3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Terepasztal
A terepasztal felelőssége, hogy tárolja, számontartja a szimulációhoz szükséges játékelemeket, például sín, állomás, vonat. Figyeli a vonatok összeütközéseit (vesztés), és hogy van e a pályán még teli kocsi (győzelem ha nincs).

3.1.2 Mozdony
Síneken közlekedik, fő felelőssége, hogy maga után húzhat egy kocsit. Színe nincsen és nem szállít utasokat. Soha nem áll meg.

3.1.3 Kocsi
Síneken közlekedik, maga után húzhat egy kocsit. Színe van és utasokat szállíthat. A benne utazó utasok, ugyan olyan színűek.

3.1.4 Utas
Az utas abban a színű kocsiban utazik, amelyik színű állomásra el szeretne jutni. Akkor száll le, ha a kocsija előtt lévő kocsi már üres, és megérkezik a kiválasztott állomásra.

3.1.5 Sín
Számontartja a kapcsolódó síneket. Felelőssége, hogy a ráérkező mozdonyt továbbirányítsa a következő sín egységre.

3.1.6 Váltó
A váltó egy olyan sín, amely n>=3 darab szomszédot tart számon. Felelőssége, hogy a ráérkező mozdonyt továbbirányítsa a megfelelő sín ágra, továbbá tárolja, hogy éppen melyik kijárat aktív.

3.1.7 Alagút Száj
A terepasztalon csak speciális helyen helyezhető el. Csak egy másik alagút bejárattal együtt létezhet, ezek párban vannak, képesek egymást számon tartani.

3.1.8 BeSín
Felelőssége, hogy időnként új mozdonyt és kocsikat generál a pálya szélén. A gyakoriság véletlenszerű, de függ a terepasztalon mozgó eddigi vonatok számától. Emellett figyel arra, hogy ha egy vonat ki megy a terepasztalról, akkor azt kis idő elteltével “visszadorsítja”, újra beengedi a játékba.

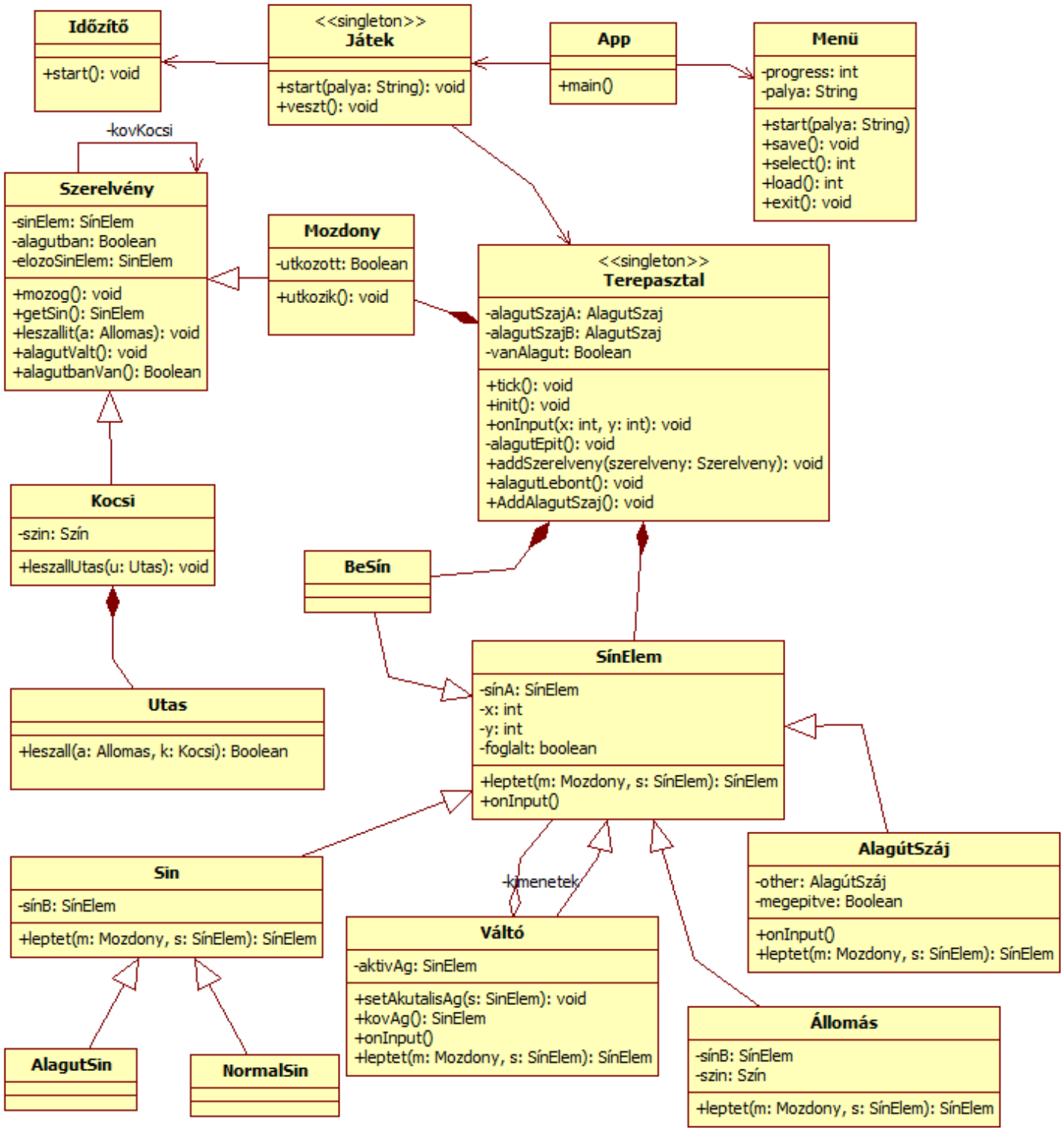
3.1.9 Állomás
A sínek mellett helyezkedik el. Felelőssége, hogy tárolja a színét, melyet a ráérkező vonat lekérdezhet. Az utasok pedig eszerint a szín szerint szállnak le a vonatról.

3.1.10 Timer
A szimuláció időbeli futására szolgál, hatására a mozdonyok továbbhaladnak. Meghívja az újrarajzolást.

3.1.11 Menü
A program funkcióinak elérésére szolgál, innen lehet a játékot elindítani, eredményjelző megtekintése, játék betöltése / mentése és kilépni. Ezen funkciók elérésére használt menüpontokat tárolja.

3.1.12 Játék
Felelőssége a játék felépítése, objektumok létrehozása, vezérlése, terepasztal betöltése. Kezeli az időzítőt.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 AlagutSin

- **Felelősség**
  Ugyanaz, mint a Sin felelőssége, de nem jelenik meg a térképen, mert földalatti.
- **Ősosztályok**
  SinElem->Sin->AlagutSin
- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**

Nincs.

3.3.2 AlagutSzaj

- **Felelősség**
  Számon tartja az alagut másik AlagutSzaj-át, és hogy meg van-e építve.
- **Ősosztályok**
  SinElem->AlagutSzaj
- **Interfészek**
  Nincs.
- **Attribútumok**
  - **other: AlagutSzaj –** A másik AlagutSzaj referenciája
  - **megepitve: boolean -** Meg van-e építve az alagút ezen szája.
- **Metódusok**
  - **void onInput() -** A bevitelre reagál az osztály.

3.3.3 Allomas

- **Felelősség**

Tárol egy Színt, melyet a ráérkező vonat lekérdezhet.

- **Ősosztályok**

SinElem -> Allomas

- **Interfészek**

Nincs.

- **Attribútumok**
  - **sinB: SinElem –** a következő SinElem
  - **szin: Szín -** Az Utas eszerint dönti el, hogy le száll-e
- **Metódusok**

Nincs.

3.3.4 App

- **Felelősség**

Felelőssége a view, controll és modell inicializálása.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**
  - **void main() -** létrehozza az Idozito, Menu, Jatek objektumokat.

3.3.5 BeSin

- **Felelősség**

Belépési pontot biztosít az új vonatoknak a Terepasztalra. Nem engedi a vonatot kimenni a terepasztalról, felrobban.

- **Ősosztályok**

SinElem->BeSin

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**
  - **SínElem leptet(Mozdony m, SínElem s) -** Ha s=SinA akkor a vonat felrobban metódusát hívja meg, különben bevezeti a vonatot SinA-ra

3.3.6 Idozito

- **Felelősség**

Felelőssége a periodikus jelgenerálás. A játék időbeli szimulálásának alapja.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**
  - **void start() -** elindítja a jelgenerálási folyamatot

3.3.7 Jatek

- **Felelősség**

Objektumok létrehozása: Terepasztal és az abban helyet foglaló Sínelemek betöltése fájlból. Időzítő tick továbbítása. Megjeleníti az Eredményjelzőt.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**
  - **void start(palya: String) -** Létrehozza a Terepasztalt, és adott fájlból tölti be rá az elemeket.
  - **void veszt() -** A játékos elveszti a játékot.

3.3.8 Kocsi

- **Felelősség**

Ugyanaz a felelőssége, mint szülőjének, a Szerelvénynek. Ezenkívül tárolja a színét, és szól az utasoknak, hogy szálljanak le, ha akarnak.

- **Ősosztályok**

Szerelveny -> Kocsi

- **Interfészek**

Nincs.

- **Attribútumok**
  - **szin: Szín –** A kocsi színe. Ez alapján dönti el az Utas, hogy leszéll-e az Állomáson.
  - **utasok: List<Utas> -** A kocsin utazó utasok listája

- **Metódusok**
  - **void leszallUtas(u Utas)**

3.3.9 Menu

- **Felelősség**

A program egyes menupontjait tárolja.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs**.**

- **Attribútumok**
  - **palya: String -** A pálya neve.
  - **progress: int** - Meddig jutott el a játékos a pályákon.

- **Metódusok**
  - **String palya() -** Visszaadja a kiválasztott pályához tartozó file címét.
  - **void save() -** Elmenti az állást, hogy mennyi pályát nyertünk meg

- **int select() -** Visszatér a pálya számával, amit kiválasztunk
- **int load() -** Betölt egy állást, és visszatér azzal, hogy meddig jutottunk el.
- **void exit() -** Kilép a játékból.

3.3.10 Mozdony

- **Felelősség**

Kérdezgeti az alatta álló SínElemet, hogy melyik lesz a következő SínElem (leptet). Szól az első Kocsinak, hogy mozogjon (mozog).

- **Ősosztályok**

Szerelveny -> Mozdony

- **Interfészek**

Nincs.

- **Attribútumok**
  - **kocsi: Kocsi –** Az első kocsi referenciája
  - **utkozott: Boolean** - Alapesetben false, ha ütközik a vonat akkor true-ra állítódik
- **Metódusok**
  - **void utkozik() - Átállítja az utkozott változó értékét**

3.3.11 NormalSin

- **Felelősség**

Ugyanaz, mint a Sin felelőssége, megjelenik a térképen.

- **Ősosztályok**

SinElem->Sin->NormalSin

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**

Nincs.

3.3.12 Sin

- **Felelősség**

Továbbirányítja a mozdonyt a következő SínElemre.

- **Ősosztályok**

SinElem->Sin

- **Interfészek**

Nincs.

- **Attribútumok**
  - **sinB: SinElem –** a másik kapcsolódó SínElem
- **Metódusok**

Nincs.

3.3.13 SinElem

- **Felelősség**

Mozgásteret biztosít a vonatok számára: adott SinElemről jött Mozdonynak megmondja, hogy melyik SinElem következik.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**
  - **sinA: SinElem –** Az egyik kapcsolódó SínElem
  - **foglalt: boolean -** Igaz, amikor szerelvény halad át rajta.
  - **x: int -** az x pozíciója a SinElemnek
  - **y: int -** az y pozíciója a SinElemnek
- **Metódusok**
  - **SinElem leptet(Mozdony m, SinElem s) -** Megadja a mozdonynak a következő SinElemet az előző SinElem függvényében
  - **void onInput() -** A bevitelre reagál az osztály.

3.3.14 Szerelveny

- **Felelősség**

Számon tartja, az előző és az aktuális SínElemet, amin tartózkodik. Tárolja, hogy alagútban van-e. Jelre tovább mozog, és Állomásra érve szól a mögötte lévő szerelvénynek, hogy

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**
  - **sin**: **SinElem** – Melyik sínen található a szerelvény
  - **elozoSin: Sinelem** – Melyik sínen volt utoljára
  - **alagutban**: **boolean –** Alagútban van-e az adott szerelvény vagy sem
  - **kovKocsi: Szerelveny -** A következő szerelvény referenciája
- **Metódusok**
  - **void mozog()–** A következő SínElemre lépteti a Szerelveny-t.
  - **SinElem getSin()–** visszatér a jelenlegi SinElemmel.
  - **void leszallit(a: Allomas) –** Megkérdezi az utasokat, hogy leszállnak-e.
  - **void AlagutValt() –** Átkapcsolja a szerelvényt, hogy alagútban van-e vagy sem.
  - **boolean alagutbanVan() -** visszatér az alagutban attribútum értékével

3.3.15 Terepasztal

- **Felelősség**

A terepasztal felelőssége, hogy tárolja a SínElem-eket, BeSín-eket és Mozdony-okat. Figyeli a vonatok összeütközéseit (vesztés), és hogy van e a pályán még teli kocsi (győzelem ha nincs). Ő építi és bontja le az alagutakat, adja hozzá a szerelvényeket a játékhoz.

- **Ősosztályok**
  Nincs.
- **Interfészek**
  Nincs.
- **Attribútumok**
  - **mozdonyok: List<Mozdony> -** A mozdonyok listája
  - **sinelemek: List<SinElem> -** A sínelemek listája
  - **besinek: List<BeSin> -** A bemeneti sínpárok listája
  - **alagutSzajA: AlagutSzaj -** Az egyik aktív alagútszáj
  - **alagutSzajB: AlagutSzaj -** A másik aktív alagútszáj
  - **vanAlagut: Boolean -** Van-e éppen megépült alagút
- **Metódusok**
  - **void tick() –** Szól a Mozdonyoknak, hogy mozogjanak.
  - **void init() –** Feltölti elemekkel a terepasztalt
  - **void addSzerelveny(szerelveny: Szerelveny) -** Kiválaszt egy véletlen BeSín-t és lerakja rá a Szerelvényt
  - **void alagutLebont() -** Lebont egy alagutat.
  - **void AddAlagutSzaj() -** Hozzáad egy alagútszájat.
  - **void onInput(int x, int y) -** Megnézi mindegyik SinElemre, hogy rakattintottak-e

3.3.16 Utas

- **Felelősség**
  Ellenőrzi, hogy adott megállónál le akar-e szállni. (Komparálja a Kocsijának és az Állomás színét, de már nem kell vizsgálnia az előtte lévő kocsi ürességét)
- **Ősosztályok**
  Nincs.
- **Interfészek**
  Nincs.
- **Attribútumok**
  Nincs.
- **Metódusok**
  - **boolean leszall(Allomas a, Kocsi k) -** a és k színét komparálja, ez alapján igaz/hamis értéked ad vissza a kocsinak, hogy le akar-e szállni.

3.3.17 Valto

- **Felelősség**

SinElem listájából az éppen aktívnak választott ág felé irányítja a mozdonyt. Ha ág felől jön, akkor SinA felé irányítja, és automatikusan vált aktív ágat.

- **Ősosztályok**

SinElem->Valto

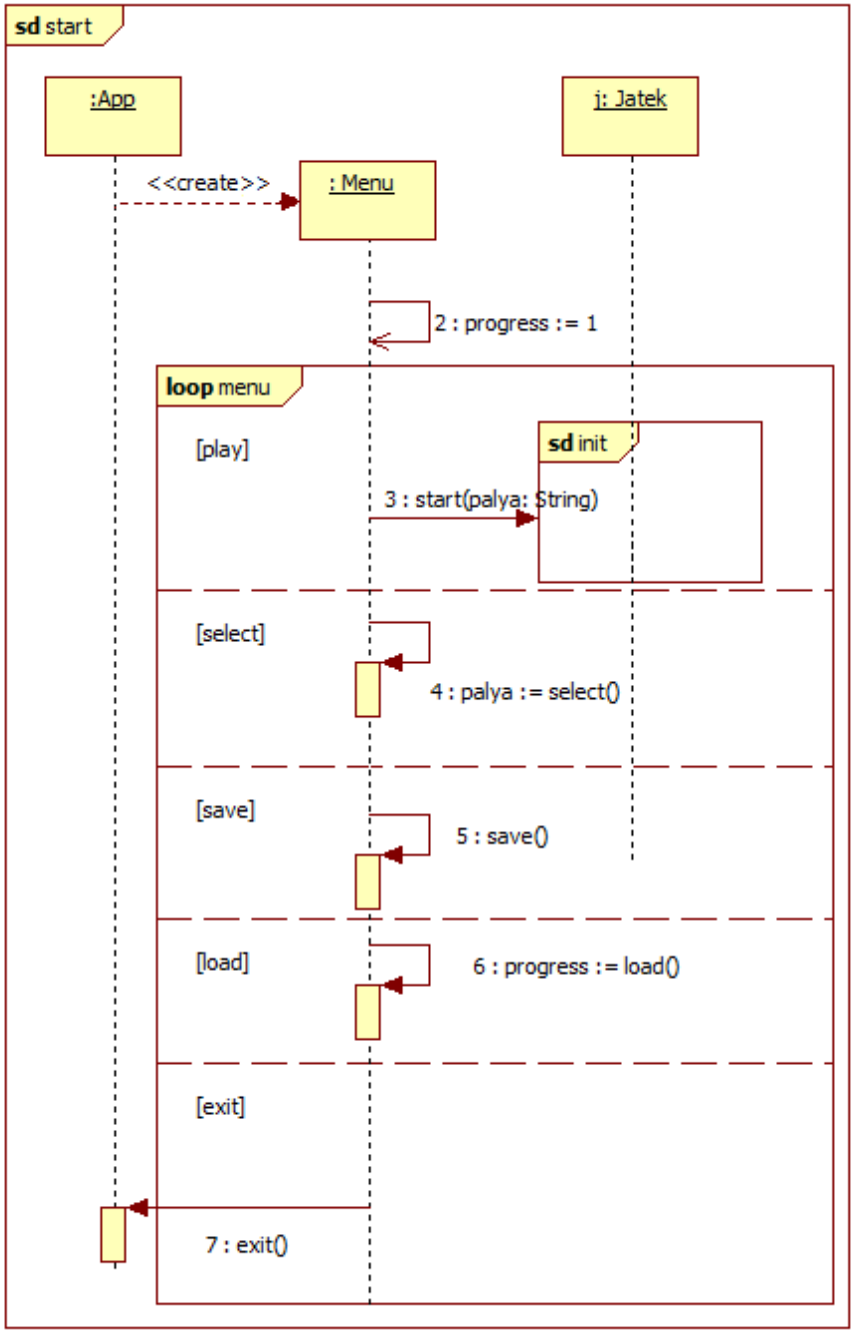
- **Interfészek**

Nincs.

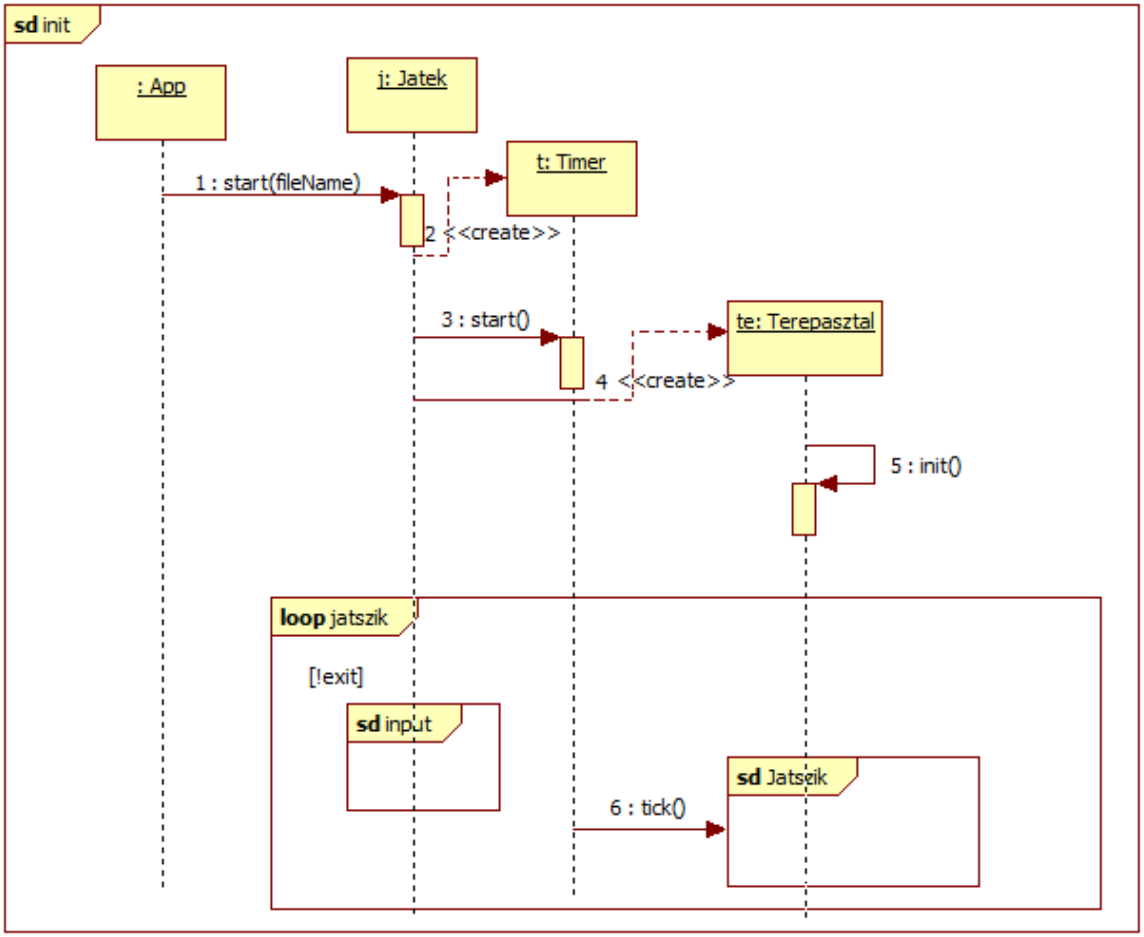
- **Attribútumok**
  - **kimenetek: List<SinElem> -** az összes kimenő SinElem referenciája
  - **aktivAg: SinElem -** melyik ág az aktív
- **Metódusok**
  - **void setAktualisAg(Sinelem s) -** aktivAg setter
  - **SinElem kovAg() -** sinek listából az aktivAg utáni SinElemet adja vissza
  - **void onInput() -** A bevitelre reagál az osztály.

3.4 Szekvencia diagramok

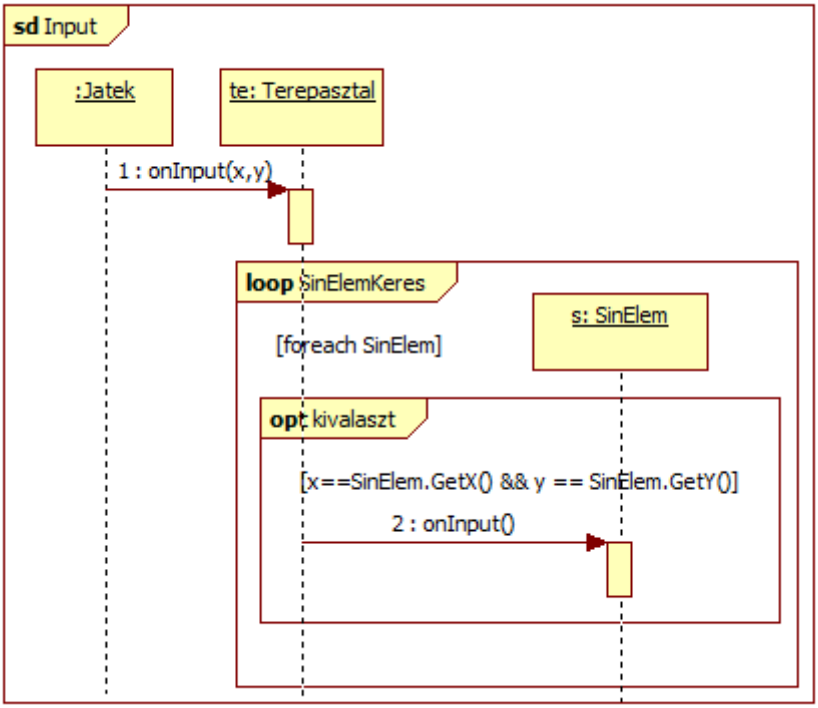
3.4.1 Start



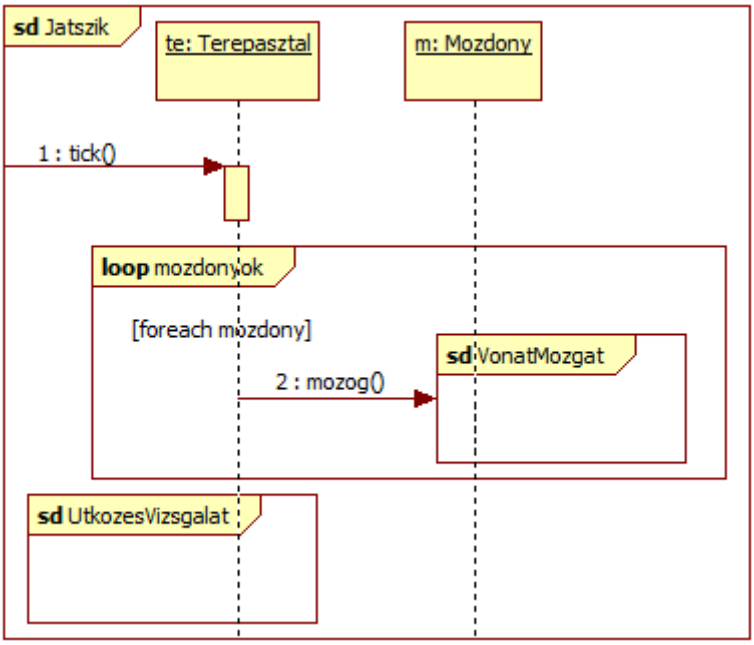
3.4.2 InitSzekvencia



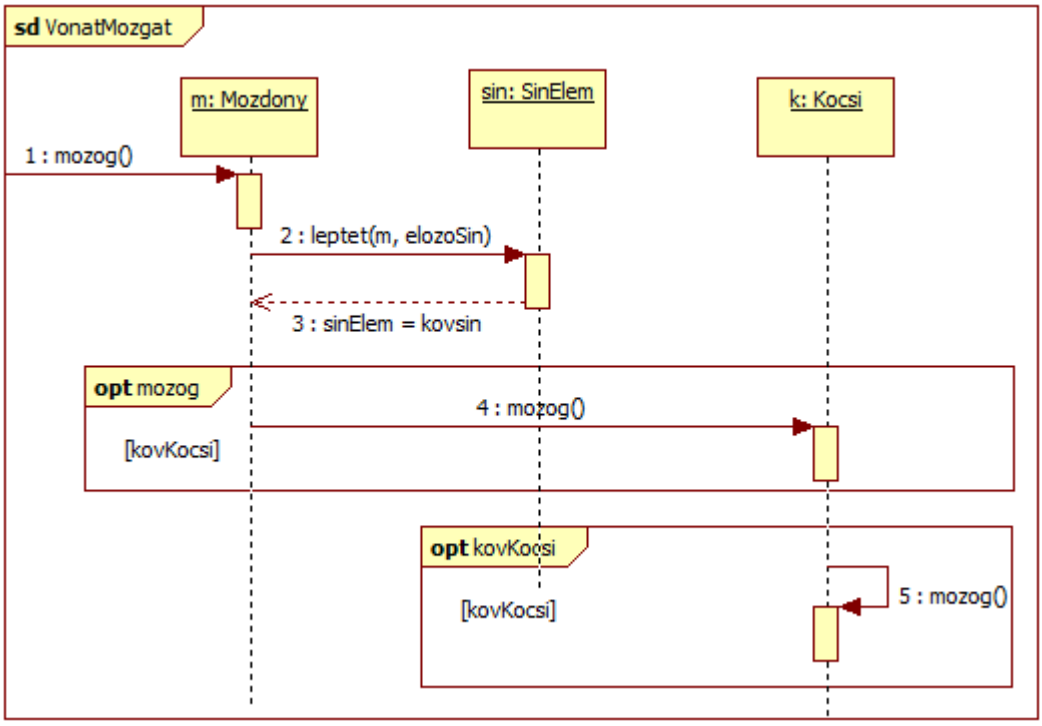
3.4.3 Input



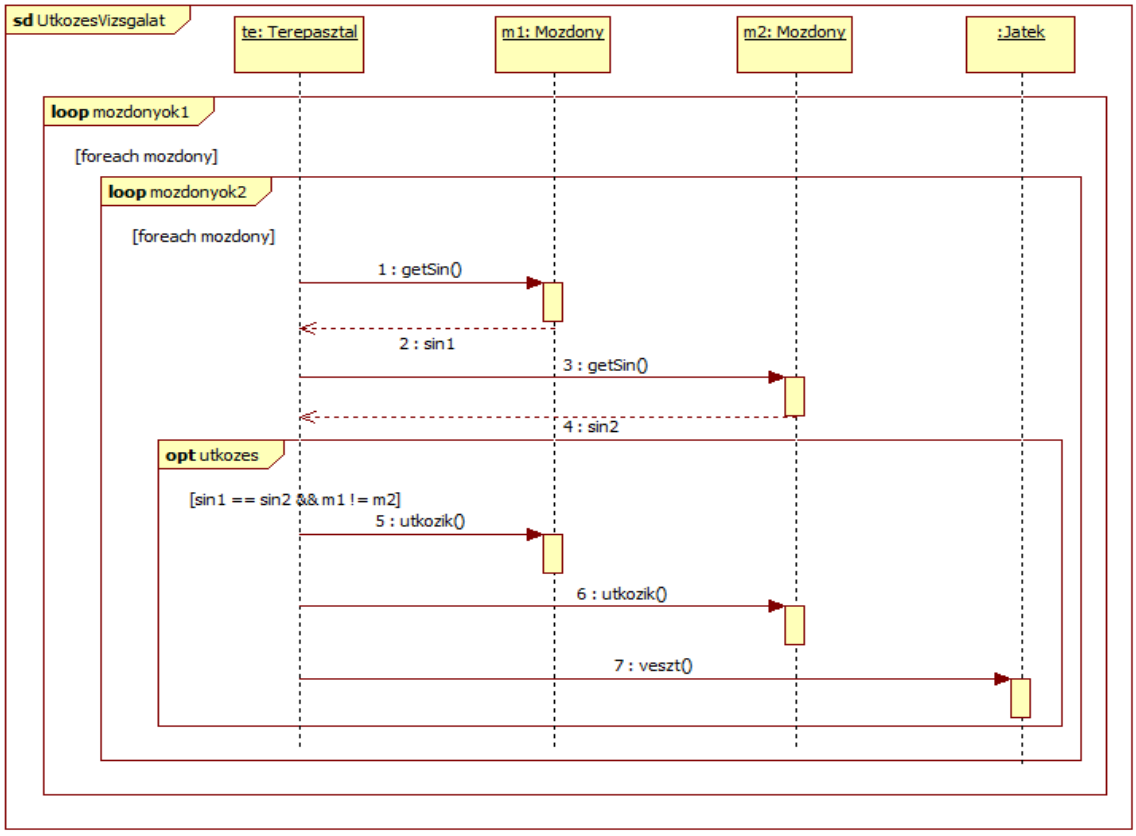
3.4.4 Jatszik



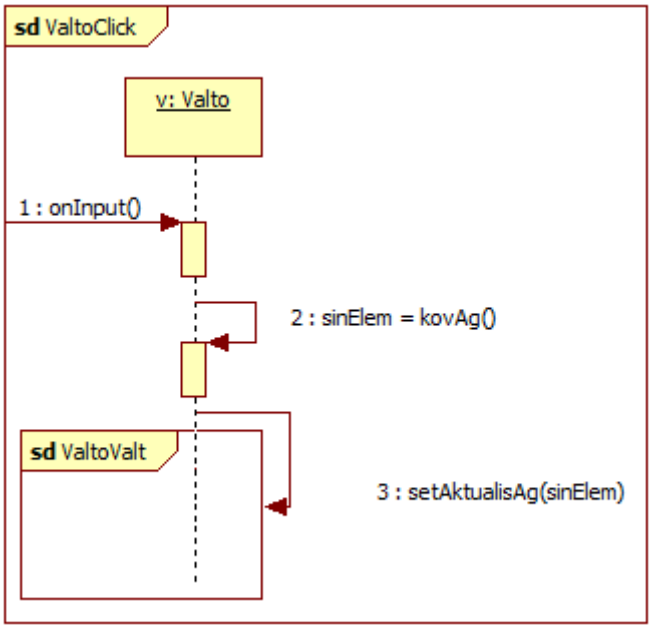
3.4.5 VonatMozgat



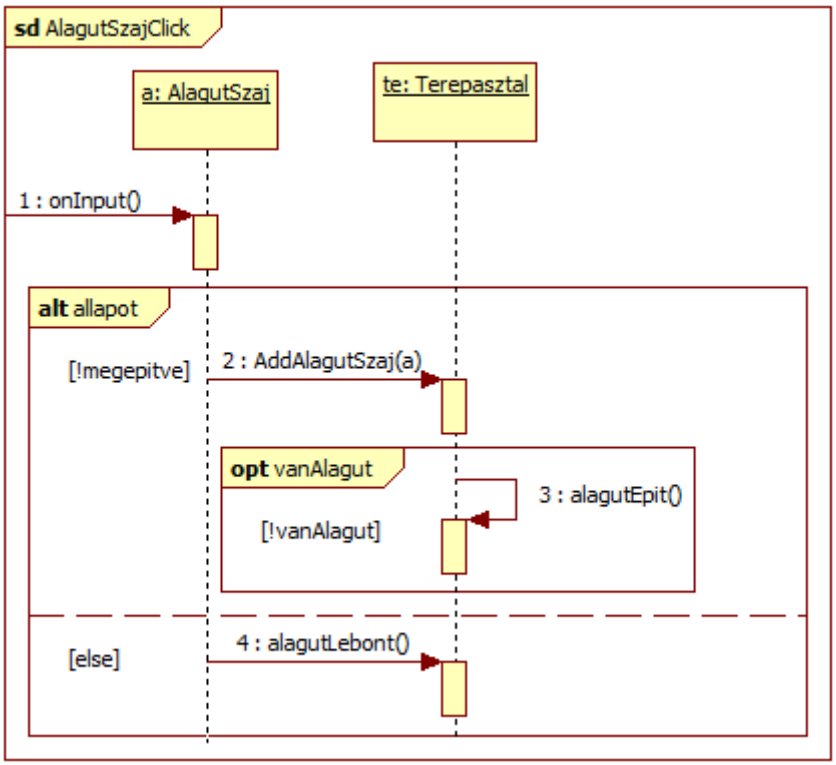
3.4.6 UtkozesVizsglat



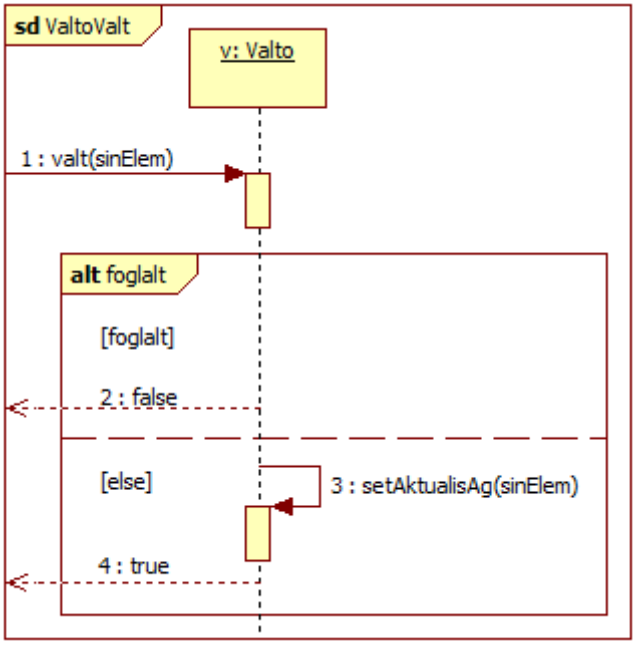
3.4.7 ValtoClick



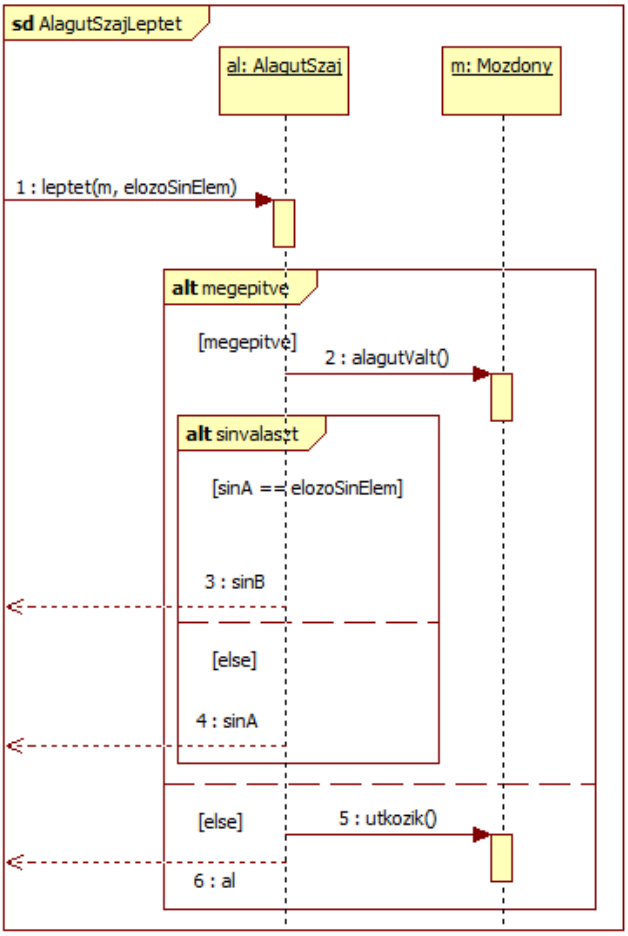
3.4.8 AlagutSzajClick



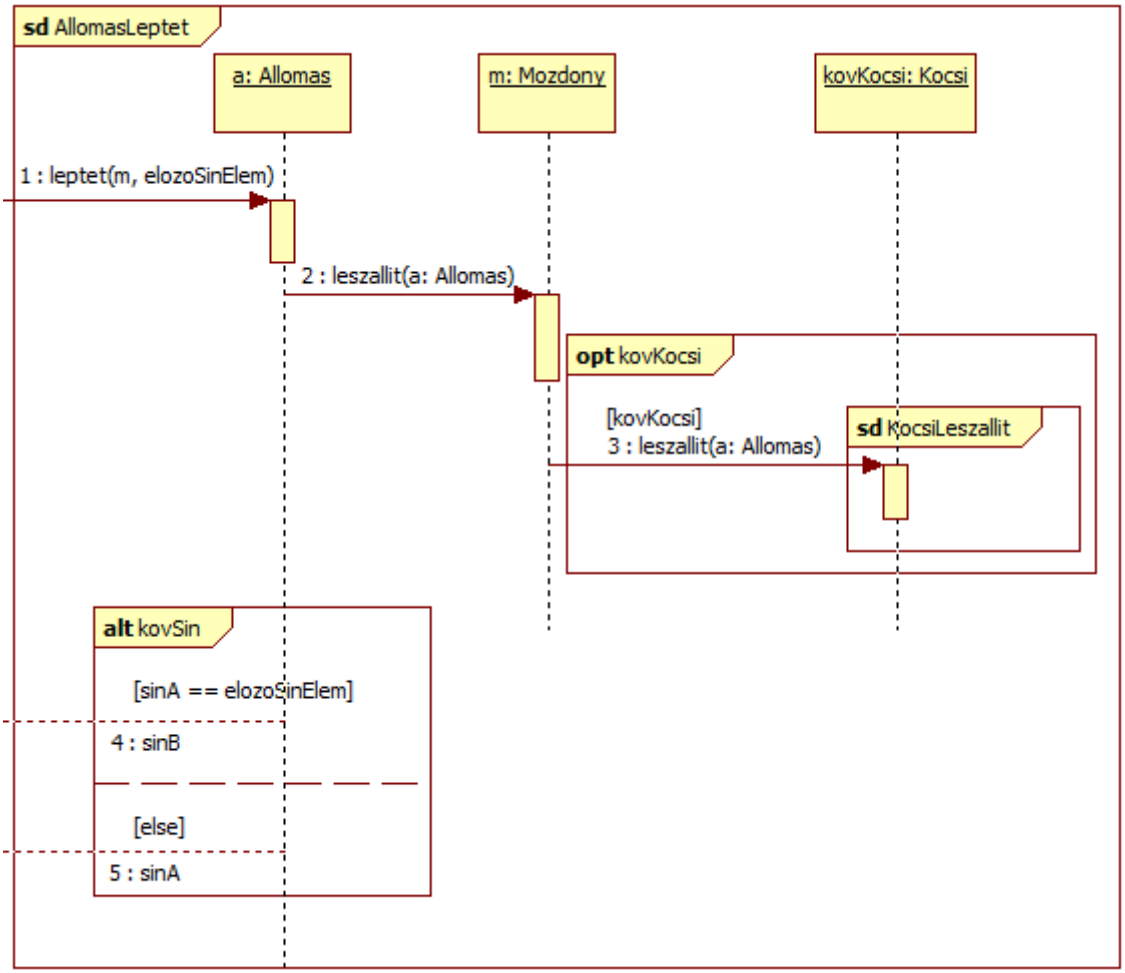
3.4.9 ValtoValt



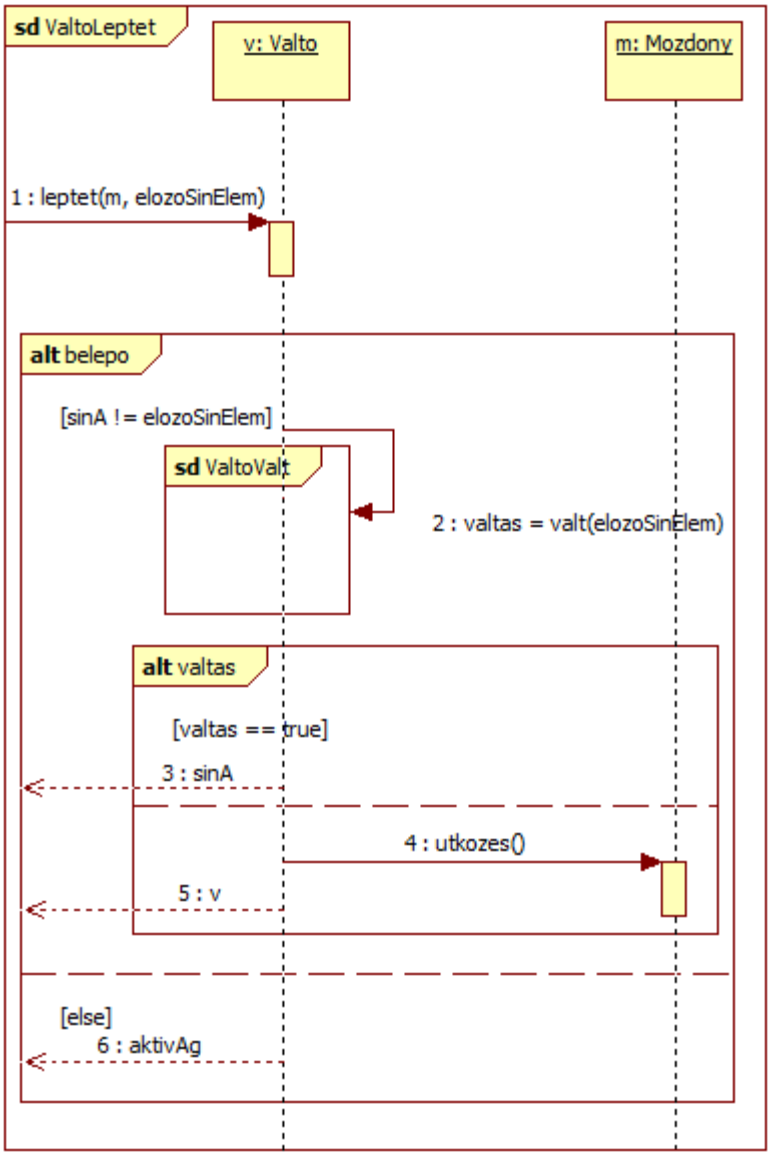
3.4.10 AlagutSzajLeptet



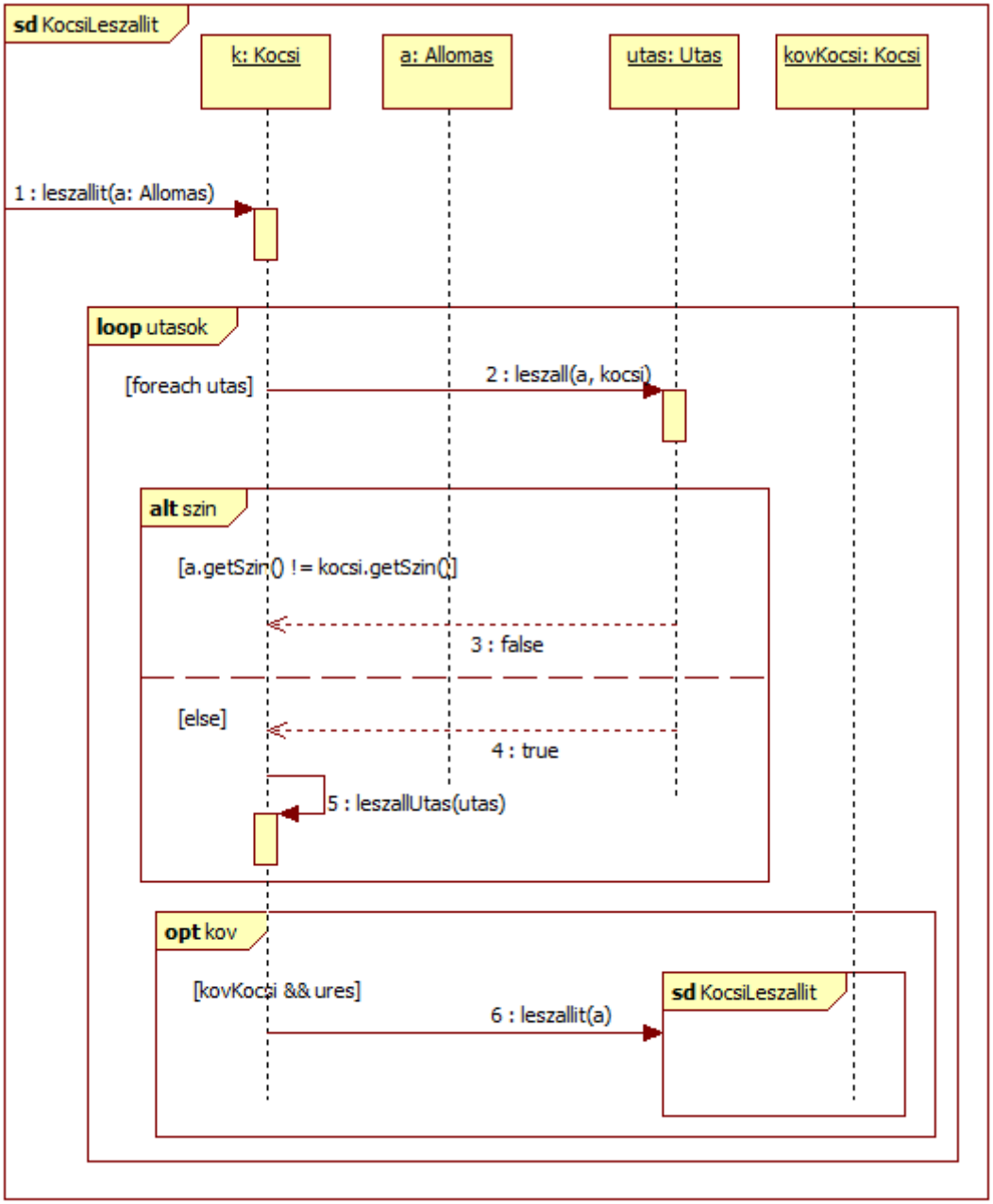
3.4.11 AllomasLeptet



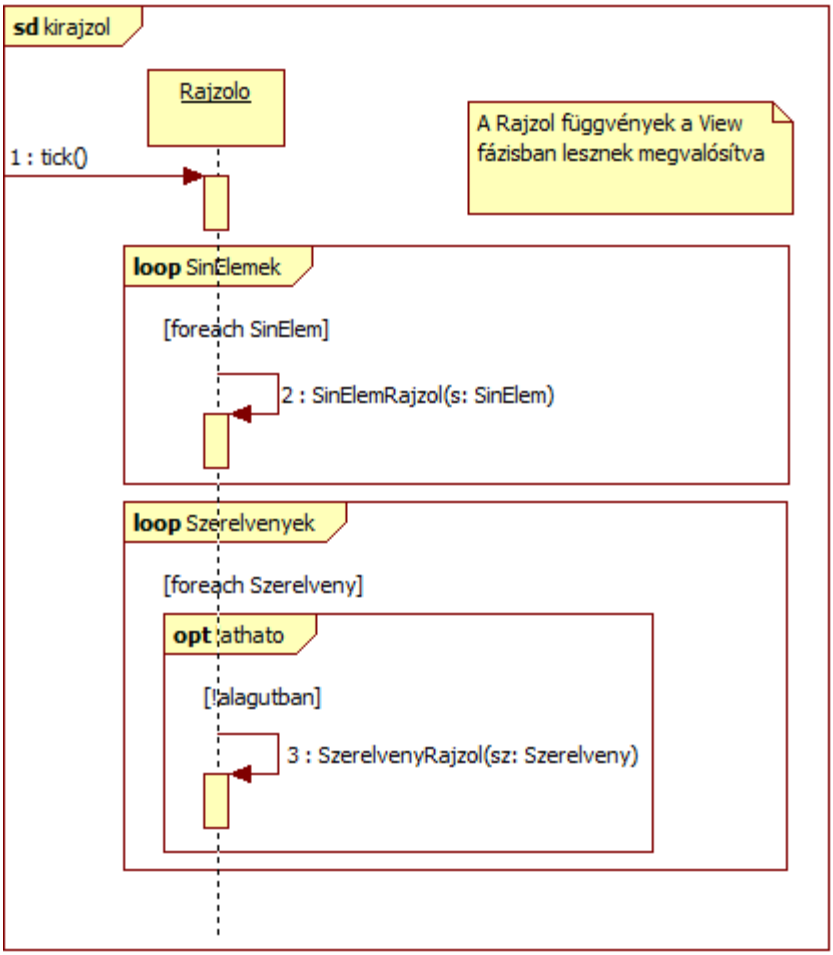
3.4.12 ValtoLeptet



3.4.13 KocsiLeszallit

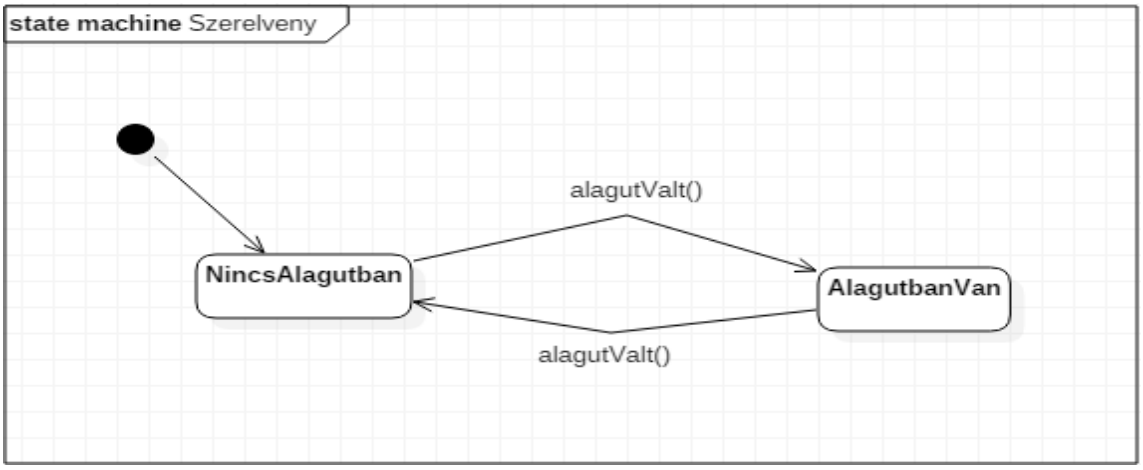


3.4.14 Kirajzol



3.5 State-chartok

3.5.1 Szerelveny



3.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2017.02.24. 16:30	4 óra	Dócs Krátky Varga	Struktúra diagram
2017.02.24. 16:30	2 óra	Szili	Objektum katalógus / osztályok
2017.02.25. 11:00	9 óra	Varga	Osztályok, osztály diagram
2017.02.25 11:00	7 óra	Dócs	Osztálydiagram, Szekvencia diagramok
2017.02.25 11:00	8 óra	Szili Sillye	Osztály- és szekvencia diagramok
2017.02.25 20:00	0.5 óra	Sillye Szili	Állapotdiagram
2017.02.25 16:00	5 óra	Krátky	Osztályok, szekvencia diagram
2017.02.26 11:30	11 óra	Krátky Sillye Szili	Szekvenciadiagramok, utómunkálatok
2017.02.26 11:30	7 óra	Dócs	git verziókezelés elsajátítása, Osztályok leírása
2017.02.26 14:00	8,5 óra	Varga	szekvencia diagram, finomítások
2017.02.26 19:00	1.5 óra	Dócs	Input és Rajzol szekvencia diagramok
2017.02.26 22:00	0,5 óra	Szili	dokumentáció formázása