# Részletes tervek

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

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

## Osztályok és metódusok tervei.

### Osztály1

#### Felelősség

[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]

#### Ősosztályok

[Mely osztályokból származik (öröklési hierarchia)]

Legősebb osztály → Ősosztály2 → Ősosztály3...

#### Interfészek

[Mely interfészeket valósítja meg.]

#### Attribútumok

[Milyen attribútumai vannak, beleértve a nem publikusakat is.]

* **attribútum1**: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
* **attribútum2**: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa

#### Metódusok

[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról. Minden olyan metódusnak szerepelnie kell, amelyiket az osztály megvalósít vagy felüldefiniál.]

* **int foo(Osztály3 o1, Osztály4 o2)**: metódus leírása, láthatósága (UML jelöléssel)
* **int bar(Osztály5 o1)**: metódus leírása, láthatósága (UML jelöléssel)

### Osztály2

#### Felelősség

[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]

#### Ősosztályok

[Mely osztályokból származik (öröklési hierarchia)]

Legősebb osztály → Ősosztály2 → Ősosztály3...

#### Interfészek

[Mely interfészeket valósítja meg.]

#### Attribútumok

[Milyen attribútumai vannak, beleértve a nem publikusakat is.]

* **attribútum1**: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
* **attribútum2**: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa

#### Metódusok

[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról. Minden olyan metódusnak szerepelnie kell, amelyiket az osztály megvalósít vagy felüldefiniál.]

* **int foo(Osztály3 o1, Osztály4 o2)**: metódus leírása, láthatósága (UML jelöléssel)
* **int bar(Osztály5 o1)**: metódus leírása, láthatósága (UML jelöléssel)

## A tesztek részletes tervei, leírásuk a teszt nyelvén

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

Kiadható parancsok:

**CREATE\_INSECT tecton\_name insect\_name**

**Leírás: A rovar létrejön és rákerül az argumentumként megadott céltektonra, ha ezen van gombafonál**

**Opciók: Argumentumok: A céltekton, ahova létrejönne, az objektum neve**

**MOVE insect\_name tecton\_name**

**Leírás: A rovar átmenne az argumentumként megadott céltektonra**

**Opciók: Argumentumok: A rovar, amelyik mozogna;**

**A céltekton, ahova mozogna**

**EAT insect\_name**

**Leírás: A rovar megeszik egy spórát a tektonján**

**Opciók: Argumentumok: A rovar, amelyik enne**

**CUT insect\_name**

**Leírás: A rovar elvág egy fonalat a tektonján**

**Opciók: Argumentumok: A rovar, amelyik vágna**

### Rovar létrehozása és letevése

* **Leírás**

Létrehozunk egy rovarat egy tektonra. Beállitja a létrehozott rovar tektonját arra amelyikre létrehozták, és ennek az occupants listájára hozzáadja az újonnan létrehozott rovart.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Új rovar létrehozása, tektonok beállitása*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

// Act

CREATE\_INSECT ft1 i1

// Assert

STATE ft1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft11

i1: Insect

location = ft11

maxMoves = 2

remainingMoves = 2

sporesEaten = 0

effectTimer = 0

state = Normal

### Rovar mozgatása

* **Leírás**

A rovar mozog egy, a tektonjával szomszédos, tektonra

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megváltozott-e, és tükrözi-e ezt a mozgást a két tekton állapota*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

CREATE\_TECTON FertileTecton ft2

ADD\_NEIGHBOUR ft1 ft2

SET\_BREAKTIMER ft1 5

SET\_BREAKTIMER ft2 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

CREATE\_MYCELIUM Mycelium my2

ADD\_MYCELIUM\_TO\_TECTON my2 ft2

CREATE\_INSECT ft1 i1

// Act

MOVE i1 ft2

// Assert

STATE ft1

STATE ft2

STATE my1

STATE my2

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my2

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

my2: Mycelium

location = ft2

i1: Insect

location = ft2

maxMoves = 2

remainingMoves = 1

sporesEaten = 0

effectTimer = 0

state = Normal

### Rovar sikertelen mozgatása nem-szomszédos tektonra

* **Leírás**

A rovar mozogna egy, a tektonjával nem szomszédos, tektonra, ami nem sikerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megmaradt-e a régi, és tükrözi-e ezt a mozgást a két tekton állapota*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

CREATE\_TECTON FertileTecton ft2

SET\_BREAKTIMER ft1 5

SET\_BREAKTIMER ft2 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

CREATE\_MYCELIUM Mycelium my2

ADD\_MYCELIUM\_TO\_TECTON my2 ft2

CREATE\_INSECT ft1 i1

// Act

MOVE i1 ft2

// Assert

STATE ft1

STATE ft2

STATE my1

STATE my2

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my2

}

occupants List<Insect> = {

}

my1: Mycelium

location = ft1

my2: Mycelium

location = ft2

i1: Insect

location = ft1

maxMoves = 2

remainingMoves = 2

sporesEaten = 0

effectTimer = 0

state = Normal

### Rovar sikertelen mozgatása olyan tektonra, amelyen nincs gombafonál

* **Leírás**

A rovar mozog egy, a tektonjával szomszédos, tektonra, de nem sikerül, mert nincs rajta gombafonál

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megváltozott-e, és tükrözi-e ezt a mozgást a két tekton állapota*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

CREATE\_TECTON FertileTecton ft22

ADD\_NEIGHBOUR ft1 ft2

SET\_BREAKTIMER ft1 5

SET\_BREAKTIMER ft2 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

CREATE\_INSECT ft1 i1

// Act

MOVE i1 ft2

// Assert

STATE ft1

STATE ft2

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft2

}

myceliumCapacity int = 1

spores Queue<Spore> = {}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 2

remainingMoves = 1

sporesEaten = 0

effectTimer = 0

state = Normal

### Rovar általi spóraevés következtében kettészakadás

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására kettészakad

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

PUT\_SPORE SplitSpore SplitS1 ft1

CREATE\_INSECT ft1 i1

// Act

EAT i1

// Assert

STATE ft1

STATE my1

STATE i1

STATE i2

//i2-vel van jelölve az újonnan létrejött rovar

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

i2

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 2

remainingMoves = 1

sporesEaten = 1

effectTimer = 0

state = Normal

i2: Insect

location = ft1

maxMoves = 2

remainingMoves = 0

sporesEaten = 0

effectTimer = 0

state = Normal

### Rovar általi spóraevés következtében Slow állapotba kerülés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására Slow állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

PUT\_SPORE SlownessSpore SlowSp1 ft1

CREATE\_INSECT ft1 i1

// Act

EAT i1

// Assert

STATE ft1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 1

remainingMoves = 0

sporesEaten = 1

effectTimer = 3

state = Slow

### Rovar általi spóraevés következtében Fast állapotba kerülés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására Fast állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

PUT\_SPORE SpeedSpore SpeedSp1 ft1

CREATE\_INSECT ft1 i1

// Act

EAT i1

// Assert

STATE ft1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 3

remainingMoves = 0

sporesEaten = 1

effectTimer = 3

state = Fast

### Rovar általi spóraevés következtében PreventCut állapotba kerülés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására PreventCut állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

PUT\_SPORE PreventCutSpore PreventCutSp1 ft1

CREATE\_INSECT ft1 i1

// Act

EAT i1

// Assert

STATE ft1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 2

remainingMoves = 0

sporesEaten = 1

effectTimer = 3

state = CannotCut

### Rovar általi spóraevés következtében Stunned állapotba kerülés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására PreventCut állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

PUT\_SPORE StunSpore StunSp1 ft1

CREATE\_INSECT ft1 i1

// Act

EAT i1

// Assert

STATE ft1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 0

remainingMoves = 0

sporesEaten = 1

effectTimer = 1

state = Stun

### Rovar általi sikertelen spóraevés

* **Leírás**

A rovar megeszik egy spórát a tektonján, aminek a hatására PreventCut állapotba kerül

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Spóraevést követően a tektonról eltűnik-e a spóra, illetve ennek hatása beállitódik-e a rovaron*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

SET\_BREAKTIMER ft1 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

CREATE\_INSECT ft1 i1

// Act

EAT i1

// Assert

STATE ft1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

i1

}

my1: Mycelium

location = ft1

i1: Insect

location = ft1

maxMoves = 2

remainingMoves = 2

sporesEaten = 0

effectTimer = 0

state = Normal

### Rovar általi gombafonál elvágás

* **Leírás**

A rovar mozog egy, a tektonjával szomszédos, tektonra

* **Ellenőrzött funkcionalitás, várható hibahelyek**

*Rovar tektonja megváltozott-e, és tükrözi-e ezt a mozgást a két tekton állapota*

* **Bemenet**

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

// Arrange

CREATE\_TECTON FertileTecton ft1

CREATE\_TECTON FertileTecton ft2

ADD\_NEIGHBOUR ft1 ft2

SET\_BREAKTIMER ft1 5

SET\_BREAKTIMER ft2 5

CREATE\_MYCELIUM Mycelium my1

ADD\_MYCELIUM\_TO\_TECTON my1 ft1

CREATE\_MUSHROOM\_BODY ft2 mb1

CREATE\_INSECT ft1 i1

// Act

CUT i1

// Assert

STATE ft1

STATE ft2

STATE mb1

STATE my1

STATE i1

* **Elvárt kimenet**

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

ft1: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {ft2}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = null

mycelia Queue<Mycelium> = {

my1

}

occupants List<Insect> = {

}

ft2: FertileTecton

breakTimer int = 5

neighbours List<Tecton> = {

ft1

}

myceliumCapacity int = 1

spores Queue<Spore> = {

}

mushroomBody MushroomBody = mb1

mycelia Queue<Mycelium> = {

}

occupants List<Insect> = {

i1

}

mb1: MushroomBody

location = ft2

remainingEjects = 3

my1: Mycelium

location = ft1

deathTimer = 2

i1: Insect

location = ft2

maxMoves = 2

remainingMoves = 0

sporesEaten = 0

effectTimer = 0

state = Normal

## A tesztelést támogató programok tervei

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

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2025.04.06 16:00 | 2 óra | Bencze | Tevékenység:  Parancsok és tesztesetek átemelése, illetve tesztesetek be- és kimenetének leírása |
|  |  |  |  |
| … | … | … | … |