

## 8. Részletes tervek

### 8.0 Javítások

#### 8.0.1 Bemeneti nyelv

##### *placepipe*

**Leírás:** A szerelő lehelyezi a nála lévő csövet, amelynek egyik végét ahhoz a csomóponthoz köti, amelyiken éppen áll. (Kiegészítettük a *pipe\_name* paraméterrel.)

**Opciók:** *placepipe* <mechanic\_name> <node\_name> <pipe\_name>

- *mechanic\_name* - A szerelő neve
- *node\_name* - A csomópont neve
- *pipe\_name* - A lehelyezett cső neve

**Példa:**

- *placepipe mechanic1 cistern1 pipe1*

#### 8.0.2 Kimeneti nyelv

##### *state*

Mechanic esetében történt egy apró javítás, nem a nála lévő pumpa és cső nevét írjuk ki, hanem azt, hogy van-e nála.

- *Mechanic*

Mechanic <name> on <position> has pipe: <true / false> has pump: <true / false>

## 8.1 Osztályok és metódusok tervei.

### 8.1.1 Cistern

- **Felelősség**

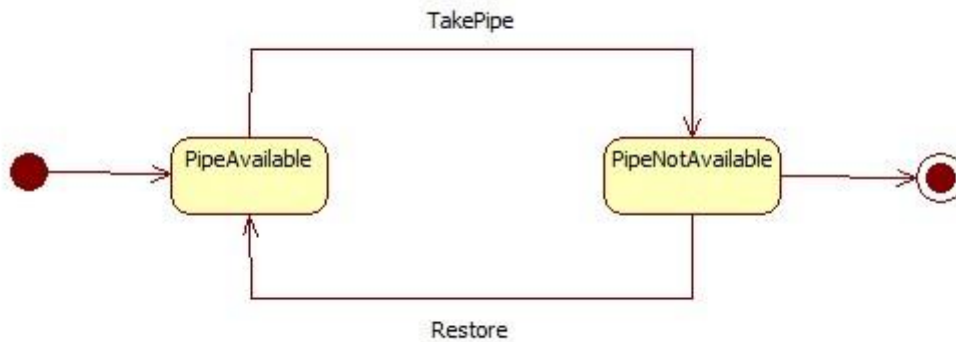
Ciszternákért felelős osztály. A ciszternába befolyt vizet, és az ott felvehető csőt, pumpát kezeli.

- **Ősosztályok**

Field → FieldNode.

- **Interfészek**

Tickable.

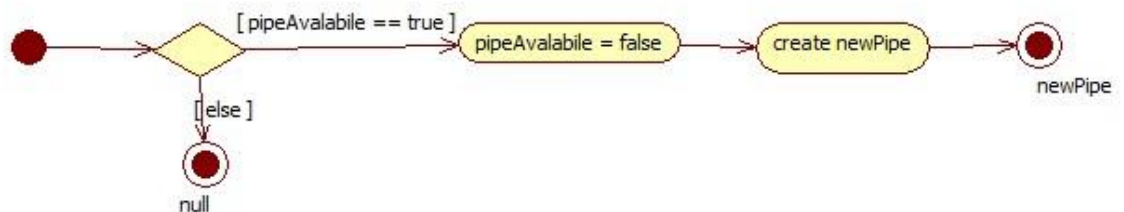


- **Attribútumok**

- **drainedWater**: a vízmennyiség ami befolyt a ciszternákba, aminek függvényében számolódnak a szerelő csapat pontjai.
- **pipeAvailable**: van-e elérhető cső, amelyet a szerelők felvehetnek az eszköztárukba.

- **Metódusok**

- **-void addDrainedWater(int amount)**: Növeli a ciszternába folyt vízmennyiséget a paraméterként kapott értékkel.
- **+Pipe takePipe()**: Egy cső felvétele a mezőről. Ha sikeres a cső felvétele akkor visszatér egy csővel, amely a szerelő eszköztárába kerül. Ha a felvétel sikertelen, akkor null értékkel tér vissza.



- **+Pump takePump()**: Egy pumpa felvétele a mezőről. A metódus létrehoz és visszatérít egy új pumpát.
- **+int getDrainedWater()**: Visszatéríti a ciszternába befolyt vízmennyiséget.
- **+void tick()**: Egy időegység elteltét jelenti. Az időegységnek megfelelő víz befolyik a ciszternához kapcsolt csövekből a ciszternába. Ezzel a mennyiséggel nő a ciszternába befolyt víz mennyisége.

```

SET pipeAvalabile := true
For each pipe in the pipes:
    SET drained := pipe.drain(MAX_VALUE)
    Call addDrainedWater(drained)
End loop.
  
```

### 8.1.2 Field

- **Felelősség**

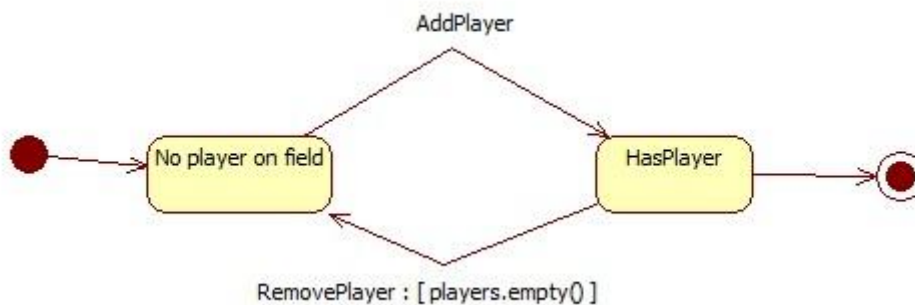
A pályaelemekért felelős őket reprezentáló absztrakt osztály.

- **Ősosztályok**

-

- **Interfészek**

-



- **Attribútumok**

- **#List<Player> players:** a pályaelemen álló játékosok listája.

- **Metódusok**

- **+Field addPlayer(Player p):** Ha a paraméterként kapott játékos rálép a pályaelemre, akkor ezt hozzáadja a pályaelemhez és a játékosok listájához.
- **+boolean removePlayer(Player p):** A paraméterként kapott játékos ellép a pályaelemről, ezt kiveszi a játékosok listájából és visszatérít egy true értéket.
- **+int getNumberOfPlayers():** Visszatéríti a pályaelemen álló játékosok számát.
- **+Pipe takePipe():** Egy cső felvétele a mezőről, amely alapértelmezett értéként null - t térít vissza, hiszen nem lehet akármelyik pályaelemről csövet felvenni.
- **+Pipe takePump():** Egy pumpa felvétele a mezőről, amely alapértelmezett értéként null - t térít vissza, hiszen nem lehet akármelyik pályaelemről pumpát felvenni.

### 8.1.3 FieldNode

- **Felelősség**

Egy absztrakt osztály amely olyan csomópontot reprezentál amihez csöveket lehet csatlakoztatni.

- **Ősosztályok**

Field.

- **Interfészek**

-

- **Attribútumok**

- **#List<Pipe>pipes:** A csomóponthoz csatlakoztatott csövek listája.

- **Metódusok**

- **+void connect(Pipe p):** A paraméterként kapott csövet a csomóponthoz csatlakoztatja, és hozzáadja ezt a csövek listájához.
- **+void disconnect(Pipe p):** Lecsatlakoztatja a csomópontról a paraméterként kapott csövet, és törli ezt a csövek listájából.

### 8.1.4 Mechanic

- **Felelősség**

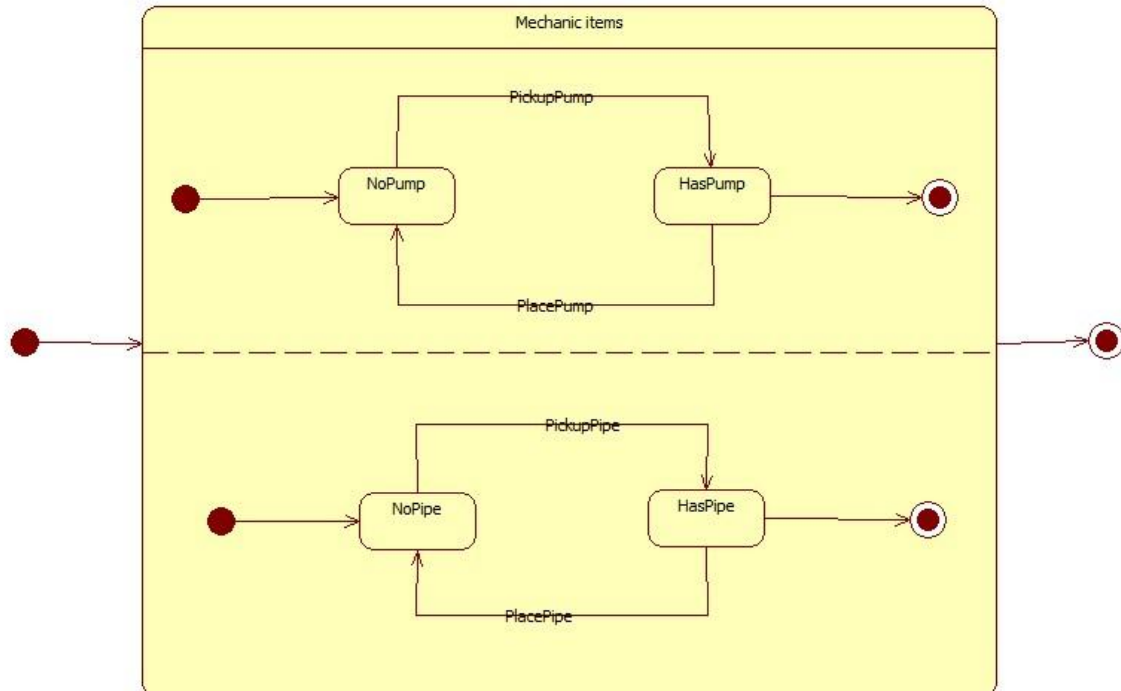
A szerelőkért felelős osztály, céljuk, hogy minél több vizet szállítsanak a forrástól a ciszternáig.

- **Ősosztályok**

Player

- **Interfészek**

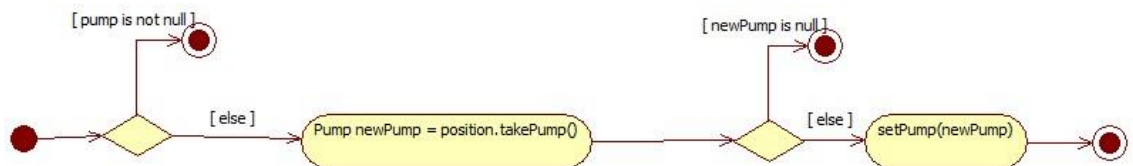
-



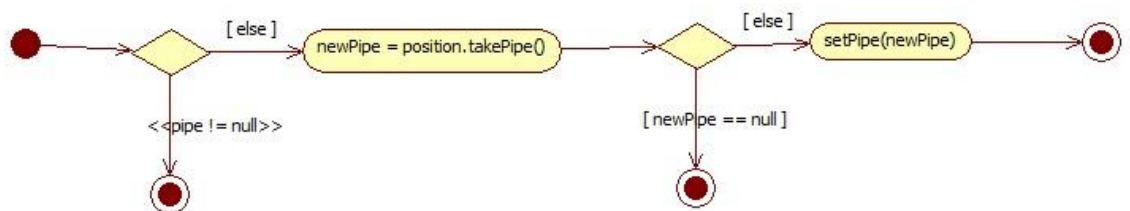
- **Attribútumok**

- **-Pipe pipe:** a szerelő eszköztárában lehet egy vagy nulla cső, amelyet majd lehelyezhet

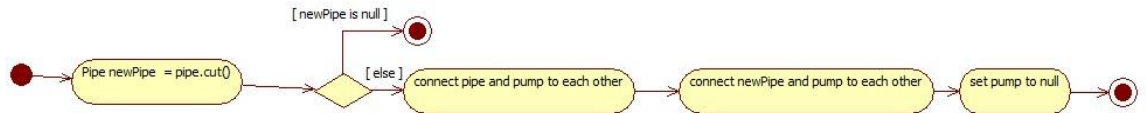
- **-Pump pump:** a szerelő eszköztárában lehet egy vagy nulla pumpa, amelyet majd lehelyezhet
- **Metódusok**
  - **+void fixPipe(Pipe p):** A szerelő megjavít egy kilyukasztott csövet. Ezután a csövön keresztül, a víz átjut egyik pumpától a másikig, anélkül, hogy elfolyjon a homokba. A megjavítás után a csövet egy adott ideig nem lehet újra lyukasztani. A metódus meghívja a paraméterként kapott pipe *repair()* metódusát.
  - **+void fixPump(Pump p):** A szerelő megjavít egy meghibásodott pumpát. A pumpa ezután képes lesz a csöveken tovább pumpálni a vizet más csövek felé. A metódus meghívja a paraméterként kapott pump *repair()* metódusát.
  - **+void connectPipe(Pipe p, FieldNode n):** A paraméterként kapott csőnek egyik végét felcsatlakoztatja a paraméterként kapott csomópontra. A cső másik fele szabadon marad.
  - **+void disconnectPipe(Pipe p, FieldNode n):** A paraméterként kapott csomópontra csatlakoztatott csőnek azt a végét, amelyik arra a pumpára van csatlakoztatva, amelyiken a szerelő áll, lecsatlakoztatja.
  - **+void pickupPump():** A szerelő felvesz egy pumpát a ciszternától. A felvétel megengedett, ha a szerelőnél nincs pumpa. Ez a pumpa a szerelő eszköztárába kerül, majd ezt a pályán bárhol lehelyezheti, ahol ez megengedett.



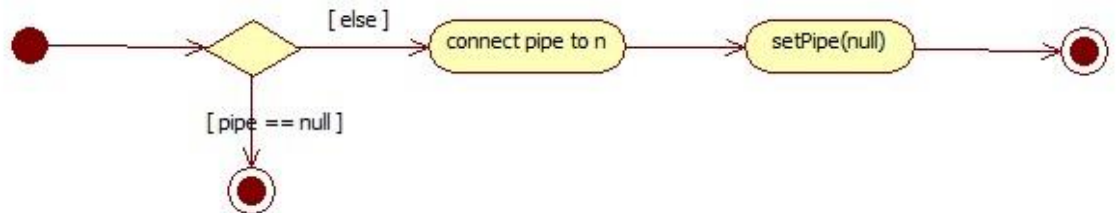
- **+void pickupPipe():** A szerelő felvesz egy csövet a ciszternától. Ha már rendelkezik csővel, akkor a felvétel nem megengedett, viszont ha nincs, akkor a felvett cső az eszköztárába kerül, amelyet lehelyezhet egy pumpánál.



- **+void placePump(Pump pump, Pipe pipe):** A szerelő letesz egy pumpát, amely eddig az eszköztárában volt. A paraméterként kapott csőre letett pumpa a csövet kettéválasztja, egy új cső létrejön és a megfelelő végeit a régi és az új csőnek, rácsatlakoztatja a pumpára. Ezután az eszköztára kiürül és felvehet egy más pumpát az elkövetkezendő körökben.



- **+void placePipe(FieldNode n):** A szerelő letesz egy csövet, amely az eszköztárában volt. Egyik végét arra a pumpára köti amelyiken áll, a másik vége szabadon lóg. Ezután az eszköztára kiürül.



- **+void setPump(Pump pump):** Beállítja a paraméterként kapott pumpát a szerelő eszköztárába lévő pumpának.
- **+void setPipe(Pipe pipe):** Beállítja a paraméterként kapott csövet a szerelő eszköztárába lévő csőnek.

### 8.1.5 Pipe

- **Felelősség**

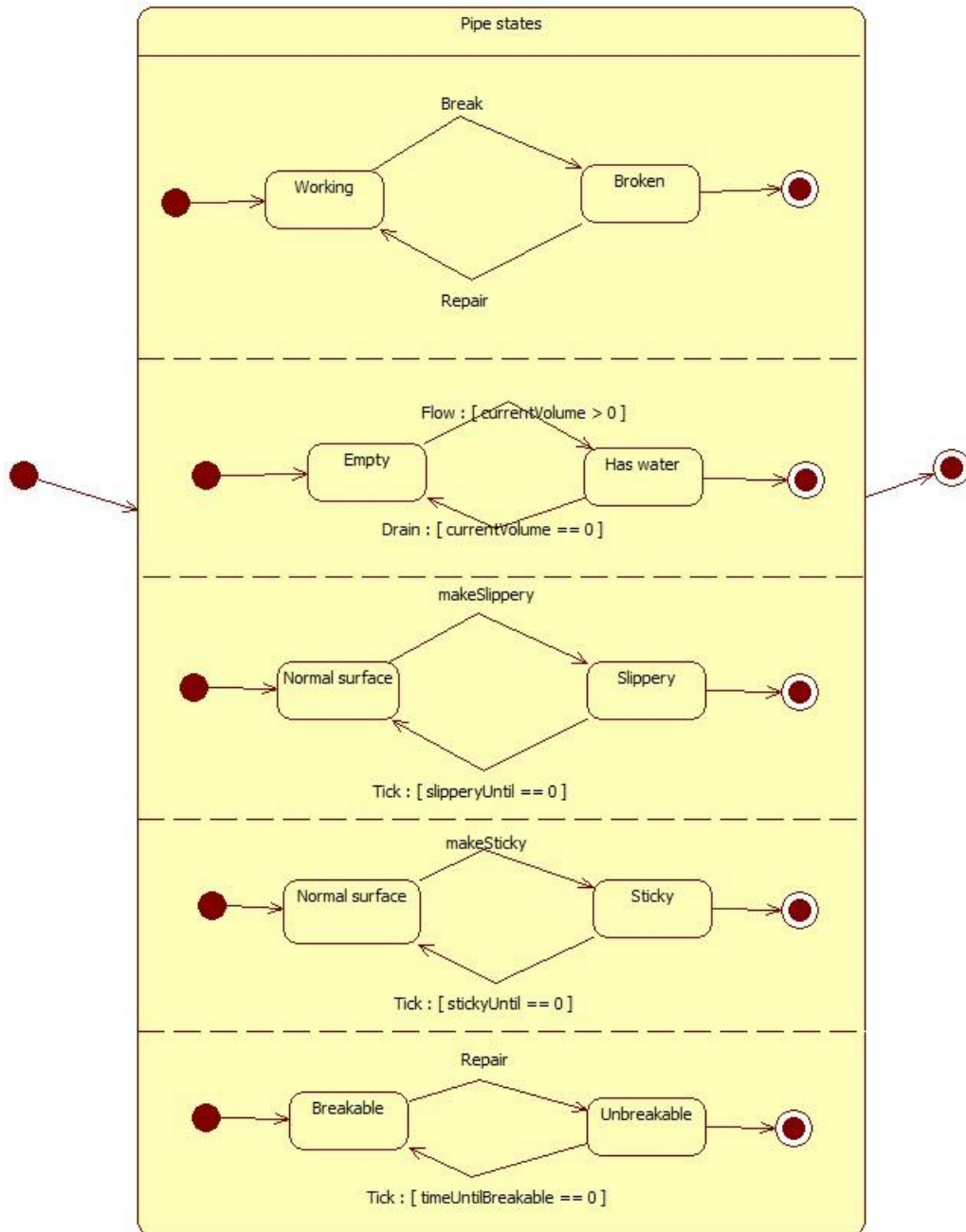
A csövekért, valamint a víz szállításáért felelős osztály. A játékosok ezen a pályaelemen is közlekednek, de egyszerre csak egy játékos lehet egy csövön.

- **Ősosztályok**

Field.

- **Interfészek**

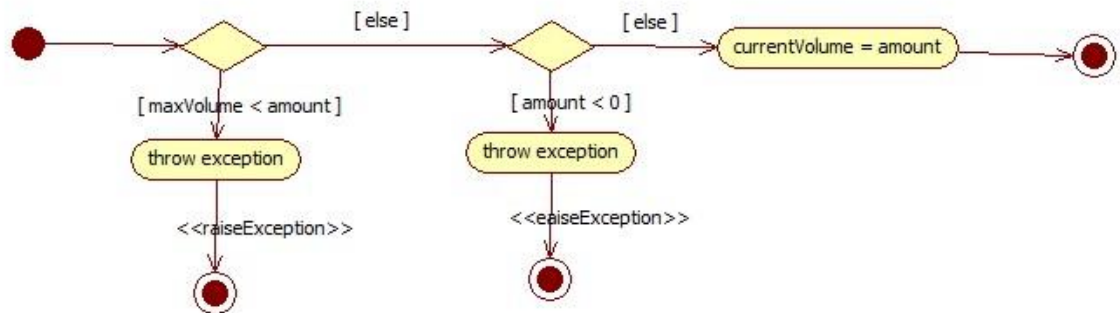
Tickable.



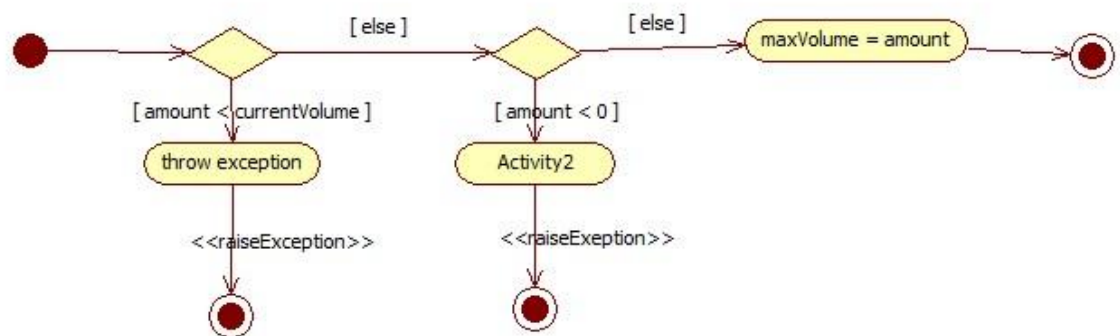
- **Attribútumok**

- **-int MAX\_VOLUME:** a cső maximális kapacitása
- **-int SLIPPERY\_TIME:** az idő amíg egy cső csúszós lesz
- **-int STICKY\_TIME:** az idő amíg egy cső ragadós lesz
- **-int MAX\_BREAKABLE\_TIME:** a maximális idő, amíg újra lehet lyukasztani
- **-Random random:** a random időértékek generálásához szükséges
- **-boolean isBroken:** a cső állapota, ha lyukas-e vagy sem.
- **-int maxVolume:** a cső maximális kapacitása
- **-int currentVolume:** a csővön érkező aktuális vízmennyiség
- **-int wastedWater:** a lyukakon kifolyt vízmennyiség
- **-int timeUntilBreakable:** a hátramaradt idő, amíg újra lehet a csövet lyukasztani
- **-int slipperyUntil:** a hátramaradt idő, amíg a cső csúszós

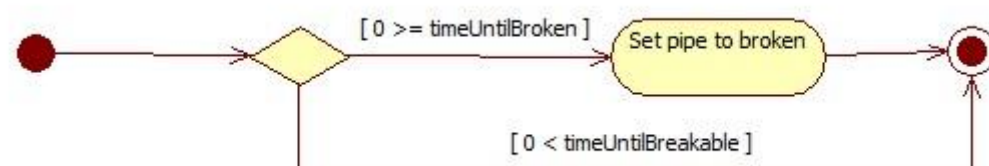
- **-int stickyUntil:** a hátramaradt idő, amíg a cső ragadós
- **-List<FieldNode> ends:** lista, amely a cső végeit reprezentálja, amelyek csomópont típusúak
- **Metódusok**
  - **+void setWaterVolume(int amount):** Beállítja az aktuális vízmennyiséget a csőbe.



- **+void setMaxVolume(int amount):** Beállítja a cső maximális kapacitását.

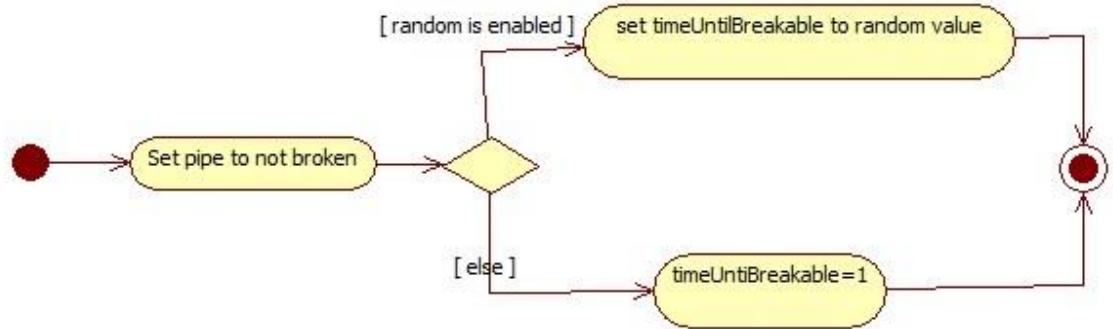


- **+void makeSlippery():** A cső egy adott ideig csúszossá válik. Ha egy játékos rálép, akkor átcúszik arra a pályaelemre, amelyikre a cső másik vége csatlakoztatva van.
- **+void makeSticky():** A cső egy adott ideig ragadóssá válik. Ha valamelyik játékos rálép, akkor rövid ideig nem tud a pályaelemről továbblépni. Minden tick után az idő, ameddig még ragadós a cső, 1-gyel csökken.
- **+void breakPipe():** Ha az idő amíg újra lehet lyukasztani lejárt, akkor kilyukasztja a csövet, úgy, hogy az összes benne lévő és rajta átfolyó víz kifolyik belőle. A cső lyukas állapota ezután igaz lesz.

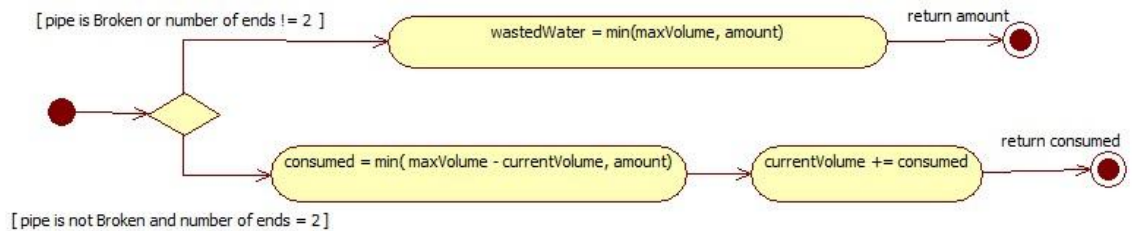




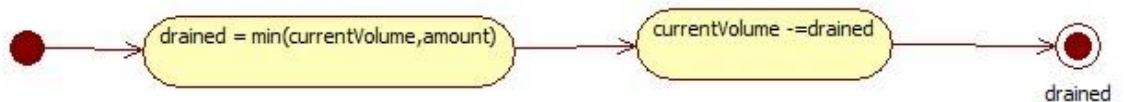
- **+void repair():** Megjavítja a kilyukadt csövet. Ezután a cső lyukas állapota hamis lesz, és egy 0 és 10 közötti értékkel beállítja, hogy mennyi ideig nem lehet újra lyukasztani a csövet.



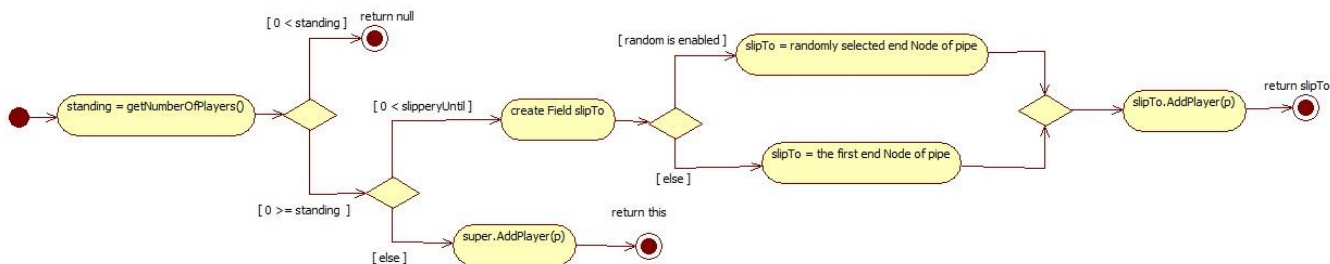
- **+int flow(int amount):** A paraméterként kapott mennyiséget a csőbe töltjük, és ha a cső lyukas vagy nincs mindkét fele rácsatlakoztatva egy csomópontra, akkor ez a víz elfolyik. Ha sikeres a vízbetöltés akkor visszatér azzal a mennyiséggel amelyik átfolyt a csővön.



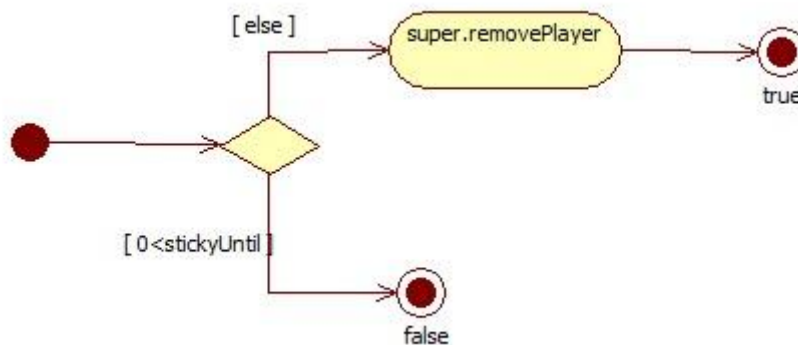
- **+int drain(int amount):** A paraméterként kapott mennyiséget kiszívjuk a csőből. Visszatérünk azzal az értékkel amit sikeresen kiszívtunk.



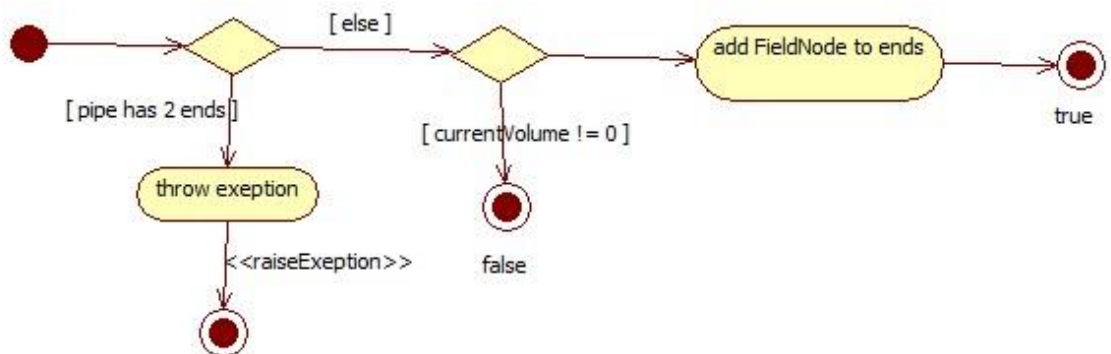
- **+Field addPlayer(Player p):** Ha a paraméterként kapott játékos a pályaelemre akar lépni, akkor ezt hozzáadja a pályaelem játékosok listájához. Ha a csővön már tartózkodik egy másik játékos, akkor ez a lépés nem lehetséges. Ha a cső csúszós és a játékos rá akar lépni, akkor átcúszik a cső másik végére, ha ez csatlakoztatott valamely pályaelemhez.



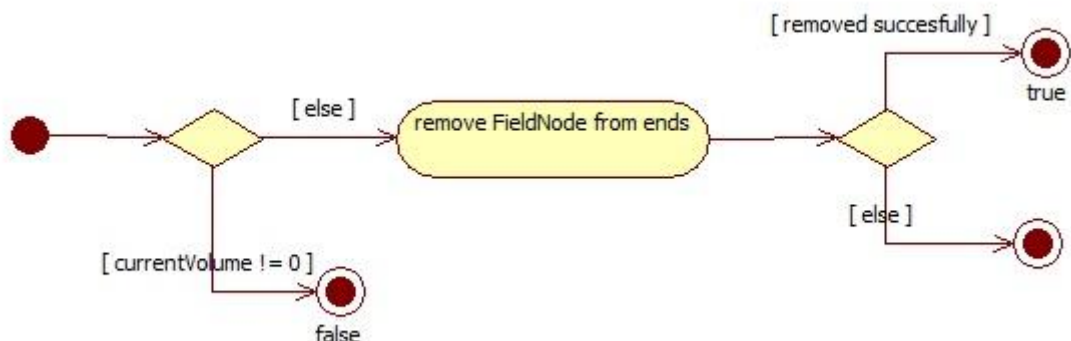
- **+boolean removePlayer(Player p):** A paraméterként kapott játékos ellép a pályaelemről és ezt kitörli a játékosok listájából. Ha a cső amiről el akar lépni a játékos ragadós, akkor azt nem teheti meg, addig amíg a cső ragadós ideje le nem jár.



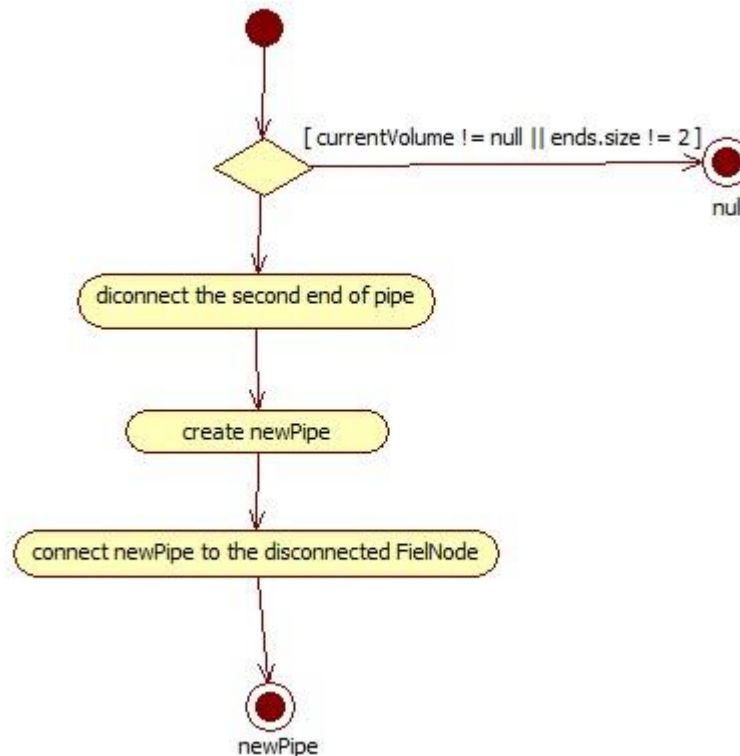
- **+boolean connect(FieldNode n):** A paraméterként kapott csomópontot beállítja a cső egyik végének, és visszatéríti, ha sikerült-e a csomópont csatlakoztatása. Ha a csőben van víz, akkor nem lehetséges ez.



- **+boolean disconnect(FieldNode n):** Eltávolítja a paraméterként kapott csomópontot a cső végei közül és visszatéríti, ha sikerült-e a csomópont eltávolítása. Ha a csőben van víz akkor ez lehetséges.



- **+Pipe cut():** Félbevágja a csövet, ezzel egy új csövet létrehozva, majd visszatéríti az újonnan létrejött csövet, ha sikeres volt a félbevágás, különben null értékkel tér



vissza.

- **+void tick():** Egy időegység elteltét jelenti. Az idő amíg újból lehet lyukasztani, amíg ragadós vagy amíg csúszós, csökken.

```

IF 0 < timeUntilBreakable
    timeUntilBreakable := timeUntilBreakable - 1
END IF
IF 0 < slipperyUntil
    slipperyUntil := slipperyUntil - 1
END IF
IF 0 < stickyUntil
    stickyUntil := stickyUntil - 1
END IF
  
```

### 8.1.6 Player

- **Felelősség**

A játékosokért felelős és azokat megvalósító absztrakt osztály, amiből két további osztály származik: `Mechanic` és `Saboteur`.

- **Ősosztályok**

-

- **Interfészek**

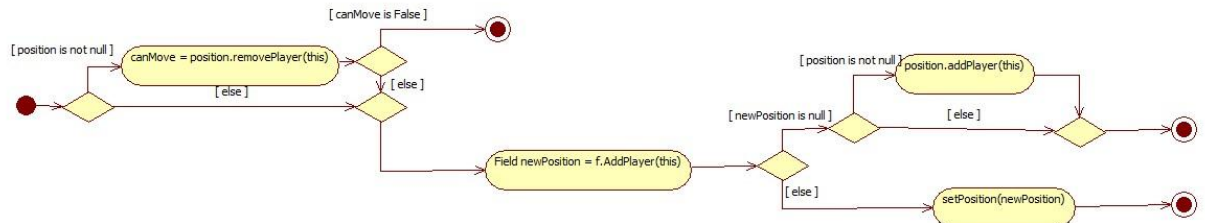
-

- **Attribútumok**

- **# Field position:** a játékos pozíciójának tárolására használt attribútum

- **Metódusok**

- **+void moveTo(Field f):** A játékos pozíciót vált. Ha nem tud egy adott mezőre átlépni, akkor a helyén marad, viszont ha sikerül akkor az új pozíciója az a mező lesz, amelyikre lépett.



- **+void SetPumpDirection(Pump pump, Pipe from, Pipe to):** A játékos a paraméterként kapott pumpát átállítja, hogy melyik csőből melyikbe pumpálja a vizet.
- **+void setPosition(Field f):** Beállítja a játékos pozícióját a paraméterként megadott mezőre.
- **+void breakPipe(Pipe p):** A játékos egy paraméterként kapott csövet kilyukaszt. Ezután a cső állapota lyukas lesz, és a kilyukasztott csőből a víz a homokba folyik.
- **+void makeSticky(Pipe p):** A játékos ragadóssá teszi a paraméterként kapott csövet. Ezután a cső állapota ragadós lesz 3 időegységig (tick), és ha valamelyik játékos rálép, akkor rövid ideig nem tud a pályaelemről továbblépni.

### 8.1.7 Pump

- **Felelősség**

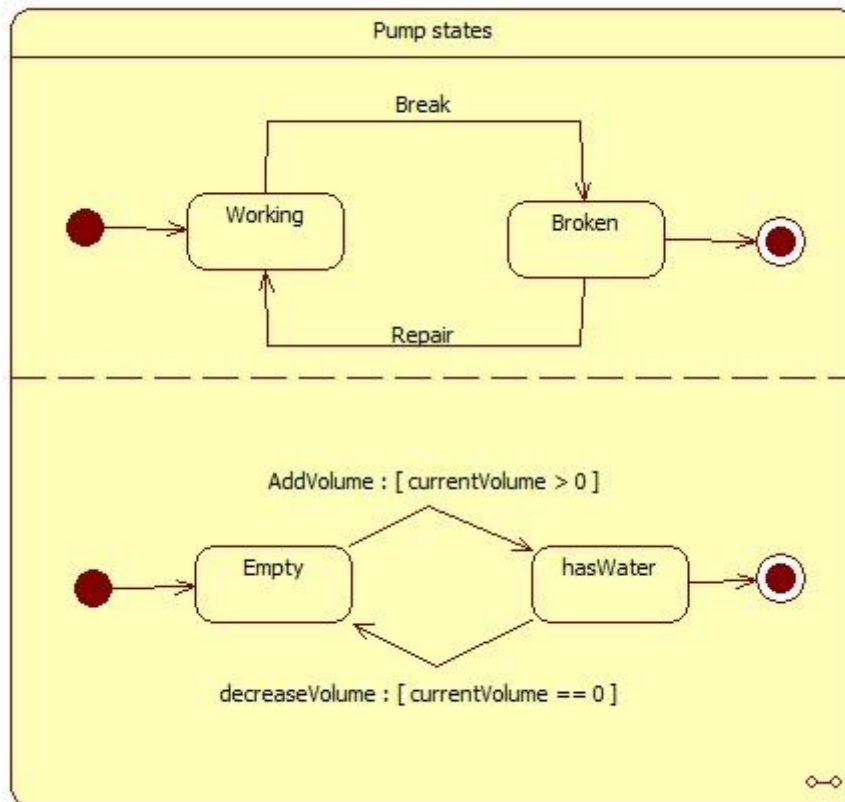
A víz pumpálásáért felelős osztály.

- **Össztályok**

Field → FieldNode

- **Interfészek**

Tickable.

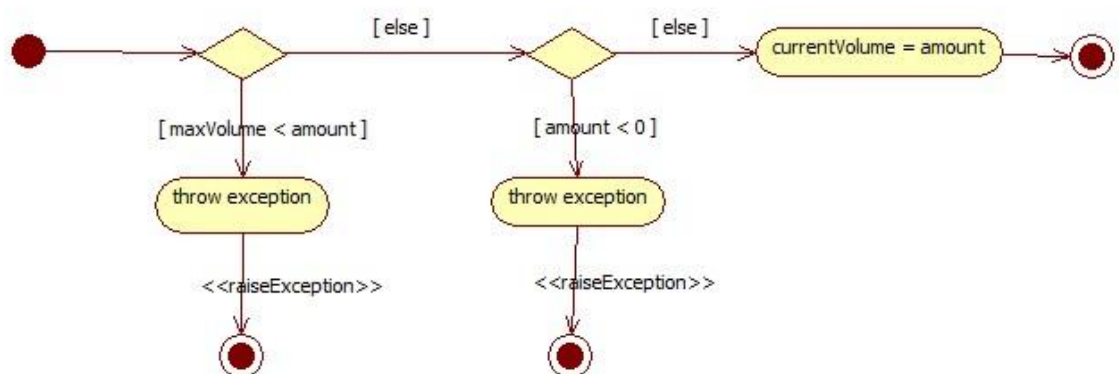


- **Attribútumok**

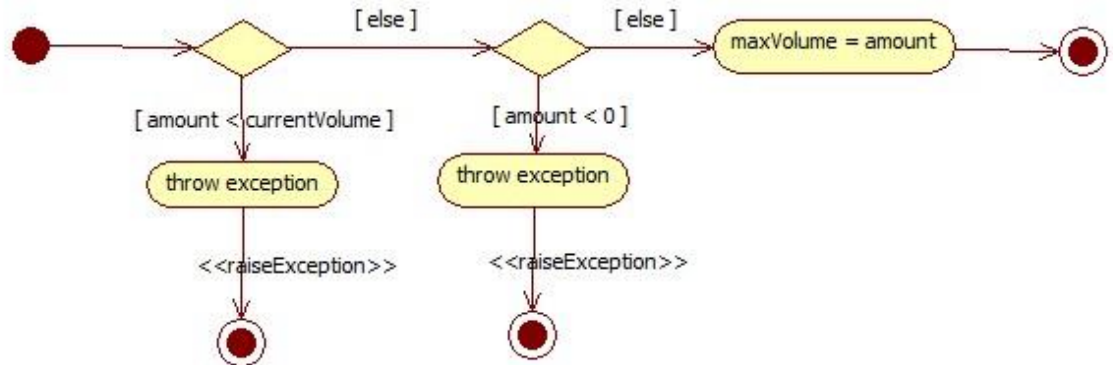
- **-Random random:** random értékek generálásához szükséges
- **-int MAX\_VOLUME:** a pumpa tartályának maximális kapacitása
- **-boolean isBroken:** a pumpa állapotát mutatja, ha meghibásodott-e.
- **-int maxVolume:** a pumpa tartályába tölthető maximális vízmennyiség.
- **-int currentVolume:** a pumpa tartályában lévő vízmennyiség.
- **-Pipe pipeIn:** a pumpa bemeneti csöve, amelyből érkezik a víz.
- **-Pipe pipeOut:** a pumpa kimeneti csöve, amelyiken keresztül továbbítódik a víz.

- **Metódusok**

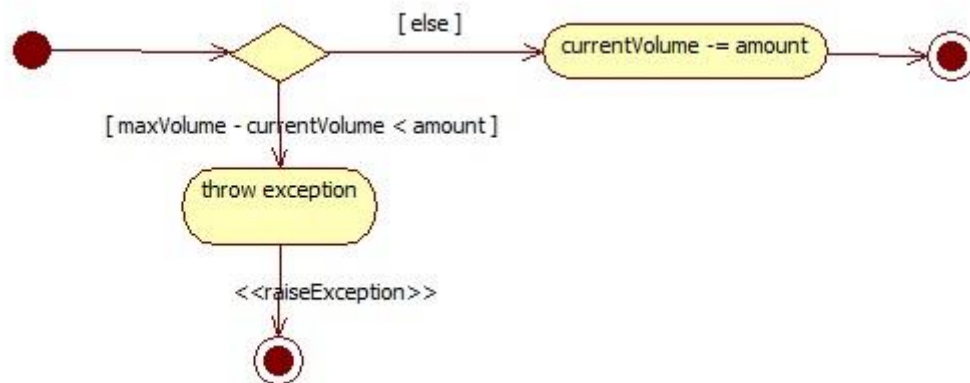
- **+void setWaterVolume(int amount):** Beállítja a paraméterként kapott értékkel a pumpa tartályában lévő aktuális vízmennyiséget.



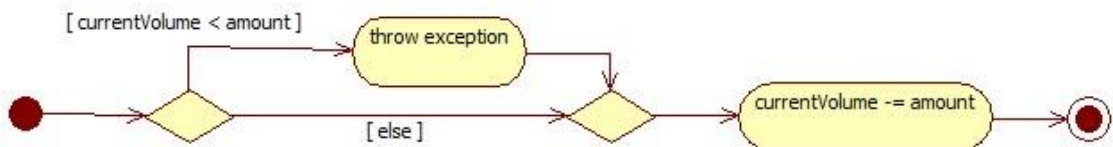
- **+void setMaxVolume(int amount):** Beállítja a paraméterként kapott értékkel a pumpa tartályának a maximális kapacitását.



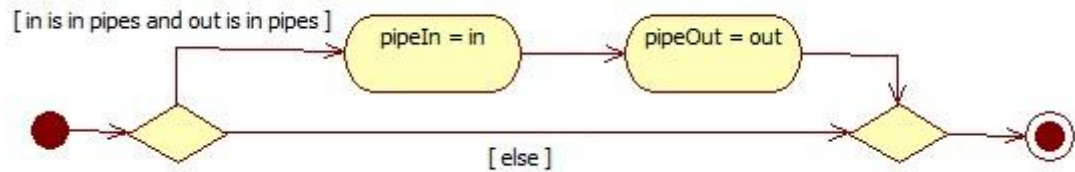
- **-void addVolume(int amount):** Növeli a pumpa tartályában lévő vízmennyiséget a paraméterként kapott értékkel.



- **-void decreaseVolume(int amount):** Csökkenti a paraméterként kapott értékkel a pumpa tartályában lévő vízmennyiséget.



- **+void breakPump():** A pumpa meghibásodik, ezután ez használhatatlanná válik, nem képes vizet pumpálni, addig amíg egy szerelő meg nem javítja.
- **+void repair():** A pumpa megjavítása, ezután már használható, újra pumpálhat vizet, egyik csőből egy másikba.
- **+void changeFlow(Pipe in, Pipe out):** Megváltoztatja, hogy melyik csőből melyikbe pumpálja a vizet. A paraméterként kapott *in* csövön fog érkezni a víz, és a paraméterként kapott *out* csövön továbbítódik.



- **+void tick():** Egy időegység elteltét jelenti. A pumpa véletlenszerűen elromolhat. Amennyiben a pumpa nincs elromolva, akkor a bemeneti csőből a kimeneti csőbe pumpál adott mennyiségű vizet.

```

IF isBroken
    return
END IF
  
```

```

IF isRandom
    break := generate random number between 0 and 1
    IF break = 1
        CALL breakPump()
    END IF
END IF
  
```

```

IF pipeIn != null
    drained := pipeIn.drain(maxVolume - currentVolume)
    CALL addVolume(drained)
END IF
  
```

```

IF pipeOut != null
    pushedOut := pipeOut.flow(currentVolume)
    CALL decreaseVolume(pushedOut)
END IF
  
```

### 8.1.8 Saboteur

- **Felelősség**

A szabotőrökért felelős osztály, akiknek céljuk, hogy megakadályozzák a szerelőket, hogy vizet szállítsanak a forrástól a ciszternák fele.

- **Ősosztályok**

Player.

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **+void makeSlippery(Pipe p):** A szabotőr rövid ideig csúszóssá teszi a paraméterként kapott csövet. Ezután, ha a cső állapota csúszós lesz, és ha valamely játékos rálép, akkor véletlenszerűen a cső valamelyik végéhez kapcsolódó elemre kerül.

### 8.1.9 Spring

- **Felelősség**

A forrásokért felelős osztály, amely a rendszert, pontosabban a belőle kivezető csöveket látja el vízzel.

- **Ősosztályok**

Field → FieldNode.

- **Interfészek**

Tickable.

- **Attribútumok**

-

- **Metódusok**

- **+void tick():** Egy időegység elteltét jelenti. A forráshoz kapcsolt csövekbe a lehető legtöbb vizet folytatja a forrásból.

```
For each pipe in the pipes:  
    Call pipe.flow(MAX_VALUE)  
Endl loop.
```

### 8.1.10 Tickable

- **Felelősség**

Tickable interfész, amely az idő múlásáért felelős.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **-void tick():** Az idő múlását reprezentáló függvény, amelyet a pályaelemek implementálnak.



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

### 8.2.1 Pumpa meghibásodás

- **Leírás**

A víz folyása közben, egy pumpa meghibásodik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a meghibásodás után a pumpa állapota valóban megváltozik, valamint a víz pumpálása ennek megfelelően meg is szűnik.

- **Bemenet**

```
random off
add pump pump1
add pipe p1
add pipe p2
connect p1 pump1
connect p2 pump1
setpump pump1 p1 p2
setcurrentvolume p1 50
tick pump1
state
setcurrentvolume p1 50
breakpump pump1
tick pump1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 50  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current  
 volume: 0 standing players: null

Pipe p1 with ends: pump1, null max volume: 500 current volume: 50 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 50  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: true max volume: 500 current  
 volume: 0 standing players: null

### 8.2.2 Lépés üres csőre

- **Leírás**

Egy játékos megpróbálkozik egy olyan csőre lépni, amelyen nem áll más játékos.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy egy játékos valóban sikeresen rá tud lépni egy üres csőre, valamint a lépést követően a játékos pozíciója és a csövön álló játékosok listája módosult.

- **Bemenet**

```
add pipe pipe1
```

```
add mechanic mechanic1
move mechanic1 pipe1
state
```

- **Elvárt kimenet**

Pipe pipe1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: mechanic1  
 Mechanic mechanic1 on pipe1 has pipe: false has pump: false

### 8.2.3 Lépés foglalt csőre

- **Leírás**

Egy játékos megpróbálkozik egy olyan csőre lépni, amelyen már áll egy másik játékos, tehát a lépés oda nem lehetséges.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy egy csőre egy időben valóban csak egy játékos léphet rá, amennyiben valaki már áll a csövön és egy új játékos próbál oda lépni, akkor a lépés nem megy végbe, a játékos pozíciója nem módosul. Amennyiben a forráson álló szerelő átkerül a pumpára, vagy a forráson és csövön álló játékosok száma módosul, a futás egyértelműen hibás.

- **Bemenet**

```
add mechanic m1
add mechanic m2
add pipe p1
add spring s1
connect p1 s1
move m1 p1
move m2 s1
move m2 p1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: s1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: m1  
 Mechanic m1 on p1 has pipe: false has pump: false  
 Mechanic m2 on s1 has pipe: false has pump: false  
 Spring s1 with ends: p1 standing players: m2

### 8.2.4 Lépés üres pumpára

- **Leírás**

Egy játékos megpróbálkozik egy olyan pumpára lépni, amelyen még nem áll egy másik játékos sem.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A lépést követően a játékos pozíciója, valamint a pumpán álló játékosok listája helyes működés esetén módosul.

- **Bemenet**

```
add pump p1
add saboteur s1
```

move s1 p1  
state

- **Elvárt kimenet**

Pump p1 with ends: null input: null output: null broken: false max volume: 500 current volume: 0 standing players: s1  
Saboteur s1 on p1

### 8.2.5 Lépés foglalt pumpára

- **Leírás**

Egy játékos megpróbálkozik egy olyan pumpára lépni, amelyen már áll egy másik játékos is. Pumpa esetén ez megengedett, tehát a lépésnek sikeresnek kell lennie.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy pumpa esetén valóban lehetséges az, hogy egy elemre két játékos is rálépjen. Amennyiben a pumpán álló játékosok listájában megjelenik mindkét játékos, és az ők pozíciójuk is ezzel megfelelő, akkor a működés helyes.

- **Bemenet**

add mechanic m1  
add saboteur s1  
add pump p1  
add pipe mpos  
add pipe spos  
connect mpos p1  
connect spos p1  
move m1 mpos  
move s1 spos  
move m1 p1  
state m1  
state p1  
move s1 p1  
state

- **Elvárt kimenet**

Mechanic m1 on p1 has pipe: false has pump: false

Pump p1 with ends: mpos, spos input: null output: null broken: false max volume: 500 current volume: 0 standing players: m1

Pump p1 with ends: mpos, spos input: null output: null broken: false max volume: 500 current volume: 0 standing players: m1, s1

Mechanic m1 on p1 has pipe: false has pump: false

Pipe mpos with ends: p1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pipe spos with ends: p1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Saboteur s1 on p1

### 8.2.6 Lépés ciszternára

- **Leírás**

Egy játékos megpróbálkozik egy ciszternához kötött csőről a ciszternára lépni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ezzel a tesztesettel azt ellenőrizzük, hogy a ciszternára lépés művelete helyesen van végezve. A teszteset lefutása után a játékos aktuális pozíciója a ciszterna kell, hogy legyen, illetve a ciszternán álló játékosok listájában szerepelnie kell az adott játékosnak.

- **Bemenet**

```
add cistern cistern
add pipe pipe
connect pipe cistern
add mechanic m1
move m1 pipe
move m1 cistern
state
```

- **Elvárt kimenet**

```
Mechanic m1 on cistern has pipe: false has pump: false
Cistern cistern with ends: pipe pipe available: false drained water: 0 standing players: m1
Pipe pipe with ends: cistern, null max volume: 500 current volume: 0 wasted water: 0
broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null
```

### 8.2.7 Lépés csúszós csőre

- **Leírás**

Egy játékos rálép egy csúszós csőre, aminek következtében egy, a csővel szomszédos csomópontra kerül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy csúszós csőre lépéskor a játékos valóban nem a csőre, hanem a cső valamelyik szomszédos csomópontjára kerül.

- **Bemenet**

```
random off
add pipe pipe1
add pump pump1
add pump pump2
connect pipe1 pump1
connect pipe1 pump2
add mechanic m1
makeslippery pipe1
move m1 pipe1
state
```

- **Elvárt kimenet**

```
Mechanic m1 on pump1 has pipe: false has pump: false
Pipe pipe1 with ends: pump1, pump2 max volume: 500 current volume: 0 wasted water: 0
broken: false breakable: 0 slippery: 3 sticky: 0 standing player: null
```

Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 500  
 current volume: 0 standing players: m1  
 Pump pump2 with ends: pipe1 input: null output: null broken: false max volume: 500  
 current volume: 0 standing players: null

### 8.2.8 Ragadós csőről lépés

- **Leírás**

Egy játékos megpróbál lelépni egy ragadós csőről, ami nem lehetséges.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a lelépés valóban sikertelen-e, valamint a próbálkozást követően a játékos pozíciója, és a csövön álló emberek száma valóban nem módosult.

- **Bemenet**

add pipe pipe1  
 add pump pump1  
 connect pipe1 pump1  
 add mechanic m1  
 move m1 pipe1  
 makesticky pipe1  
 state  
 move m1 pump1  
 state

- **Elvárt kimenet**

Mechanic m1 on pipe1 has pipe: false has pump: false  
 Pipe pipe1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 3 standing player: m1  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 500  
 current volume: 0 standing players: null

Mechanic m1 on pipe1 has pipe: false has pump: false  
 Pipe pipe1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 3 standing player: m1  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 500  
 current volume: 0 standing players: null

### 8.2.9 Lépés forrásra

- **Leírás**

Egy játékos rálép egy forrásra, amelyen már áll játékos.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a lépést követően valóban módosul a játékos pozíciója, valamint a forráson álló játékosok listája.

- **Bemenet**

add spring s1  
 add pipe p1

```
connect p1 s1
add mechanic m1
add mechanic m2
move m1 p1
move m2 s1
state
move m1 s1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: s1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: m1  
 Mechanic m1 on p1 has pipe: false has pump: false  
 Mechanic m2 on s1 has pipe: false has pump: false  
 Spring s1 with ends: p1 standing players: m2

Pipe p1 with ends: s1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Mechanic m1 on s1 has pipe: false has pump: false  
 Mechanic m2 on s1 has pipe: false has pump: false  
 Spring s1 with ends: p1 standing players: m2, m1

## 8.2.10 Cső egyik végének lecsatlakoztatása

- **Leírás**

Egy szerelő lecsatlakoztatja egy cső egyik végét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy lehet-e olyan cső, aminek egyik vége szabad, valamint hogy a lecsatlakoztatás után valóban megszűnt-e a korábbi csatlakozás.

- **Bemenet**

```
add mechanic m1
add pump pump1
add pipe pipe1
add spring spring1
connect pipe1 pump1
connect pipe1 spring1
move m1 pump1
disconnect pipe1 pump1 m1
state
```

- **Elvárt kimenet**

Mechanic m1 on pump1 has pipe: false has pump: false  
 Spring spring1 with ends: pipe1 standing players: null  
 Pump pump1 with ends: null input: null output: null broken: false max volume: 500 current volume: 0 standing players: m1  
 Pipe pipe1 with ends: spring1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

### 8.2.11 Cső mindkét végének lecsatlakoztatása

- **Leírás**

Egy szerelő lecsatlakoztatja egy cső mindkét végét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy egy csőnek valóban lecsatlakoztatható mindkét vége, valamint a lecsatlakoztatás után valóban megszűnik-e a kapcsolat a cső és a csomópontok között.

- **Bemenet**

```
add pipe pipe1
add pump pump1
add pump pump2
add mechanic m1
connect pipe1 pump1
connect pipe1 pump2
move m1 pump1
state
disconnect pipe1 pump1 m1
disconnect pipe1 pump2 m1
state
```

- **Elvárt kimenet**

Mechanic m1 on pump1 has pipe: false has pump: false  
 Pipe pipe1 with ends: pump1, pump2 max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 500  
 current volume: 0 standing players: m1  
 Pump pump2 with ends: pipe1 input: null output: null broken: false max volume: 500  
 current volume: 0 standing players: null

Mechanic m1 on pump1 has pipe: false has pump: false  
 Pipe pipe1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: null input: null output: null broken: false max volume: 500 current  
 volume: 0 standing players: m1  
 Pump pump2 with ends: null input: null output: null broken: false max volume: 500 current  
 volume: 0 standing players: null

### 8.2.12 Szerelő átcsatlakoztat egy csövet egyik pumpáról egy másikra

- **Leírás**

Egy szerelő lecsatlakoztat egy csövet egy pumpáról, majd átcsatlakoztatja a szabad véget egy másik pumpára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy az átcsatlakoztatás után valóban lecsatlakozik-e a cső a régi pumpáról, és rácsatlakozik az újra.

- **Bemenet**

```
add pump basepump
```

```

add pump oldpump
add pump newpump
add mechanic m1
add pipe pipe
connect pipe oldpump
connect pipe basepump
move m1 basepump
state pipe
disconnect pipe oldpump m1
connect pipe basepump m1
state

```

- **Elvárt kimenet**

Pipe pipe with ends: oldpump, basepump max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pump newpump with ends: null input: null output: null broken: false max volume: 500 current volume: 0 standing players: null

Mechanic m1 on basepump has pipe: false has pump: false

Pump basepump with ends: pipe, pipe input: null output: null broken: false max volume: 500 current volume: 0 standing players: m1

Pipe pipe with ends: basepump, basepump max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pump oldpump with ends: null input: null output: null broken: false max volume: 500 current volume: 0 standing players: null

### 8.2.13 Szerelő csövet vesz fel a ciszternától sikeresen

- **Leírás**

Egy szerelő csövet vesz fel az eszköztárába a ciszternáról.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a felvételt követően a ciszterna valóban megjelenik a szerelő eszköztárában, valamint a ciszternánál eltűnik a cső.

- **Bemenet**

```

add cistern c1
add mechanic m1
move m1 c1
tick c1
pickupipe m1
state

```

- **Elvárt kimenet**

Mechanic m1 on c1 has pipe: true has pump: false

Cistern c1 with ends: null pipe available: false drained water: 0 standing players: m1

### 8.2.14 Szerelő csövet vesz fel a ciszternától sikertelenül

- **Leírás**



Egy szerelő csövet vesz fel az eszköztárába a ciszternáról, de nem kerül cső az eszköztárába, mert már van nála cső.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy ilyen esetben valóban nem kerül a szerelő eszköztárába még egy cső, valamint a jelenlegi cső marad nála, és a ciszternától sem vevődik el.

- **Bemenet**

```
load mechanicwithpipeandpump.txt
add cistern c2
tick c2
move m1 c2
pickpipe m1
state
```

- **Elvárt kimenet**

Mechanic m1 on c2 has pipe: true has pump: true  
 Cistern c1 with ends: null pipe available: false drained water: 0 standing players: null  
 Cistern c2 with ends: null pipe available: true drained water: 0 standing players: m1

### 8.2.15 Szerelő csövet próbál felvenni a forrásnál

- **Leírás**

Egy szerelő csövet próbál felvenni egy forrásnál, de nem kerül hozzá cső, mert a forrásról nem tud felvenni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a szerelő valóban nem tud más pályaelemen csövet fel venni, csak a ciszternánál. Amennyiben a felvételi próbálkozás után a szerelőnél megjelenik egy cső, a működés hibás.

- **Bemenet**

```
add spring spring
add mechanic m1
move m1 spring
pickpipe m1
state
```

- **Elvárt kimenet**

Spring spring with ends: null standing players: m1  
 Mechanic m1 on spring has pipe: false has pump: false

### 8.2.16 Szerelő pumpát vesz fel a ciszternától sikeresen

- **Leírás**

Egy szerelő pumpát vesz fel az eszköztárába a ciszternáról.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy amennyiben egy szerelő felvesz egy pumpát a ciszternától, az valóban megjelenik az eszköztárában.

- **Bemenet**

add cistern cistern  
add mechanic m1  
move m1 cistern  
pickuppump m1  
state

- **Elvárt kimenet**

Mechanic m1 on cistern has pipe: false has pump: true  
Cistern cistern with ends: null pipe available: false drained water: 0 standing players: m1

### 8.2.17 Szerelő pumpát vesz fel a ciszternától sikertelenül

- **Leírás**

Egy szerelő megpróbál pumpát felvenni az eszköztárába a ciszternáról, de nem kerül pumpa az eszköztárába, mert már van nála pumpa.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Amennyiben a felvevési kísérlet után egy újabb pumpa jelenik meg a szerelő eszköztárában, vagy a ciszternától elvevődik a pumpa, a működés hibás.

- **Bemenet**

load mechanicwithpipeandpump.txt  
add cistern c2  
tick c2  
move m1 c2  
pickuppump m1  
state

- **Elvárt kimenet**

Mechanic m1 on c2 has pipe: true has pump: true  
Cistern c1 with ends: null pipe available: false drained water: 0 standing players: null  
Cistern c2 with ends: null pipe available: true drained water: 0 standing players: m1

### 8.2.18 Szerelő pumpát próbál felvenni a forrásnál

- **Leírás**

Egy szerelő egy pumpát próbál felvenni egy forrásnál, de nem kerül az eszköztárába pumpa, mert a forrásról nem tud felvenni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a szerelő valóban nem tud más pályaelemen pumpát felvenni, csak a ciszternánál. Amennyiben a felvevési próbálkozás után a szerelőnél megjelenik egy pumpa, a működés hibás.

- **Bemenet**

add spring spring  
add mechanic m1  
move m1 spring

pickuppump m1  
state

- **Elvárt kimenet**

Spring spring with ends: null standing players: m1  
Mechanic m1 on spring has pipe: false has pump: false

### 8.2.19 Szabotőr csövet lyukaszt

- **Leírás**

Egy szabotőr aki csövön áll megpróbálja kilyukasztani a csövet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Annak hatására, hogy a szabotőr megpróbálja kilyukasztani a csövet, az az elvárásunk, hogy a cső lyukadt állapotba kerüljön, azaz a cső *isBroken* tagváltozója *true* értékű legyen.

- **Bemenet**

add pipe p1  
add saboteur s1  
move s1 p1  
puncturepipe p1 s1  
state

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: true breakable: 0 slippery: 0 sticky: 0 standing player: s1  
Saboteur s1 on p1

### 8.2.20 Szabotőr már megfoltozott csövet lyukaszt

- **Leírás**

A szabotőr megpróbálja kilyukasztani azt a csövet, amelyen áll, és ez a cső egy már megfoltozott cső, amelynek a foltozás után még nem járt le az ideje, hogy újból ki lehessen lyukasztani.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A tesztelésről azt várjuk el, hogy annak ellenére, hogy a szabotőr megpróbálja kilyukasztani a megfoltozott csövet, a művelet ne legyen sikeres, mivel a csövet még nem lehet kilyukasztani, tehát a cső *isBroken* tagváltozója *false* értékű kell, hogy legyen.

- **Bemenet**

random off  
add pipe p1  
add saboteur s1  
move s1 p1  
puncturepipe p1  
fixpipe p1  
puncturepipe p1 s1  
state

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 1 slippery: 0 sticky: 0 standing player: s1  
Saboteur s1 on p1

### 8.2.21 Szabotőr már megfoltozott de lyukasztható csövet lyukaszt

- **Leírás**

Egy szabotőr olyan csövet lyukaszt ami meg lett foltozva ezért egy ideig nem lehet lyukasztani de ez az idő már lejárt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a cső kilyukasztása után letelt határidő lejártával a cső újból sikeresen lyukasztható.

- **Bemenet**

```
random off
add pipe p1
add saboteur s1
move s1 p1
puncturepipe p1
fixpipe p1
state p1
tick p1
state p1
puncturepipe p1 s1
state p1
```

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 1 slippery: 0 sticky: 0 standing player: s1

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: s1

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: true breakable: 0 slippery: 0 sticky: 0 standing player: s1

### 8.2.22 Szerelő csövet lyukaszt

- **Leírás**

Egy szerelő aki egy csövön áll megpróbálja kilyukasztani a csövet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Annak hatására, hogy a szabotőr megpróbálja kilyukasztani a csövet, az az elvárásunk, hogy a cső lyukadt állapotba kerüljön, azaz a cső *isBroken* tagváltozója *true* értékű legyen.

- **Bemenet**

```
add pipe p1
add mechanic m1
```

```

move m1 p1
puncturepipe p1 m1
state

```

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: true breakable: 0 slippery: 0 sticky: 0 standing player: m1  
 Mechanic m1 on p1 has pipe: false has pump: false

### 8.2.23 Szerelő megjavít egy pumpát

- **Leírás**

A szerelő megjavít egy törött pumpát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a megjavított pumpa esetén a javítás sikeres, valamint a pumpálás ezt követően sikeresen megy.

- **Bemenet**

```

random off
add pipe p1
add pump pump1
connect p1 pump1
setcurrentvolume p1 50
add pipe p2
connect p2 pump1
setpump pump1 p1 p2
add mechanic m1
move m1 pump1
breakpump pump1
state pump1
fixpump pump1 m1
tick pump1
state

```

- **Elvárt kimenet**

Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: true max volume: 500 current volume: 0 standing players: m1

Pipe p1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 50 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Mechanic m1 on pump1 has pipe: false has pump: false

Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current volume: 0 standing players: m1

### 8.2.24 Szerelő megfoltoz egy csövet

- **Leírás**

Egy szerelő megfoltozza a csövet amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a megfoltozást követően a lyukas cső állapota megváltozik, azaz lyukasból foltozott lesz.

- **Bemenet**

```
random off
add pipe p1
add mechanic m1
puncturepipe p1
move m1 p1
state p1
fixpipe p1 m1
state p1
```

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: true breakable: 0 slippery: 0 sticky: 0 standing player: m1

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 1 slippery: 0 sticky: 0 standing player: m1

### 8.2.25 Szerelő ragadóssá tesz egy csövet

- **Leírás**

A szerelő ragadóssá teszi a csövet, amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Vizsgáljuk, hogy a művelet után a cső állapota tényleg megváltozik, tehát nem ragadósból, ragadós lesz.

- **Bemenet**

```
add pipe pipe1
add mechanic m1
move m1 pipe1
state pipe1
makesticky pipe1 m1
state pipe1
```

- **Elvárt kimenet**

Pipe pipe1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: m1

Pipe pipe1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 3 standing player: m1

**8.2.26 Szabotőr csúszóssá tesz egy csövet**

- **Leírás**

A szabotőr csúszóssá teszi a csövet, amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Vizsgáljuk, hogy a művelet után a cső állapota tényleg megváltozik, tehát nem csúszósból, csúszós lesz.

- **Bemenet**

```
add pipe pipe1
add saboteur s1
move s1 pipe1
state pipe1
makeslippery pipe1 s1
state pipe1
```

- **Elvárt kimenet**

Pipe pipe1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0  
broken: false breakable: 0 slippery: 0 sticky: 0 standing player: s1

Pipe pipe1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0  
broken: false breakable: 0 slippery: 3 sticky: 0 standing player: s1

**8.2.27 Vizet pumpálunk a ciszternába**

- **Leírás**

A ciszternába befolyik a víz egy rákötött csövön keresztül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a ciszternába való pumpálás következtében valóban a megfelelő mennyiségű víz jelenik meg a ciszterna tartályában.

- **Bemenet**

```
add cistern c1
add pipe p1
connect p1 c1
setcurrentvolume p1 50
state
tick c1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: c1, null max volume: 500 current volume: 50 wasted water: 0 broken:  
false breakable: 0 slippery: 0 sticky: 0 standing player: null

Cistern c1 with ends: p1 pipe available: false drained water: 0 standing players: null

Pipe p1 with ends: c1, null max volume: 500 current volume: 0 wasted water: 0 broken:  
false breakable: 0 slippery: 0 sticky: 0 standing player: null

Cistern c1 with ends: p1 pipe available: true drained water: 50 standing players: null

### 8.2.28 Forrásból víz folyik

- **Leírás**

A forrásból víz kerül a hozzá kötött csövekbe.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük hogy tick hatására víz folyik-e a forráshoz kötött csövekbe, valamint a víz mennyisége az elvárásoknak megfelelő-e.

- **Bemenet**

```
add spring s1
add pipe p1
add pipe p2
connect p1 s1
connect p2 s1
state
tick s1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: s1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: s1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Spring s1 with ends: p1, p2 standing players: null

Pipe p1 with ends: s1, null max volume: 500 current volume: 0 wasted water: 500 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: s1, null max volume: 500 current volume: 0 wasted water: 500 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Spring s1 with ends: p1, p2 standing players: null

### 8.2.29 Forráshoz kötött cső eldugult

- **Leírás**

A forráshoz közvetlenül kötött cső tele van, tehát a forrásból a víz nem tud kifolyni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy az eldugult csőbe valóban nem tud víz folyni a forrásból. Amennyiben a forrásból mégis folyik ki víz a csőbe, és az átlépi a kapacitását, a működés nem elfogadható.

- **Bemenet**

```
add spring s1
add pipe pipe1
add pump pump1
connect pipe1 s1
connect pipe1 pump1
setmaxvolume pipe1 10
setmaxvolume pump1 10
setcurrentvolume pipe1 10
```



```
setcurrentvolume pump1 10
state
tick s1
state
```

- **Elvárt kimenet**

Pipe pipe1 with ends: s1, pump1 max volume: 10 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 10  
 current volume: 10 standing players: null  
 Spring s1 with ends: pipe1 standing players: null

Pipe pipe1 with ends: s1, pump1 max volume: 10 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 10  
 current volume: 10 standing players: null  
 Spring s1 with ends: pipe1 standing players: null

### 8.2.30 Játékos megváltoztatja a pumpa folyási irányát

- **Leírás**

A játékos megváltoztatja, hogy a pumpa melyik csőből melyik másikba pumpálja a vizet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a megváltoztatás után a pumpa bemeneti és kimeneti csöve valóban megváltozott-e.

- **Bemenet**

```
add pump pump1
add pipe p1
add pipe p2
add pipe p3
add pipe p4
connect p1 pump1
connect p2 pump1
connect p3 pump1
connect p4 pump1
add mechanic m1
move m1 pump1
setpump pump1 p1 p2 m1
state pump1
setpump pump1 p3 p4 m1
state pump1
```

- **Elvárt kimenet**

Pump pump1 with ends: p1, p2, p3, p4 input: p1 output: p2 broken: false max volume: 500  
 current volume: 0 standing players: m1

Pump pump1 with ends: p1, p2, p3, p4 input: p3 output: p4 broken: false max volume: 500  
 current volume: 0 standing players: m1

### 8.2.31 Pumpa tickelődik

- **Leírás**

A pumpa továbbítja a bemeneti csővéről érkezett vizet a kimeneti csőbe.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a pumpálás helyesen működik-e, tehát a pumpa valóban a bemeneti csővéből szívja ki a vizet, majd valóban a kimeneti csővébe továbbítja a csövet.

- **Bemenet**

```
add pump pump1
add pipe p1
add pipe p2
connect p1 pump1
connect p2 pump1
setpump pump1 p1 p2
random off
setcurrentvolume p1 10
state
tick pump1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: pump1, null max volume: 500 current volume: 10 wasted water: 0

broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0

broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current volume: 0 standing players: null

Pipe p1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0

broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 10

broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current volume: 0 standing players: null

### 8.2.32 Csúszós cső tickelődik

- **Leírás**

Adott idő elteltével a csúszós cső visszaáll az eredeti állapotába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy adott időmennyiség elteltével a csúszós cső visszaáll-e alap állapotába.

- **Bemenet**

```
add pipe p1
makeslippery p1
state
tick p1
tick p1
tick p1
```

state

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 3 sticky: 0 standing player: null

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

### 8.2.33 Ragadós cső tickelődik

- **Leírás**

Adott idő elteltével a ragadós cső visszaáll az eredeti állapotába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy adott időmennyiség elteltével a ragadós cső visszaáll-e alap állapotába.

- **Bemenet**

add pipe p1  
makesticky p1  
state  
tick p1  
tick p1  
tick p1  
state

- **Elvárt kimenet**

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 3 standing player: null

Pipe p1 with ends: null, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

### 8.2.34 Ciszterna tickelődik

- **Leírás**

Egy ciszterna beszívja a hozzá kötött csövekben lévő vizet. Emellett a tick hatására újra termelődik a cső, ha azt előzőleg egy szerelő elvette onnan.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a ciszterna a tick hatására valóban kiszívja a hozzá csatlakoztatott csövekből a vizet. Emellett ha a ciszternánál nincsen elvehető cső, akkor az is újra termelődik.

- **Bemenet**

add cistern c1  
add pipe p1  
add pipe p2  
add pipe p3  
connect p1 c1  
connect p2 c1  
connect p3 c1

```

setcurrentvolume p1 5
setcurrentvolume p2 10
setcurrentvolume p3 0
state
tick c1
state

```

- **Elvárt kimenet**

Pipe p1 with ends: c1, null max volume: 500 current volume: 5 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: c1, null max volume: 500 current volume: 10 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p3 with ends: c1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Cistern c1 with ends: p1, p2, p3 pipe available: false drained water: 0 standing players: null

Pipe p1 with ends: c1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: c1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p3 with ends: c1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Cistern c1 with ends: p1, p2, p3 pipe available: true drained water: 15 standing players: null

### 8.2.35 A szerelő lehelyez egy pumpát egy csőre

- **Leírás**

A szerelő lehelyez egy pumpát, úgy, hogy egy csövet kettévág, és a cső végeire a pumpát csatlakoztatja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a lehelyezést követően a szerelő helyesen vágja ketté a csövet, majd csatlakoztatja a pumpára az új, és a régi cső szabad végeit.

- **Bemenet**

```

load mechanicwithpipeandpump.txt
add pipe pipe1
add pump pump1
add pump pump2
connect pipe1 pump1
connect pipe1 pump2
move m1 pipe1
placepump m1 newPump
state

```

- **Elvárt kimenet**

Pump newPump with ends: pipe1, newPipe input: null output: null broken: false max volume: 500 current volume: 0 standing players: null  
 Mechanic m1 on pipe1 has pipe: true has pump: false

Pipe newPipe with ends: pump2, newPump max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe pipe1 with ends: pump1, newPump max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: m1  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 500 current volume: 0 standing players: null  
 Pump pump2 with ends: newPipe input: null output: null broken: false max volume: 500 current volume: 0 standing players: null  
 Cistern c1 with ends: null pipe available: false drained water: 0 standing players: null

### 8.2.36 A szerelő lehelyez egy csövet

- **Leírás**

A szerelő lehelyez egy csövet, amelynek egyik végét ráköti a pumpára amelyiken áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a cső lehelyezését követően a pumpára sikerült felcsatlakoztatni-e a cső egyik végét. Emellett a szerelő eszköztárából a cső természetesen el kell tűnjön.

- **Bemenet**

```
load mechanicwithpipeandpump.txt
add pump pump1
move m1 pump1
placepipe m1 pump1 pipe1
state
```

- **Elvárt kimenet**

Mechanic m1 on pump1 has pipe: false has pump: true  
 Pump pump1 with ends: pipe1 input: null output: null broken: false max volume: 500 current volume: 0 standing players: m1  
 Pipe pipe1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Cistern c1 with ends: null pipe available: false drained water: 0 standing players: null

### 8.2.37 Telített pumpába vizet pumpálunk

- **Leírás**

Egy olyan pumpába próbálunk vizet pumpálni, amelyik megtelt, és nem képes tovább adni a vizet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a tick hatására a pumpa valóban nem képes kiszívni a bemeneti csőből a vizet, ami ennek hatására ott marad.

- **Bemenet**

```
random off
add pump pump1
add pump pump2
add pipe p1
add pipe p2
```

```

connect p1 pump1
connect p2 pump1
connect p2 pump2
setpump pump1 p1 p2
setmaxvolume pump1 10
setmaxvolume p2 10
setcurrentvolume pump1 10
setcurrentvolume p2 10
setcurrentvolume p1 10
state
tick pump1
state

```

- **Elvárt kimenet**

Pipe p1 with ends: pump1, null max volume: 500 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pipe p2 with ends: pump1, pump2 max volume: 10 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 10 current  
 volume: 10 standing players: null

Pump pump2 with ends: p2 input: null output: null broken: false max volume: 500 current  
 volume: 0 standing players: null

Pipe p1 with ends: pump1, null max volume: 500 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pipe p2 with ends: pump1, pump2 max volume: 10 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 10 current  
 volume: 10 standing players: null

Pump pump2 with ends: p2 input: null output: null broken: false max volume: 500 current  
 volume: 0 standing players: null

### 8.2.38 Üres csőből próbál vizet kiszívni a pumpa

- **Leírás**

A pumpa bemeneti csöve üres, a pumpa ilyenkor a tartályából továbbítja a vizet, amennyiben ott található víz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy ha a bemeneti cső üres, akkor a pumpa nem tud kiszívni vizet onnan, ilyenkor pedig a tartályából továbbítja a vizet a kimeneti csőbe.

- **Bemenet**

```

random off
add pump pump1
add pipe p1
add pipe p2
connect p1 pump1
connect p2 pump1
setpump pump1 p1 p2
setcurrentvolume pump1 10

```

```
state
tick pump1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current  
 volume: 10 standing players: null

Pipe p1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 10  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current  
 volume: 0 standing players: null

### 8.2.39 Szabad végű csőből víz folyik a sivatagba

- **Leírás**

A pumpa kimeneti csővének másik végén keresztül a víz a sivatagba folyik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a szabad végű csővön keresztül a víz valóban a sivatagba folyik (azaz a cső wasted water értéke nő a víz mennyiségével).

- **Bemenet**

```
add pump pump1
add pipe p1
add pipe p2
connect p1 pump1
connect p2 pump1
setpump pump1 p1 p2
setcurrentvolume p1 10
state
tick pump1
state p2
```

- **Elvárt kimenet**

Pipe p1 with ends: pump1, null max volume: 500 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current  
 volume: 0 standing players: null

Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 10  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null

### 8.2.40 Lyukas csőből víz folyik a sivatagba

- **Leírás**

A pumpa kimeneti csőve lyukas, tehát az ott folyó víz a sivatagba folyik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a lyukas csőbe pumpált vízmennyiség valóban a sivatagba jut.

- **Bemenet**

```
random off
add pump pump1
add pipe p1
add pipe p2
connect p1 pump1
connect p2 pump1
setpump pump1 p1 p2
setcurrentvolume p1 10
puncturepipe p2
state
tick pump1
state
```

- **Elvárt kimenet**

Pipe p1 with ends: pump1, null max volume: 500 current volume: 10 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: true breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current  
 volume: 0 standing players: null

Pipe p1 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 0  
 broken: false breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pipe p2 with ends: pump1, null max volume: 500 current volume: 0 wasted water: 10  
 broken: true breakable: 0 slippery: 0 sticky: 0 standing player: null  
 Pump pump1 with ends: p1, p2 input: p1 output: p2 broken: false max volume: 500 current  
 volume: 0 standing players: null

### 8.3 A tesztelést támogató programok tervei

A teszteléshez használt segédprogram által előre meghatározott tesztekkel ellenőrizhetjük a program egyes funkcióinak működését.

#### Main osztály:

- A Main osztály végzi a program vezérlését.
- Először ellenőrzi, hogy a helyes bemeneti paraméterekkel van-e meghívva, és beolvassa a tesztesetek és kimeneti fájlok elérési útjait.
- Ezután ellenőrzi, hogy a megadott elérési utak érvényesek-e, és ha nem, akkor hibát dob.
- A tesztesetek fájljait beolvassa és elválasztja őket az input és az expected fájlokra, majd egy HashMap-be tárolja őket.
- Ellenőrzi, hogy minden tesztesethez van-e input és expected fájl.



- Végül végigiterál a teszteseteken, elindítja a Proto programot, és ellenőrzi a tesztesetek eredményeit az expected kimeneti fájlal összehasonlítva.

**TestCase osztály:**

- A TestCase osztály tárolja az adott tesztesethez tartozó input és expected fájlokat.
- A run() függvény végzi a teszteset futtatását a Proto programmal, valamint az eredmények ellenőrzését az expected fájlal.
- A run() függvény először elküldi a teszteset inputját a Proto programnak.
- Ezután összehasonlítja a Proto program kimenetét az expected fájl tartalmával, soronként haladva.
- Ha van különbség a két fájl sorai között, akkor hibát dob, és a teszteset nem sikeres. Ellenkező esetben a teszteset sikeres.

**8.4 Napló**

Kezdet	Időtartam	Résztevők	Leírás
2023.04.26. 18:00	2 óra	Barabási Nagy Palásti Kurcsi Ganzer	<b>Értekezlet.</b> <b>Döntés:</b> Feladatokat felosztottuk: Tesztesetek: Nagy(8.2.1, 8.2.5, 8.2.11-8.2.22), Kurcsi(8.2.6 - 8.2.10), Palásti(8.2.23-8.2.30), Ganzer(8.2.36-8.2.40), Barabási(8.2.31-8.2.35). Emellett Kurcsi, Barabási és Ganzer felelősek a részletes tervek megírásáért. Palásti készíti a tesztelést támogató program terveit.
2023.04.26. 20:00	2 óra	Nagy	<b>Tevékenység:</b> Tesztesetek(8.2.1 - 8.2.5, 8.2.11 - 8.2.14)
2023.04.27. 12:00	1 óra	Palásti	<b>Tevékenység:</b> A tesztelést támogató program terveinek megírása.
2023.04.27. 14:00	3 óra	Kurcsi	<b>Tevékenység:</b> Pump, Pipe osztályok részletes terveinek megírása
2023.04.27. 15:00	3 óra	Barabási	<b>Tevékenység:</b> Player, Mechanic, Saboteur, Spring osztályok részletes terveinek megírása
2023.04.27. 15:00	3 óra	Ganzer	<b>Tevékenység:</b> Cistern, Field, FieldNode osztályok részletes terveinek megírása

2023.04.28. 09:00	2 óra	Nagy	<b>Tevékenység:</b> Tesztesetek(8.2.15 - 8.2.22)
2023.04.28 12:00	1 óra	Kurcsi	<b>Tevékenység:</b> Tesztesetek(8.2.6 - 8.2.10)
2023.04.28. 12:00	2 óra	Palásti	<b>Tevékenység:</b> Tesztesetek(8.2.23 - 8.2.30)
2023.04.28 14:00	1 óra	Barabási	<b>Tevékenység:</b> Tesztesetek(8.2.31 - 8.2.35)
2023.04.28 16:00	1 óra	Ganzer	<b>Tevékenység:</b> Tesztesetek(8.2.36 - 8.2.40)
2023.04.29 13:00	3 óra	Barabási Nagy Palásti Kurcsi Ganzer	<b>Értekezlet:</b> Elvégzett feladatok egyeztetése és ellenőrzése. <b>Döntés:</b> Palásti végzi a dokumentum véglegesítését.
2023.04.29 18:00	1 óra	Palásti	<b>Tevékenysége:</b> A dokumentum ellenőrzése, hibák keresése és javítása, dokumentum véglegesítése.