

## 8. Részletes tervek

*[A dokumentum célja, hogy pontosan specifikálja az implementálandó osztályokat, beleértve a privát attribútumokat és metódusokat, ezek definícióját is.*

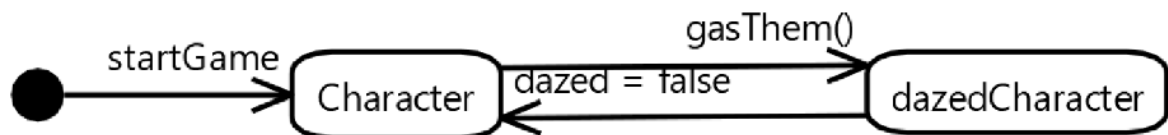
*A dokumentum második fele részletesen be kell mutassa a korábban definiált be- és kimeneti nyelv szintaxisát felhasználva, hogy mely tesztekkel lesz a prototípus ellenőrizve.]*

### 8.1 Osztályok és metódusok tervei.

#### 8.1.1 Character

- **Felelősség**

Ez az osztály a számítógép és játékosok által irányított emberek őosztálya. Minden karakter vehet fel tárgyakat és mozoghat a szobák között. Minden karakterre vonatkozik a szobák férőhelye. Elkábulhatnak a gáztól.



- **Őosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **bool dazed**: Igaz ha a karakter gáztól kábult állapotban van, hamis ha nincs.
- - **Room currentRoom**: Az a szoba amelyikben jelenleg a karakter tartózkodik
- - **Item[5] inventory**: Itt tárolódnak a karakternél fellelhető tárgyak.

- **Metódusok**

- + **void enterRoom(door D)**: belép az ajtó túloldalán található (azaz az ajtó által eltárolt két szoba közül abba amiben nem tartózkodik jelenleg a karakter)
- + **void pickUpItem(Item i)**: A karakter megpróbálja felvenni a paraméterként kapott tárgyat
- + **void dropItem(Item i)**: A paraméterként kapott tárgy az inventory-ból a jelenlegi szobába a „földre” kerül.
- + **void dropEverything()**: A karakter eldobja az összes nála lévő tárgyat, ez akkor következik be, ha elkábult és nincs maszkja.
- + **void removeItem(Item i)**: A paraméterként kapott tárgy kitörlődik a játékból, például ha elhasználódik.
- + **void useItem()**: a paraméterként kapott tárgyat megpróbálja használni, ilyen tárgyak: légfrissítő, camembert, transistor.
- + **void updateRoom(Room r)**: Beállítja a jelenlegi szobát a paraméterként kapott szobára.

- **+void teacherDuty():** Ha egy karakter tanár, akkor megpróbál mindenkit kicsapni a szobában, ha nem akkor nem csinál semmit.
- **+void setDazed(boolean dazed):** beállítja a Dazed változót a paraméterként átvett értékre.
- **+void getDazed():** Visszaadja a
- **+Room getCurrentRoom():** visszaadja a szobát amiben a karakter tartózkodik
- **+void setClothed(int i):** csak definíció, nem valósít meg lényegi működést a karakter osztályban.
- **+int dropOut():** (-1)-el tér vissza, ezzel jelezve ha a karakter nem hallgató, a szobának jelez hogy ne buktassa ki az egyetemről.

### 8.1.2 Student

- **Felelősség**

A hallgatók a játékosok által irányítható karakterek. Rendelkeznek a karakterek összes tulajdonságával, ezen kívül az összes tárgyat képesek használni is. Céljuk a logarléc megtalálása a megadott időn belül, ezzel megnyerve a játékot.

- **Ősosztályok**

Character → Student

- **Interfészek**

-

- **Attribútumok**

-

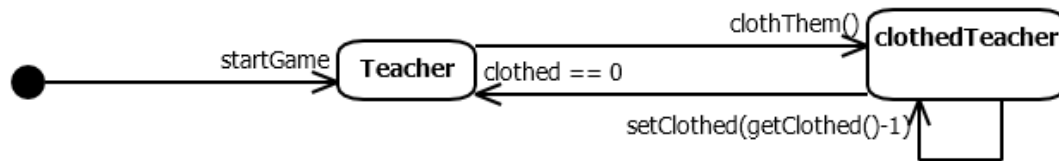
- **Metódusok**

- **+void connect(Transistor t1, Transistor t2):** A két paraméterként kapott tranzisztort megpróbálja összekapcsolni. Ez akkor lehet sikertelen, ha bármelyik tranzisztornak van már párja.
- **+int dropOut():** Ellenőrzi hogy van-e a hallgatónak védelme, és ezt visszatérési értékben jelzi a szobának. 0: nincs védelem, 1: TVSZ (Book), 2: sor (Beer), 3: rongy (Cloth).
- **+void pickUpItem(Item i):** A karakter megpróbálja felvenni a paraméterként kapott tárgyat, annyiban különbözik a tanár azonos nevű függvényétől, hogy a hallgatónak meg van engedve a logarléc felvétele.

### 8.1.3 Teacher

- **Felelősség**

A gép/controller által irányított oktatókat reprezentáló osztály. Mindenre képes amire egy karakter is. Ezen felül, ha bekerül egy szobába, megpróbálja kibuktatni az ott lévő hallgatókat. A logarlécen kívül képes felvenni a tárgyakat. Akkor nyernek az oktatók, ha lejár az idő, vagy az összes hallgató kibukott.



- **Össosztályok**

Character → Student

- **Interfészek**

-

- **Attribútumok**

- - **int clothed**: Azt tartja számon hogy még meddig van rongy hatása alatt az oktató.

- **Metódusok**

- **+int getClothed()**: Visszaadja a clothed attribútum értékét.
- **+int setClothed(int i)**: Beállítja a clothed attribútum értékét.
- **+void pickUpItem(Item i)**: Megpróbálja felvenni a paraméterként kapott tárgyat, a hallgató azonos nevű metódusától annyiban különbözik, hogy ha a tárgy logarléc, akkor nem veszi fel.
- **+void teacherDuty()**: Megpróbálja kibuktatni az összes hallgatót aki a szobában tartózkodik.

### 8.1.4 Cleaner

- **Felelősség**

A gép/controller által irányított takarítót reprezentáló osztály. „Butább” az oktatónál, csak kevés specifikus dologra képes.

- **Össosztályok**

Character → Student

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **+ void setCurrentRoom(Room r)**: A takarító legfontosabb metódusa, amellel hogy beállítja a szobát amiben jelenleg tartózkodik, kiszellőzteti azt (az a szoba nem lesz gázos) és kitessekel minden más karaktert a szobából.

## 8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

*[A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokot, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel. A tesztek leírásakor az előző dokumentumban (proto koncepciója) megadott szintakszist kell használni.]*

### 8.2.1 Teszteset1

- **Leírás**

*[szöveges leírás, kb. 1-5 mondat.]*

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- **Bemenet**

*[a proto bemeneti nyelvén megadva (lásd előző anyag)]*

- **Elvárt kimenet**

*[a proto kimeneti nyelvén megadva (lásd előző anyag)]*

### 8.2.2 Teszteset2

- **Leírás**

*[szöveges leírás, kb. 1-5 mondat.]*

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- **Bemenet**

*[a proto bemeneti nyelvén megadva (lásd előző anyag)]*

- **Elvárt kimenet**

*[a proto kimeneti nyelvén megadva (lásd előző anyag)]*

## 8.3 A tesztelést támogató programok tervei

*[A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.]*

**8.4 Napló**

Kezdet	Időtartam	Résztevők	Leírás
2010.03.21. 18:00	2,5 óra	Horváth Németh Tóth Oláh	Értekezlet. Döntés: Horváth elkészíti az osztálydiagramot, Oláh a use-case leírásokat.
2010.03.23. 23:00	5 óra	Németh	Tevékenység: Németh implementálja a tesztelő programokat.
...	...	...	...