

## 7. Prototípus koncepciója

### 7.0 Változásokkal kapcsolatos módosítások

#### 7.0.1 Szöveges elemzés

**Sheldon új pályaelemet, kereszteződő síneket vásárolt. A kereszteződés egyszintű, a különböző irányokból jövő vonatok a kereszteződésben ütköznek.**

Egy új KeresztSin osztályt vezetünk be, amihez négy SinElem csatlakozik. Így ennek az osztálynak a feladata lesz, hogy az adott irányból érkező vonatot a kereszteződésnek megfelelően továbbítsa.

**Egyes állomásokon utasok a megfelelő színű üres kocsikba (a kocsi szerelvényben elfoglalt helyzetétől függetlenül) fel tudnak szállni.**

Az Allomas osztály kapott egy új boolean ures változót. A pálya betöltésénél meg lehet adni, hogy egyes állomásokon legyenek utasok. Az állomás feladata, hogy az elhaladó, üres és megegyező színű kocsikra utasokat szállítsanak fel a kocsik új felszall metódusával.

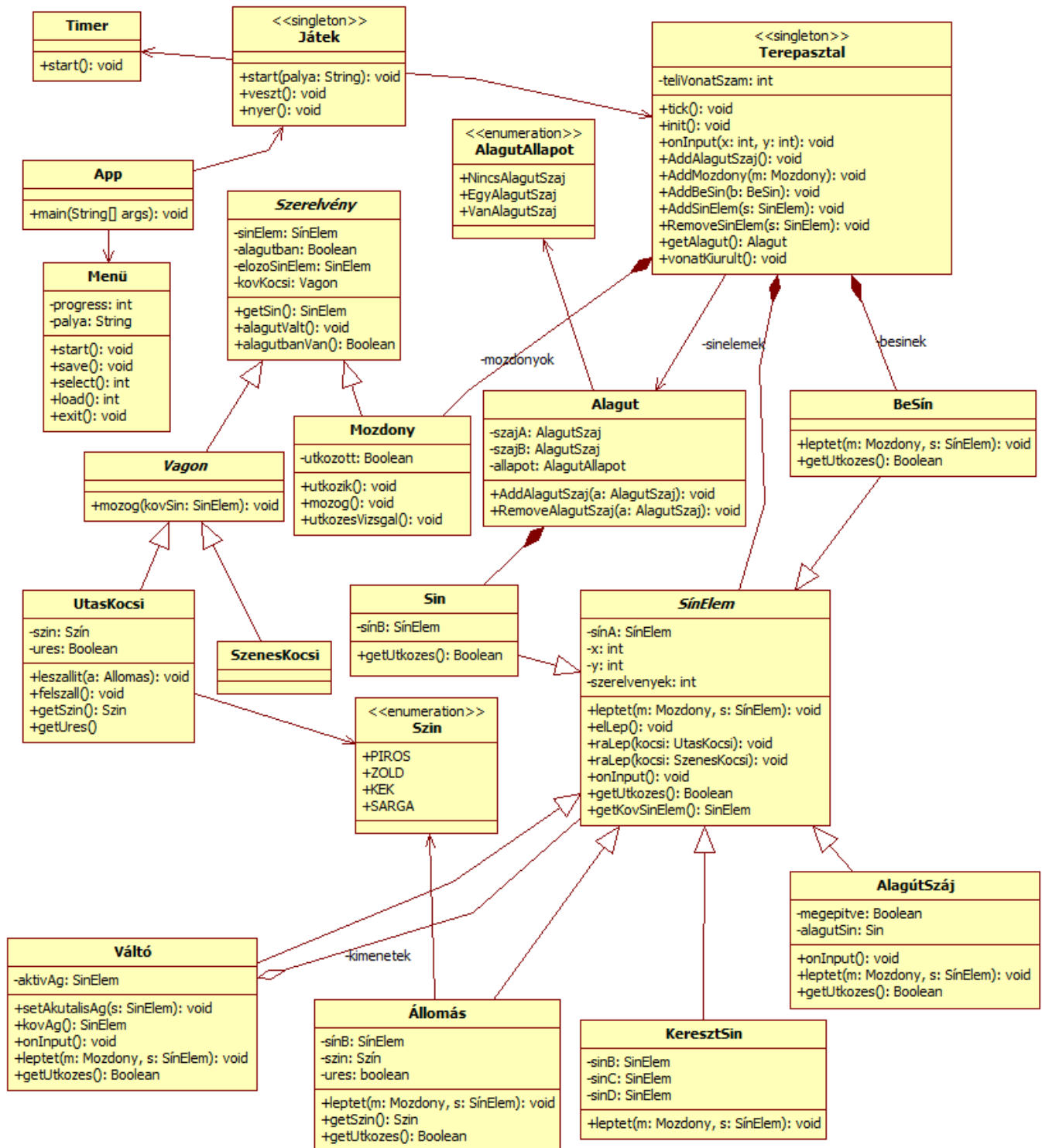
Akkor is felszállnak az állomáson tartózkodó utasok, ha az állomásra érkező vonat akkor ürül ki.

**Sheldon bővítette a vagonkészletét. Vett szeneskocsikat, amiken nem utaznak utasok, nem is tudnak felszállni. Az utasok leszállásánál az ilyen vagonokat nem vesszük figyelembe.**

Bevezettünk egy új absztrakt osztályt, Vagon, ami a Szerelveny osztályból származik. Az új SzenesKocsi osztály ebből a Vagon osztályból származik, és ez valósítja meg a leírásban kért szeneskocsit. A Kocsi osztályt az egyértelműség kedvéért átneveztük UtasKocsi-ra. Az UtasKocsi osztály is a Vagon-ból származik.

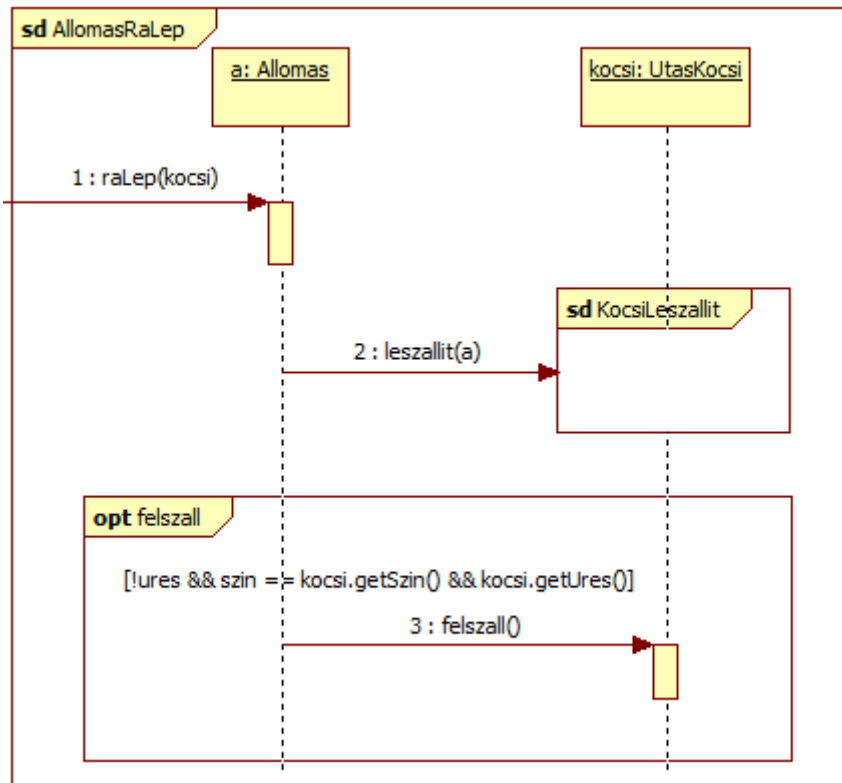
Továbbá a SinElem osztály kapott egy új metódust: raLep(k: SzenesKocsi), így ez fog meghívódni, mikor szeneskocsi lép a sínelemre. Így az állomáson külön lehet kezelni a szeneskocsi érkezését is.

## 7.0.2 Módosított osztálydiagram



## 7.0.3 Módosított szekvenciadiagrammok

### 7.0.3.1 AllomasLeptet



## 7.1 Prototípus interface-definíciója

### 7.1.1 Az interfész általános leírása

A proto interfésze a szabványos bemenetről olvas be utasításokat a Tesztelőtől, és a szabványos kimenetre írja ki a parancsok kimeneteit. Ezáltal lehetőség nyílik nem csak konzolos szövegbevitelre, hanem fájlból olvasásra is, illetve igény szerint a kimenet fájlba mentésére.

### 7.1.2 Bemeneti nyelv

**loadmap <palyanev>**

**Leírás:** Pálya betöltése

**Opciók:** palyanev: Melyik pályát töltsse be.

**play**

**Leírás:** Elindítja a játékot

**Opciók:** -

**tick <db>**

**Leírás:** Idő léptetése egységgel.

**Opciók:** db: Léptetések száma.

**valto <id>****Leírás:** Váltó aktív ágának változtatása.**Opciók:** id: Váltandó váltó azonosítója**alagutszaj <id>****Leírás:** Alagútszájra kattintás.**Opciók:** id: Adott alagútszaj azonosítója.**random <enabled>****Leírás:** Tesztelés céljából determinisztikus lefutás kikényszerítése.**Opciók:** enabled: véletlenszerűség engedélyezése.**reset****Leírás:** Állapot alaphelyzetbe állítása.**Opciók:** -**info <id> <attributum>****Leírás:** Információ lekérdezése az adott elemmel kapcsolatban.**Opciók:****argumentum nélkül:** minden id kilistázása**id:** az adott id-jű elem attribútumainak kilistázása**attributum:** az adott attribútum értéknek kiírása

Szükség volt konfigurációs fájlok tervezésére is, melyekben az egyes pályák felépítését tároljuk. Egy ilyen fájl a következő adatokat tárolja: sínelemek és azok tulajdonságai, sínelemek közötti kapcsolatok leírása, vonatok összetétele és indulása (determinisztikus lefutás esetén).

Fontos volt belátni, hogy addig nem kezdhetjük el a kapcsolatokat felépíteni, míg minden sínelemet be nem olvastunk (hiszen pl. kihez kötnénk az első beolvasott elemet?). Így az a tervezői döntés született, hogy minden sínelemet egy azonosítóval jelölünk és a fájlt úgy struktúráljuk, hogy először beolvassuk az összeset, tulajdonságaikkal együtt, viszont kapcsolatok nélkül. Majd mikor már biztosan minden elemről tudomásunk van, elkezdjük egyenként beolvasni a kapcsolatokat az azonosítók alapján.

Végül a kapcsolatok leírása után szükség van egy 3. blokkra, mely a vonatok determinisztikus pályarálépését biztosítja. Itt minden vonatra megadjuk, hogy melyik BeSinről, hányadik időlépésben, milyen típusú vagonokkal indul.

A konkrét struktúra példaként az alábbi mintafájl szolgál:

besin b1	//sínelemek felsorolása:
besin b2	//[Típus ID] formátumban
normalsin n1	//elemek sorrendje nem számít
allomas a1 p 1	//a1 paraméterei: p-piros, 1-utasok
allomas a2 k 0	//a2 paraméterei: k-kék, 0-üres
valto v1	
alagutszaj s1	
alagutszaj s2	
alagutszaj s3	
keresztisin k1	
.	
b1-a n1-a	//kapcsolatok felsorolása: sorrend nem számít. [ID-ág ID-ág] formátumban
n1-b a1-a	//pl.: n1 azonosítójú normalsin típusú sínelem sinB ága kapcsolódik a1
a1-b v1-a	azonosítójú allomas típusú sínelem sinA
v1-b s1-a	ágához (és fordítva)
v1-c k1-a	
k1-b s2-a	
k1-c s3-a	
k1-d b2-a	
.	
b1 2 ppxk	//vonatok felsorolása:[besin idő vagonok]
b2 15 xxxxp	//pl.: vonat érkezik b1 azonosítójú besin típusú sínelemre a 2. ticknel.
	Szerelvények: Mozdony, piros, piros, szenes, kék

A későbbiekben itt tervezzük a sínelemek grafikus tulajdonságainak (pl. koordináták) eltárolását is.

### 7.1.3 Kimeneti nyelv

**loadmap <palyanev>:**

**<palyanev>. palya kiválasztva**

**play:**

**Jatek elindult**

**tick <db>:**

**<db> tick megtörtént**

**valto <id>:**

**<id>. valto atvaltott az <ág\_id>. agba**

**alagutszaj <id>:**

**<id>. alagutszaj megepitve**

**<id>. alagutszaj lerombolva**

**alagut megepitve**

**alagut lerombolva**

**random <enabled>:**

**random enabled**

**random disabled**

**reset:**

**reset done**

info <id>:

<id>:

<Atributum1 neve>: <atributum1 értéke>

<Atributum2 neve>: <atributum2 értéke>

...

## 7.2 Összes részletes use-case

Use-case neve	Pálya betöltése
Rövid leírás	A program betölt egy pályát.
Aktorok	Tesztelő
Forgatókönyv	Tesztelő által megadott nevű pálya betöltődik fájlból.

Use-case neve	Játék indítása
Rövid leírás	A betöltött pálya elindítása.
Aktorok	Tesztelő
Forgatókönyv	A játék elindul, az egyes objektumok inicializálódnak.

Use-case neve	Léptetés
Rövid leírás	A játék léptetése.
Aktorok	Tesztelő
Forgatókönyv	A játék továbblép megadott időegységgel.

Use-case neve	Váltó váltás
Rövid leírás	Adott váltó váltása.
Aktorok	Tesztelő
Forgatókönyv	A Tesztelő által kiválasztott váltó aktív ága átvált a következő ágba.

Use-case neve	Alagútszáj módosít
Rövid leírás	Adott alagútszáj módosítása.
Aktorok	Tesztelő
Forgatókönyv	A Tesztelő által kiválasztott alagútszáj megépül vagy lerombolódik.

Use-case neve	Random beállítása
Rövid leírás	Tesztelő engedélyezheti vagy kiiktathatja a determinisztikussági faktort a játékból.
Aktorok	Tesztelő
Forgatókönyv	Be illetve kikapcsolódnak a véletlenszerű elemek a játékban.

<b>Use-case neve</b>	Reset
<b>Rövid leírás</b>	Tesztelő a játék során bármikor visszaállíthatja a játékot a kiindulási állapotba.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A játék visszaáll az alaphelyzetbe.

<b>Use-case neve</b>	Info
<b>Rövid leírás</b>	Információ lekérése adott azonosítójú elemről.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Kiíródik az azonosítóval adott elem összes attribútuma.

### 7.3 Tesztelési terv

<b>Teszt-eset neve</b>	Játék fut
<b>Rövid leírás</b>	A tesztelő kiválasztja a megfelelő pályát, amit aztán elindít, ezután a játékban egy vonat kering olyan körülményekkel, hogy a játéknak soha ne legyen vége.
<b>Teszt célja</b>	Tesztelni, hogy a játék megfelelően elindul és működik.

<b>Teszt-eset neve</b>	Besín működik
<b>Rövid leírás</b>	A megfelelő pálya elindítása után, a besín elemeken szerelvények érkeznek a pályára.
<b>Teszt célja</b>	Tesztelni, hogy a pályára beérkeznek a szerelvények.

<b>Teszt-eset neve</b>	Vonat ütközik
<b>Rövid leírás</b>	A megfelelő pálya elindítása után, két vonat rövid időn belül ütközik egymással.
<b>Teszt célja</b>	Tesztelni, hogy két vonat ütközésekor vége a játéknak.

<b>Teszt-eset neve</b>	Játék nyer
<b>Rövid leírás</b>	A megfelelő pálya elindítása után, egy beérkező szerelvény egyetlen kocsijából rövid időn belül leszállnak az utasok, aminek hatására a játékos nyer.
<b>Teszt célja</b>	Tesztelni, hogy az utolsó kocsiból leszállva a játékos megnyeri-e a játékot.

<b>Teszt-eset neve</b>	Állomásnál leszáll
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és Állomás. Utána jön a BeSinen egy vonat, ami tartalmaz legalább egy Kocsit, a vonat halad a Sinen, amíg el nem ér egy olyan színű Állomást, ahol le tudnak szállni az utasok. Az utasok leszállnak, és lesz egy üres kocsija a vonatnak.
<b>Teszt célja</b>	Tesztelni, hogy a leszállás rendesen működik-e az Allomas és a Kocsi osztályban, azaz hogy helyesen értékeli-e ki az előtte lévő üres kocsik létét, majd helyesen ürül-e ki a Kocsi.

<b>Teszt-eset neve</b>	Állomásnál nem száll le
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és Állomás. Utána jön a BeSinen egy legalább kettő, különböző színű Kocsit tartalmazó vonat, a vonat halad a Sinen, amíg el nem ér egy Állomást, aminek a Szin-e nem egyezik meg az első nemüres Kocsiéval, de megegyezik bármelyik más Kocsi színével. Az utasok nem szállnak le, és megy tovább a vonat.
<b>Teszt célja</b>	Tesztelni, hogy a leszállás rendesen működik-e az Allomas és a Kocsi osztályban, azaz hogy helyesen értékeli-e ki az előtte lévő üres kocsik létét, majd helyesen kiürülés nélkül továbbmegy.

<b>Teszt-eset neve</b>	Állomásnál felszáll
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és Állomás. Utána jön a BeSinen egy vonat, ami tartalmaz legalább egy Kocsit. A vonat halad a Sinen, amíg el nem ér egy Állomás-t, ahol leszállnak az utasok az első kocsiból. Utána halad a Sinen, és elér egy olyan Állomást, ahol fel akarnak szállni. Az utasok felszállnak, és megy tovább a vonat.
<b>Teszt célja</b>	Tesztelni, hogy a felszállás rendesen működik-e, azaz hogy felszállnak az utasok az arra alkalmas Állomáson.

<b>Teszt-eset neve</b>	Állomásnál nem száll fel
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és Állomás. Utána jön a BeSinen egy vonat, ami tartalmaz legalább egy Kocsit. A vonat halad a Sinen, amíg el nem ér egy Állomás-t, ahol leszállnak az utasok az első kocsiból. Utána halad a Sinen, és elér egy olyan Állomást, ahol nem akarnak felszállni. Az utasok nem szállnak fel, és megy tovább a vonat.
<b>Teszt célja</b>	Tesztelni, hogy a felszállás rendesen működik-e, azaz hogy nem szállnak fel az utasok az arra alkalmatlan Állomáson.

<b>Teszt-eset neve</b>	Váltó vált
<b>Rövid leírás</b>	A megfelelő pálya elindítása után, a tesztelő a megfelelő ágba váltja a váltót, amit ezután a szerelvény rövid időn belül elér és megfelelően továbbhalad rajta.
<b>Teszt célja</b>	Tesztelni, hogy a váltó megfelelően vált-e, a szerelvények helyesen átmennek-e rajta



<b>Teszt-eset neve</b>	Váltó nem tud váltani
<b>Rövid leírás</b>	A megfelelő pálya elindítása után egy szerelvény rövid időn belül elér egy váltót, ezután a tesztelő megpróbálja átkapcsolni az adott váltót, ami nem sikerül neki.
<b>Teszt célja</b>	Tesztelni, hogy ha vonat halad át a váltón, akkor a váltó nem kapcsolható.

<b>Teszt-eset neve</b>	Alagútszájnál felrobban
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és AlagutSzaj. Utána jön a BeSinen egy vonat. A vonat halad a Sinen, eléri az AlagutSzajat, és felrobban. A játékos veszít.
<b>Teszt célja</b>	Tesztelni, hogy a vonat meg nem épített AlagutSzaj-ra lépve felrobban és a játékos veszít.

<b>Teszt-eset neve</b>	Alagútszáj módosít
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és legalább kettő AlagutSzaj. Input érkezik az egyik AlagutSzajra, majd a másikra, mire megépül az alagút. Utána input érkezik az első AlagutSzaj-ra, mire ha az lerombolódik, mert az alagút üres.
<b>Teszt célja</b>	Tesztelni, hogy az AlagutSzaj építése illetve lebontása működik-e, az Alagut megépül és megszűnik-e rendesen.

<b>Teszt-eset neve</b>	Alagút épít
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és legalább kettő AlagutSzaj. Input érkezik az egyik AlagutSzajra, majd a másikra, mire megépül az alagút.
<b>Teszt célja</b>	Tesztelni, hogy az Alagútszájra kattintva helyesen működik az építés, azaz hogyha megépül mind a kettő, akkor létrejön egy alagút.

<b>Teszt-eset neve</b>	Alagúton átmegy
<b>Rövid leírás</b>	A játék elindul, betöltődik egy teszt pálya, aminek a terepasztalán található BeSin, Sin és legalább kettő AlagutSzaj. A két AlagutSzaj-ra input érkezik, ennek hatására megépül az alagút. Utána elindul egy vonat a BeSin-en, halad a Sin-en, elér egy AlagutSzaj-ra, belemegy az alagútba, majd a túloldalon kijön belőle.
<b>Teszt célja</b>	Tesztelni, hogy az alagút rendesen működik, azaz lehet benne haladni.

## 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A teszteléshez készíteni fogunk egy segédprogramot, amely tárolja az egyes teszt esetekhez tartozó beviteli parancsokat, és az elvárt kimeneteket. Majd a parancsokat lefuttatva ellenőrzi a program kimenetelét, és ez alapján képes kiértékelni, hogy a teszteset sikeres-e vagy sem.

## 7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2017.03.24. 20:00	3.5 óra	Dócs Krátky Varga	Ötletelés és dokumentum szerkesztése
2017.03.24. 21:30	2 óra	Szili	
2017.03.24. 23:00	0.5 óra	Sillye	Dokumentum ellenőrzése
2017.03.25. 15:30	1 óra	Krátky	Ötletelés, tesztelési terv
2017.03.25. 15:30	3 óra	Sillye	Ötletelés, tesztelési terv vázlata, megvalósítása
2017.03.25. 20:30	1,5 óra	Varga	Módosítások leírása, UML diagrammok módosítása
2017.03.26. 14:30	2 óra	Dócs	Módosítások, átolvasás
2017.03.26. 23:00	1 óra	Szili	Módosítások, átolvasás, formázás
2017.03.26. 23:00	0,5 óra	Varga	Dokumentum átnézése