az oktató elveszi a hallgató lelkét.

<b>Use-case neve:</b>	SZOBA FRISSÍTÉSE: HALLGATÓ MASZKKAL ÉS GÁZ EFFEKT
<b>Rövid leírás:</b>	Egy szoba frissül, amiben egy hallgató és egy gáz effekt található. A hallgatón van egy maszk effekt.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A szoba frissül, és frissíti a benne található hallgatót és gáz hatást. A hallgató frissíti a maszk hatást. A felhasználó döntése alapján: <b>A.1</b> A gáz hatás megszűnik. <b>A.1.1</b> A maszk hatás megszűnik. <b>A.1.1</b> A maszk hatás nem szűnik meg. <b>A.2</b> A gáz hatás nem szűnik meg és elkábítja a hallgatót. <b>A.2.1</b> A maszk hatás megszűnik. <b>A.2.2</b> A maszk hatás nem szűnik meg és megszünteti a hallgató kábult állapotát.

<b>Use-case neve:</b>	SZOBA FRISSÍTÉSE: OKTATÓ ÉS RONGY EFFEKT
<b>Rövid leírás:</b>	Egy szoba frissül, amiben egy oktató és egy rongy effekt található.
<b>Aktorok:</b>	A felhasználó
<b>Forgatókönyv:</b>	A szoba frissül, és frissíti a benne található oktatót és rongy hatást. A felhasználó döntése alapján: <b>A.1</b> A rongy hatás megszűnik. <b>A.2</b> A rongy hatás nem szűnik meg és megbékíti az oktatót.

## 7.3. Tesztelési terv

### 7.3.1. A tesztek felépítése

A tesztesetek megtervezésénél az Arrange, Act, Assert mintát alkalmaztuk. A teszteseteket úgy rendeztük sorba, hogy először a legalapvetőbb funkciókat tesztelje, majd térjen rá az egyre összetettebb funkciók ellenőrzésére. Minden teszt csakis egyetlen egy funkciót tesztel.

### 7.3.2. Egyszerű működést ellenőrző tesztek

<b>Teszt-eset neve</b>	CREATEROOM
<b>Rövid leírás</b>	A teszt létrehoz egyetlen szobát. Módosítást nem hajt rajta végre. Ellenőrzi a létrehozott szoba tulajdonságait.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy a szoba osztály konstruktora megbízható.

<b>Teszt-eset neve</b>	CREATESTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Hallgatót. Más módosítást nem hajt rajta végre. Ellenőrzi, hogy a Hallgató a megfelelő tulajdonságokkal rendelkezik.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy a hallgató osztály konstruktora megbízható.

<b>Teszt-eset neve</b>	CREATEINVENTORY
<b>Rövid leírás</b>	A teszt létrehoz egy Hallgatót. Más módosítást nem hajt rajta végre. Ellenőrzi, hogy a Hallgató rendelkezik-e az elvárt belső tulajdonságú Inventory-val.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy az inventory osztály konstruktora megbízható.

Megjegyzés: Ezt a tesztet a Teacher és Janitor objektumra is elvégezhetnénk, de a tesztben releváns metódusok a Student, Teacher és Janitor esetében azonosak, mivel mind a ketten a Person osztály leszármazottai.

<b>Teszt-eset neve</b>	PLACESTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát és egy Hallgatót. A Hallgatót elhelyezi a Szobában. Ellenőrzi, hogy a szoba ténylegesen tartalmazza-e a Hallgatót.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy az elvárt belső állapotváltozások mennek végre, amikor egy Person típusú entitást egy szobába rakunk.

Megjegyzés: Ezt a tesztet a Teacher és Janitor objektumra is elvégezhetnénk, de a tesztben releváns metódusok a Student, Teacher és Janitor esetében azonosak, mivel mind a ketten a Person osztály leszármazottai.

<b>Teszt-eset neve</b>	CREATETEACHER
<b>Rövid leírás</b>	A teszt létrehoz egy Oktatót. Más módosítást nem hajt rajta végre. Ellenőrzi, hogy az Oktató a megfelelő tulajdonságokkal rendelkezik.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy az Oktató osztály konstruktora megbízható.

<b>Teszt-eset neve</b>	ADDDOOR
<b>Rövid leírás</b>	A teszt létrehoz két szobát és egy ajtót. Az ajtót a két szoba közé rakja. Ellenőrzi, hogy az ajtó objektumban a két tárolt szoba értéke megfelelő lett-e.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy az ajtó elhelyezés két szoba közt megfelelően működik.

<b>Teszt-eset neve</b>	APPLYRAGEFFECTTOROOM
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Nedves táblatörlőrongyot.. A Rongyot a diáknak átadja. Ellenőrzi, hogy a diák rendelkezik-e a megfelelő paraméterű RagEffect-el.
<b>Teszt célja</b>	A Táblatörlő által hordozott Effect Diákra való terjedésének tesztelése.

<b>Teszt-eset neve</b>	APPLYGASEFFECTTOROOM
<b>Rövid leírás</b>	A teszt létrehoz egy Camambert és egy Szobát. A Camambert elhelyezi a szobában. Ellenőrzi, hogy a szoba rendelkezik-e a megfelelő paraméterű GasEffect-el.
<b>Teszt célja</b>	A Camambert által hordozott Effect szobára való terjedésének tesztelése.

<b>Teszt-eset neve</b>	APPLYCLEANEFFECTTOROOM
<b>Rövid leírás</b>	Létrehoz egy AirFreshener-t egy Szobát és egy GasEffectet A GasEffect-et a szobára helyezi, a szobában elhelyezi a Légszűrőt. Ellenőrzi, hogy a szobában nincs többé GasEffect.
<b>Teszt célja</b>	A légszűrő tesztelése.

<b>Teszt-eset neve</b>	INHERITGASEFFECT
<b>Rövid leírás</b>	A teszt léterhoz egy szobát, egy Camambert-et, és egy diákot. A Camambert-et elhelyezi a szobában. A játékost belerakja a szobába.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy a szoba által számontartott Effectet átveszik-e a benne lévő szereplők.

Megjegyzés: Ezt a tesztet a Teacher és Janitor objektumra is elvégezhetnénk, de a tesztben releváns metódusok a Student, Teacher és Janitor esetében azonosak, mivel mind a ketten a Person osztály leszármazottai.

<b>Teszt-eset neve</b>	APPLYBEEREFFECTTOSTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Sört. A Sört a diáknak átadja. Ellenőrzi, hogy a diák rendelkezik-e a megfelelő paraméterű BeerEffect-el.
<b>Teszt célja</b>	A Sör által hordozott Effect Diákra való terjedésének tesztelése.

<b>Teszt-eset neve</b>	BEEREFFECTONSTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot, egy Táblatörlőrongyot és egy Sört. A Sört és a Rongyot a diáknak átadja. Ellenőrzi, hogy a diák elejtette-e a Rongyot.
<b>Teszt célja</b>	A BeerEffect mellékhatásának tesztelése

Megjegyzés: egynél több tárgyat nem szükséges a Diáknak adni, mivel a specifikáció csak azt kéri, hogy egy tárgyat dobjon el. Az eldobandó tárgy kiválasztása nem érdekes.

<b>Teszt-eset neve</b>	APPLYMASKEFFECTTOSTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Maszkot. A Maszkot a diáknak átadja. Ellenőrzi, hogy a diák rendelkezik-e a megfelelő paraméterű MaskEffect-el.
<b>Teszt célja</b>	A Maszk által hordozott Effect Diákra való terjedésének tesztelése.

<b>Teszt-eset neve</b>	APPLYFAKEMASKEFFECTTOSTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Hamis Maszkot. A Hamis Maszkot a diáknak átadja. Ellenőrzi, hogy a diák nem rendelkezik MaskEffect-el.
<b>Teszt célja</b>	A Hamis Maszk Effect nélküliségének tesztelése.

Megjegyzés: A fenti két teszt elvégezhető lenne az összes tárgyra és szereplőre is, de mivel az őosztályaik közösek és a FakeItem implementációknál metódusok kerülnek törlésre, ezért ezekre külön tesztek nem terveztünk.



<b>Teszt-eset neve</b>	MOVESTUDENTFROMROOMTOROOM
<b>Rövid leírás</b>	A teszt létrehoz két szobát(szobaA és szobaB), egy diákot és egy kétirányú ajtót. Az ajtót elhelyezi szobaA és szobaB közé. A diákot elhelyezi szobaA-ban. A diákot átlépteti szobaB-be. Ellenőrzi, hogy a diák átkerült-e szobaB-be.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy a szereplők képesek használni az szobák közti ajtókat.

<b>Teszt-eset neve</b>	MOVEMENTREMOVESTUDENTFROMORIGIN
<b>Rövid leírás</b>	A teszt létrehoz két szobát(szobaA és szobaB), egy diákot és egy kétirányú ajtót. Az ajtót elhelyezi szobaA és szobaB közé. A diákot elhelyezi szobaA-ban. A diákot átlépteti szobaB-be. Ellenőrzi, hogy a Diák el lett-e távolítva szobaA-ból.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy a szereplők szobák közti mozgása során az elhagyott szobából kikerül a szereplő.

<b>Teszt-eset neve</b>	DENYSTUDENTMOVEMENTFROMROOMTOROOMDUE TODIRECTION
<b>Rövid leírás</b>	A teszt létrehoz két szobát(szobaA és szobaB), egy diákot és egy Ajtót -> szobaA irányú ajtót. Az ajtót elhelyezi szobaA és szobaB közé. A diákot elhelyezi szobaA-ban. A diákot megpróbálja átléptetni szobaB-be.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy az ajtók rossz irányú mozgást nem engednek át.

<b>Teszt-eset neve</b>	DENYSTUDENTMOVEMENTFROMROOMTOROOMDUE TOCAPACITY
<b>Rövid leírás</b>	A teszt létrehoz két szobát(szobaA és szobaB), egy diákot és egy Ajtót -> szobaA irányú ajtót. Az ajtót elhelyezi szobaA és szobaB közé. A szobaB kapacitását 0-ra állítja. A diákot elhelyezi szobaA-ban. A diákot megpróbálja átléptetni szobaB-be.
<b>Teszt célja</b>	Megbizonyosodni arról, hogy a modell megtagadja a megtelt szobába való belépést.

<b>Teszt-eset neve</b>	UPDATEBEEREFFECT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Sört. A Sört a diáknak átadja. Meghívja a Diák update függvényét. Ellenőrzi, hogy frissült-e az effect.
<b>Teszt célja</b>	Az effect frissítésének ellenőrzése.

<b>Teszt-eset neve</b>	UPDATERAGEFFECT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Nedves táblatörlőt. A Táblatörlőt a diáknak átadja. Meghívja a Diák update függvényét. Ellenőrzi, hogy frissült-e az effect.
<b>Teszt célja</b>	Az effect frissítésének ellenőrzése.

<b>Teszt-eset neve</b>	UPDATEMASKEFFECT
<b>Rövid leírás</b>	A teszt létrehoz egy Diákot és egy Maszkot. A Maszkot a diáknak átadja. Meghívja a Diák update függvényét. Ellenőrzi, hogy frissült-e az effect.
<b>Teszt célja</b>	Az effect frissítésének ellenőrzése.

<b>Teszt-eset neve</b>	UPDATEGASEFFECT
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát és egy Camamber-et. A Camambert-et a szobába helyezi. Meghívja a Szoba update függvényét. Ellenőrzi, hogy frissült-e az effect.
<b>Teszt célja</b>	Az effect frissítésének ellenőrzése.

<b>Teszt-eset neve</b>	ROOMUPDATESTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát és egy Diákot. A Diákot a szobába rakja. Meghívja a szoba update függvényét. Ellenőrzi, hogy a szobában lévő diák is frissült-e.
<b>Teszt célja</b>	Az update esemény terjedésének ellenőrzése Diákra.

<b>Teszt-eset neve</b>	ROOMUPDATESTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát és egy Diákot. A Diákot a szobába rakja. Meghívja a szoba update függvényét. Ellenőrzi, hogy a szobában lévő diák is frissült-e.
<b>Teszt célja</b>	Az update esemény terjedésének ellenőrzése Diákra.

Megjegyzés: Ezt a tesztesetet a Teacher objektumra is elvégezhetnénk, de a tesztben releváns metódusok a Student és Teacher esetében azonosak, mivel mind a ketten a Person osztály leszármazottai.

<b>Teszt-eset neve</b>	ROOMUPDATEEFFECT
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát és egy BeerEffect-et. Az effectet a szobához adja. Meghívja a szoba update függvényét. Ellenőrzi, hogy a szobában lévő effect is frissült-e.
<b>Teszt célja</b>	Az update esemény terjedésének ellenőrzése Effectekre.

Megjegyzés: A BeerEffectet tetszőlegesen választottuk a teszthez. Minden effect Update metódusa ugyan úgy viselkedik, ezért szükségtelen mindegyiket tesztelni.

<b>Teszt-eset neve</b>	MERGEROOMS
<b>Rövid leírás</b>	A teszt létrehoz két szobát(szobaA és szobaB) SzobaA meghívja a merge(szobaB) metódust. A teszt ellenőrzi, hogy a szobaösszeolvastás sikeresen végbement-e.
<b>Teszt célja</b>	A szobaösszeolvadás ellenőrzése.

<b>Teszt-eset neve</b>	SPLITROOM
<b>Rövid leírás</b>	A teszt létrehoz egy szobát. A szobának meghívja a split() metódusát. Ellenőrzi, hogy a szobaosztódás megfelelően végbement-e.
<b>Teszt célja</b>	A szobaosztódás funkció tesztelése.

<b>Teszt-eset neve</b>	PICKUPITEMWITHINVENTORYSPACE
<b>Rövid leírás</b>	A teszt létrehoz egy szobát, egy hallgatót és egy Sört A hallgatót és a Sört egy szobába helyezi. A hallgatóval felveteti a Sört. Ellenőrzi, hogy Sör a Hallgató birtokába került-e
<b>Teszt célja</b>	A tárgyfelvétel funkció ellenőrzése.

Megjegyzés: Elegendő egy tetszőleges típusú Item-mel és egy tetszőleges típusú tárgyal tesztelni, mivel a tesztesethez releváns metódusok az absztrakt osztályokban kerültek definiálásra.

<b>Teszt-eset neve</b>	PICKUPITEMWITHOUTINVENTORYSPACE
<b>Rövid leírás</b>	A teszt létrehoz egy szobát, egy hallgatót és 6 darab Sört. A hallgatót és az egyik Sört egy szobába helyezi. A hallgatónak adja az 5 darab másik sört. A hallgatóval felveteti az utolsó Sört. Ellenőrzi, hogy sikertelen lett-e a tárgyfelvétel.
<b>Teszt célja</b>	A tárgyfelvétel funkció ellenőrzése.

Megjegyzés: Elegendő egy tetszőleges típusú Item-mel és egy tetszőleges típusú tárgyal tesztelni, mivel a tesztesethez releváns metódusok az absztrakt osztályokban kerültek definiálásra.

<b>Teszt-eset neve</b>	DROPITEM
<b>Rövid leírás</b>	A teszt létrehoz egy szobát, egy hallgatót és egy Sört A hallgatót a szobába helyezi és neki adja a Sört. A hallgatóval eldobatja a Sört. Ellenőrzi, hogy a Sör el lett-e távolítva a Hallgató tulajdonából és a szobába került-e.
<b>Teszt célja</b>	Az eldobás funkció ellenőrzése.

Megjegyzés: Elegendő egy tetszőleges típusú Item-mel és egy tetszőleges típusú tárgyal tesztelni, mivel a tesztesethez releváns metódusok az absztrakt osztályokban kerültek definiálásra.

### 7.3.3. Összetett működést vizsgáló tesztek.

<b>Teszt-eset neve</b>	TELEPORTUSINGTRANSISTORS
<b>Rövid leírás</b>	A teszt létrehoz két szobát(szobaA, szobaB), két tranzisztort(tranzisztorA, tranzisztorB) és egy hallgatót. A hallgatót elhelyezi szobaA-ban. A tranzisztorokat összelinkeli. TranzisztorA-t odaadja a hallgatónak, tranzisztorB-t pedig elhelyezi szobaB-ben. A hallgatót teleportáltatja TranzisztorA segítségével. Ellenőrzi, hogy a diák átkerült-e szobaB-be.
<b>Teszt célja</b>	A teleportálás funkció működésének bizonyítása.

<b>Teszt-eset neve</b>	BEERPROTECTSTUDENT
<b>Rövid leírás</b>	A teszt létrehoz egy szobát, egy Hallgatót és egy Oktatót és egy Sört. A Sört odaadja a Hallgatónak. A Hallgatót és Oktatót a Szobába rakja. Ellenőrzi, hogy az oktató nem támadja meg a Hallgatót.
<b>Teszt célja</b>	A Sör védelemnyújtásának ellenőrzése.

<b>Teszt-eset neve</b>	KNOCKOUTWITHOUTMASK
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát, egy Diákot és egy Camamber-et. A Camambert-et és a Diákot a szobába helyezi. Ellenőrzi, hogy a Diák elájult-e.
<b>Teszt célja</b>	A Cammabret működésének ellenőrzése.

Megjegyzés: Ezt a tesztet a Teacher objektumra is elvégezhetnénk, de a tesztben releváns metódusok a Student és Teacher esetében azonosak, mivel mind a ketten a Person osztály leszármazottai.

<b>Teszt-eset neve</b>	KNOCKOUTWITHMASK
<b>Rövid leírás</b>	A teszt létrehoz egy Szobát, egy Diákot, Maszkot és egy Camamber-et. A Maszkot a Diákra helyezi. A Camambert-et és a Diákot a szobába helyezi. Ellenőrzi, hogy a Diák nem elájult
<b>Teszt célja</b>	A Maszk működésének ellenőrzése.

Megjegyzés: Ezt a tesztet a Teacher objektumra is elvégezhetnénk, de a tesztben releváns metódusok a Student és Teacher esetében azonosak, mivel mind a ketten a Person osztály leszármazottai.

<b>Teszt-eset neve</b>	JANITORCLEARROOM
<b>Rövid leírás</b>	A teszt létrehoz két Szobát, egy Diákot, egy ajtót és egy Takarítót. A két szoba közé helyezi az ajtót, melyet kétirányúra állít. A diákot és a takarítót két különböző szobába helyezi. A Takarítót belépteti a diák szobájába. Ellenőrzi hogy a diák kikerült-e a szobából.
<b>Teszt célja</b>	A szoba clean() metódusának működésének tesztelése.

<b>Teszt-eset neve</b>	JANITORCLEARROOMWITHNONMOVING
<b>Rövid leírás</b>	A teszt létrehoz két Szobát, egy Diákot, egy ajtót, egy GasEffect-et és egy Takarítót. A két szoba közé helyezi az ajtót, melyet kétirányúra állít. A diákot és a takarítót két különböző szobába helyezi. A diák szobájára helyezi a gasEffectet, ezzel megbénítva őt. A Takarítót belépteti a diák szobájába. Ellenőrzi, hogy a Diák és a Takarító egy szobában marad-e.
<b>Teszt célja</b>	A szoba clean() metódusának működésének tesztelése.

<b>Teszt-eset neve</b>	TESZTESSET NEVE
<b>Rövid leírás</b>	Leírás
<b>Teszt célja</b>	A végrehajtott teszt célja

## 7.4. Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztelés megkönnyítéséhez egy teszt futtató program készül szintén Java-ban, ugyan azon feltételek mellett fordítható, és futtatható, mint a fő program.

A teszteseteket szövegfájlként olvassa be külön a futtatandó parancsokat és az ezekre a parancsokra elvárt kimenetet.

## 7.5. Napló

Kezdet	Időtartam	Résztvevők	Leírás
márc. 27. 10:00	2 óra	Balla Joób Nagy Tóth Zelei	Labor
márc. 30. 16:00	4 óra	Joób	A tesztesetek megtervezése
ápr. 02. 12:00	1 óra	Balla	Interfész leírása
ápr. 03. 11:00	1 óra	Zelei	Módosítások megtervezése, diagramok készítése
ápr. 05. 13:00	3 óra	Zelei	Módosítások megtervezése, implementálása
ápr. 07. 10:00	2 óra	Zelei	Módosított diagramok elkészítése
ápr. 07. 13:00	2 óra	Nagy	Osztálydiagram, use case készítés
ápr. 07. 23:00	1 óra	Tóth	Dokumentum átnézése