

# Tóth Zsolt

(CZ72YM)

## Feladat leírás:

Egy bolygón különböző fajtájú növények élnek, minden növény tápanyagot használ. Ha egy növény tápanyaga elfogy (a mennyiség 0 lesz), a növény elpusztul. A bolygón három fajta sugárzást különböztetünk meg: alfa sugárzás, delta sugárzás, nincs sugárzás. A sugárzásra a különböző fajtájú növények eltérő módon reagálnak. A reakció tartalmazza a tápanyag változását, illetve a következő napi sugárzás befolyásolását. A másnapi sugárzás alakulása: ha az alfa sugárzásra beérkezett igények összege legalább hárommal meghaladja a delta sugárzás igényeinek összegét, akkor alfa sugárzás lesz; ha a delta sugárzásra igaz ugyanez, akkor delta sugárzás lesz; ha a két igény közti eltérés háromnál kisebb, akkor nincs sugárzás. Az első nap sugárzás nélküli.

***Szimuláljuk a növények viselkedését és minden lépésben írjuk ki az összes növényt a rájuk jellemző tulajdonságokkal, valamint az aktuális sugárzást!***

Minden növény jellemzői: az egyedi neve (String), a rendelkezésre álló tápanyag mennyisége (egész), hogy él-e (logikai). A szimulációban részt vevő növények fajtái a következők: puffancs, deltafa, parabokor. A következőkben megadjuk, hogy miként reagálnak a különböző sugárzásokra. Először a tápanyag változik, és ha a növény ezután él, akkor befolyásolhatja a sugárzást.

**Puffancs:** Alfa sugárzás hatására a tápanyag mennyisége kettővel nő, sugárzás mentes napon a tápanyag eggyel csökken, delta sugárzás esetén a tápanyag kettővel csökken. Minden esetben úgy befolyásolja a másnapi sugárzást, hogy 10-tápanyag értékben növeli az alfa sugárzás bekövetkezését. Ez a fajta akkor is elpusztul, ha a tápanyag mennyisége 10 fölé emelkedik.

**Deltafa:** Alfa sugárzás hatására a tápanyag mennyisége hárommal csökken, sugárzás nélküli napon a tápanyag eggyel csökken, delta sugárzás hatására a tápanyag négyvel nő. Ha a tápanyag mennyisége 5-nél kisebb, akkor 4 értékben növeli a delta sugárzás bekövetkezését, ha 5 és 10 közé esik, akkor 1 értékben növeli a delta sugárzás bekövetkezését, ha 10-nél több, akkor nem befolyásolja a másnapi sugárzást.

**Parabokor:** Akár alfa, akár delta sugárzás hatására a tápanyag mennyisége eggyel nő. Sugárzás nélküli napon a tápanyag eggyel csökken. A másnapi sugárzást nem befolyásolja.

A program egy szövegfájlból olvassa be a szimuláció adatait! Az első sorban a növények száma szerepel. A következő sorok tartalmazzák a növények adatait szóközzel elválasztva: a növény nevét, a fajtáját és a kezdetben rendelkezésre álló tápanyag mennyiségét. A fajtát egy karakter azonosítja: a – puffancs, d – deltafa, p – parabokor. A növényeket leíró részt követő sorban a szimuláció napjainak száma adott egész számként. A program kérje be a fájl nevét, majd jelenítse meg a túlélők nevét! (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4

Falánk a 7

Sudár d 5

Köpcös p 4

Nyúlánk d 3

10

## Fejlesztői eszközök:

- IntelliJ 2020.3
- Java 15 (preview tartalmak is)
- Gradle 6.7
- Lombok 1.18
- Junit 5.6
- Mockito 3.7

Fordítás: *gradle build*

Futtatás: *java -jar --enable-preview vegetation-simulation-1.0.0.jar*

## Megoldási terv:

### 1, *Objektum modellek elkészítése.*

A program három növény típust kezel, melyek több közös tulajdonsággal rendelkeznek (név, tápanyag, életben van-e), így egy közös őszosztály létrehozása szükséges.

*Plan(name: String, nutrient: Int, alive: Bool)*

Az őszosztály tartalmazza a közös funkcionalitást is:

- Tápanyag vizsgálata, életben van jelző beállítása
- Sugárzás hatása adott napon

Mivel az adott napi sugárzás hatás kiszámítási módjának folyamatlépései megegyeznek, az őszosztály tartalmazza ezt a funkcionalitást. Az eltéréseket absztrakt metódusok hívásával kezeli, melyeket az alosztályok implementálnak.

Ebből a szülő osztályból származnak a konkrét osztályok (*Puffancs, Deltafa, Parabokor*)

Szükséges a sugárzás típusainak kezelése. Három adott típusú sugárzással dolgozik a program, melyet egy enumerációs osztályban ír le.

*Radiation {ALPHA, DELTA, NEUTRAL}*

Az egyes növények következő napi sugárzásra gyakorolt hatását két tulajdonság ír le. Mely sugárzás esélyét növeli, illetve mennyivel.

*Effect(radiation: Radiation, quantity: int)*

A szimuláció futtatása előtt fájlból olvassuk be a szükséges adatokat. Ezen adatok tárolására szintén létre kell hozni egy osztályt, mely tartalmazza a szimulálandó napok számát, valamint a szimulációban résztvevő növényeket.

*SimulationData(days: Int, plants: [])*

### 2, *Szimuláció megvalósítása*

A szimuláció több lépésre bontható.

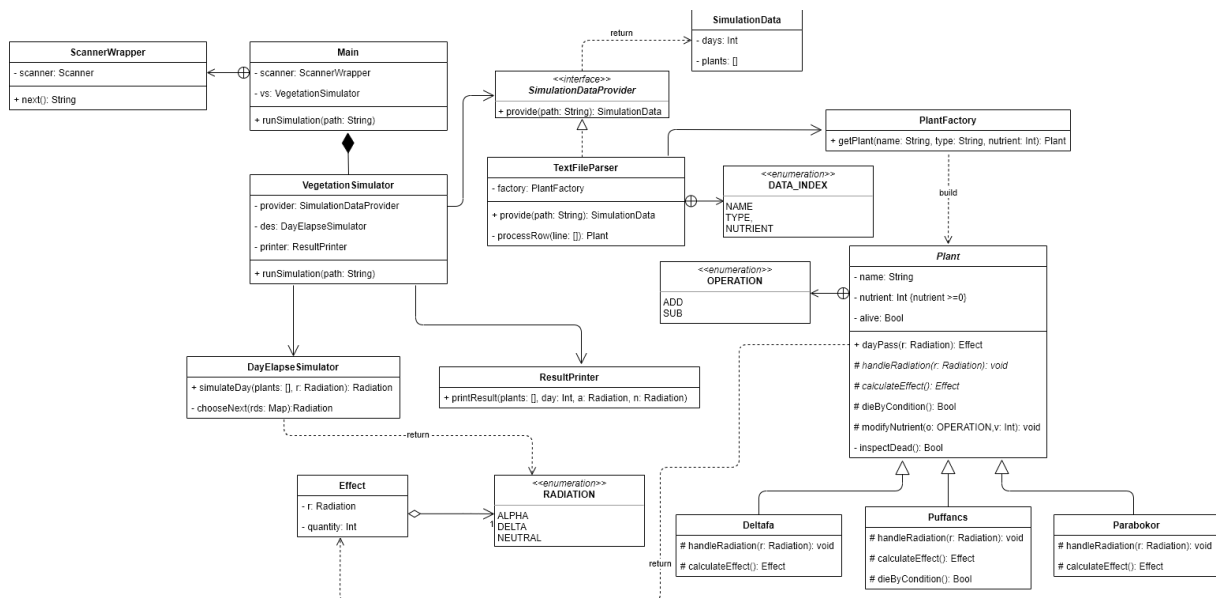
1. Interakció a felhasználóval, fájl elérési útvonalának bekérése.
2. Fájl beolvasása, növény példányok létrehozása.
3. Napszámszor végrehajtva:
  - i. Minden növényen az adott napi sugárzás hatásának végrehajtása.
  - ii. A növények sugárzásra gyakorolt hatásainak összesítése.
  - iii. Az adott nap eredményének kiírása.

A fenti lépések alapján a funkcionalitás könnyen részeegységekre bontható.

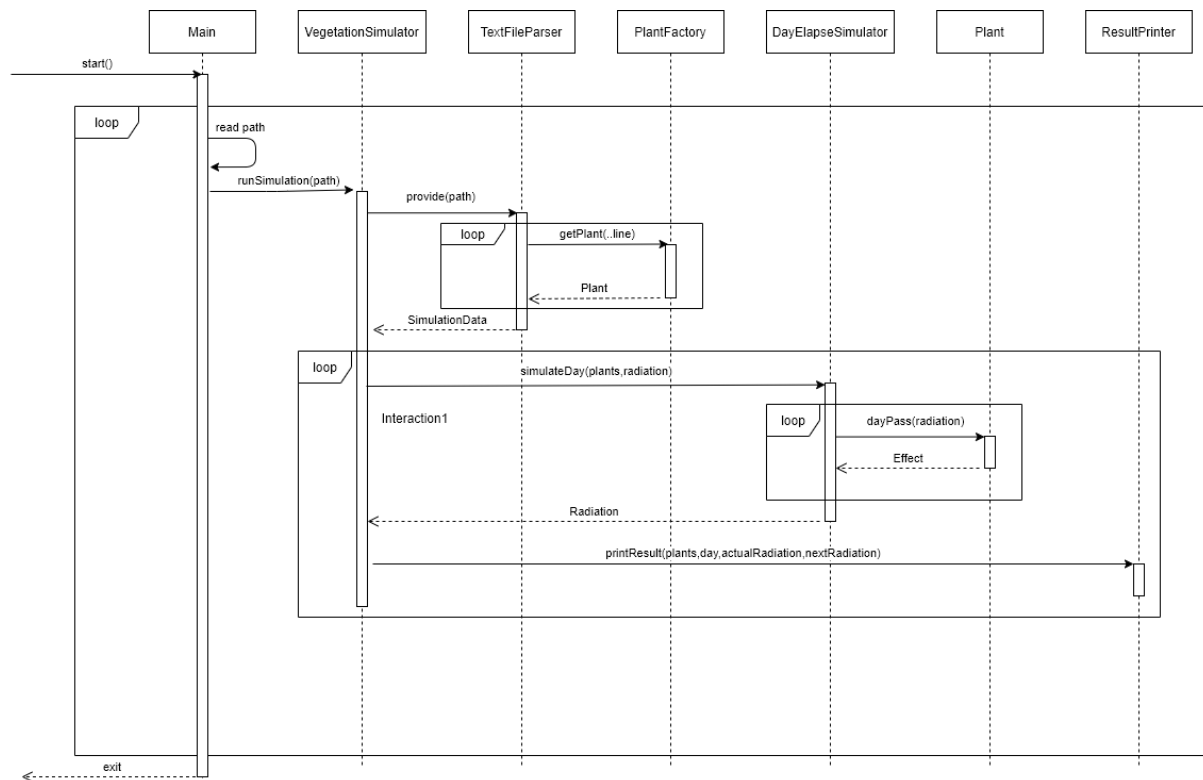
A felhasználóval történő interakció a *Main* osztályból történik. A *VegetationSimulator* vezérli a szimulációs folyamatot. A *TextFileParser* osztály az adatok beolvasásáért felelős, mely alapján a *PlantFactory* segítségével legyártja a szükséges növény (*Plant*) példányokat.

Az egyes napokon végrehajtandó számításokat a *DayElapseSimulator* osztály kezeli, mely visszaadja a következő napi sugárzást (*Radiation*). A *VegetationSimulator* a következő ciklus-iterációban (napon) ezzel az értékkel hívja a *DayElapseSimulator* kalkulációs metódusát. Az eredményeket minden ciklus-iterációban a *ResultPrinter* osztály segítségével írja a képernyőre.

## Osztály diagram:



## Szekvencia diagram:



## Tesztelési terv:

Cél az egyes osztályok fő funkcióinak és határesetének minél szélesebb körű lefedése egységteszttekkel. Valamint az IO műveletek megvizsgálása integrációs tesztekkel.

95% classes, 96% lines covered in package 'com.zlrx.progtech.vegetation.simulator'

Element	Class, %	Method, %	Line, %
helper	100% (2/2)	100% (3/3)	100% (19/19)
model	100% (12/12)	100% (34/34)	100% (74/74)
service	100% (4/4)	100% (11/11)	100% (46/46)
Main	50% (1/2)	40% (2/5)	80% (20/25)

Tesztelt metódus	Leírás	Bemenet	Elvárt eredmény
PlantFactory.dayPass (PlantFactoryTest)	Puffancs létrehozás	'a' típus	Puffancs objektumot tartalmazó Optional
	Deltafa létrehozás	'd' típus	Deltafa objektumot tartalmazó Optional
	Parabokor létrehozás	'p' típus	Parabokor objektumot tartalmazó Optional
	Üres eredmény nem ismert típus esetén	'x' típus	Optional.empty()

<b>Deltafa.dayPass (DeltafaTest)</b>	Alfa sugárzás hárommal csökkenti a tápanyag mennyiséget	deltafa 7 tápanyaggal alfa sugárzás	deltafa 4 tápanyaggal
	Delta sugárzás négyvel növeli a tápanyag mennyiségét	deltafa 1 tápanyaggal delta sugárzás	deltafa 5 tápanyaggal
	Sugárzásmentes napon eggyel csökken a tápanyag mennyisége	deltafa 12 tápanyaggal semleges sugárzás	deltafa 11 tápanyaggal
	Elpusztul ha a tápanyag nullára csökken	deltafa 2 tápanyaggal alfa sugárzás	deltafa 0 tápanyaggal életben jelző hamis
	Öt tápanyag alatt +4 egységgel növeli a delta sugárzás esélyét	deltafa 7 tápanyaggal alfa sugárzás	deltafa 4 tápanyaggal hatás 4 delta sugárzás
	Öt és tíz között +1 egységgel növeli az delta sugárzás esélyét	deltafa 13 tápanyaggal alfa sugárzás	deltafa 10 tápanyaggal hatás 1 delta sugárzás
	Tíz felett nem növeli a sugárzás esélyét	deltafa 14 tápanyaggal alfa sugárzás	deltafa 11 tápanyaggal hatás 0 sugárzással
<b>Parabokor.dayPass (ParabokorTest)</b>	Alfa sugárzás eggyel növeli a tápanyag mennyiséget	parabokor 5 tápanyaggal alfa sugárzás	parabokor 6 tápanyaggal hatás 0 sugárzás
	Delta sugárzás eggyel növeli a tápanyag mennyiségét	parabokor 5 tápanyaggal delta sugárzás	parabokor 6 tápanyaggal hatás 0 sugárzás
	Sugárzásmentes napon eggyel csökken a tápanyag mennyisége	parabokor 5 tápanyaggal semleges sugárzás	parabokor 6 tápanyaggal hatás 0 sugárzás
	Elpusztul ha a tápanyag nullára csökken	parabokor 1 tápanyaggal semleges sugárzás	parabokor 0 tápanyaggal életben jelző hamis hatás 0 sugárzás
<b>Puffancs.dayPass Plants constructor (PuffancsTest)</b>	Elpusztult puffancs nincs hatással a sugárzásra	elpusztult puffancs	hatás 0 sugárzás
	Nullánál kisebb tápanyaggal elpusztulva jön létre	-1 tápanyag konstruktorba	puffancs 0 tápanyaggal életben jelző hamis
	Alfa sugárzás kettővel növeli a tápanyag mennyiséget	puffancs 8 tápanyaggal alfa sugárzás	puffancs 10 tápanyaggal hatás 0 alfa

	Delta sugárzás kettővel csökkenti a tápanyag mennyiségét	puffancs 3 tápanyaggal delta sugárzás	puffancs 1 tápanyaggal hatás 9 alfa
	Sugárzásmentes napon eggyel csökken a tápanyag mennyisége	puffancs 5 tápanyaggal semleges sugárzás	puffancs 4 tápanyaggal hatás 6 alfa
	Elpusztul ha a tápanyag nullára csökken	puffancs 1 tápanyaggal delta sugárzás	puffancs 0 tápanyaggal életben jelző hamis hatás - sugárzás
	Elpusztul ha a tápanyag tíz fölé emelkedik	puffancs 8 tápanyaggal alfa sugárzás	puffancs 11 tápanyaggal életben jelző hamis hatás - sugárzás
<b>DayElapseSimulator</b> <b>.simulateDay</b> <b>(DayElapseSimulatorTest)</b>	Ha minden növény elpusztult, semleges sugárzást várunk	elpusztult puffancs, deltafa, parabokor	semleges sugárzás
	Ha az alfa hatások összessége legalább hárommal nagyobb a delta hatások összességénél akkor alfa sugárzás keletkezik	plants objektumok: puffancs 1 deltafák 8, 8, 13, 11 alfa sugárzás	alfa sugárzás
	Ha az delta hatások összessége legalább hárommal nagyobb a alfa hatások összességénél akkor delta sugárzás keletkezik	plants objektumok: puffancs 5 deltafák 6, 6, 5, 11 alfa sugárzás	delta sugárzás
	Ha delta összege kevesebb mint hárommal több alfa hatás összegénél, akkor semleges sugárzás keletkezik	plants objektumok: puffancs 5 deltafák 6, 8 alfa sugárzás	semleges sugárzás
	Ha alfa hatás összege kevesebb mint hárommal több delta hatás összegénél, akkor semleges sugárzás keletkezik	plants objektumok: puffancs 4 deltafák 8, 9 alfa sugárzás	semleges sugárzás
<b>TextFileParser</b> <b>.provide</b> <b>(TextFileParserIntTest)</b>	Nem létező fájl esetén nem kap adatot	hibás fájl útvonal	Optional.empty()
	Létező file esetén felolvassa a file-t és	resource mappában található fájl útvonala	Optionalba csomagol szimulációs adatok a

	létrehozza a szimulációs adatokat		fájlnak megfelelő adatokkal
<b>VegetationSimulator</b> <b>.runSimulation</b> <b>(VegetationSimulatorTest)</b>	Ha nincs adat, egyszer sem hívódik meg a kiírás és a kalkulációs metódusok	nem létező file útvonal (mock alter)	nincs interakció az elvárt metódusokkal
	Megfelelő bemenettel meghívódnak a kalkulációs metódusok és jó végeredményt adnak	Puffancs 10 Parabokor 3 Deltafák 7, 5, 5, 7	Megfelelő számú interakció Helyes végeredmény
<b>Main.start</b> <b>(MainE2ETest)</b>	A folyamat elindul, feldolgozza a fájlt és exit parancsra kilép 1 másodperces időlimiten belül	-	Isd leírás

Teszt riportok:

Test Summary

32  
tests

0  
failures

0  
ignored

0.672s  
duration

100%  
successful

Packages

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">com.zlrx.progtech.vegetation.simulator.MainE2ETest</a>	1	0	0	0.523s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.helper.PlantFactoryTest</a>	4	0	0	0.027s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.model.plants.DeltafaTest</a>	7	0	0	0.026s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.model.plants.ParabokorTest</a>	4	0	0	0.002s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.model.plants.PuffancsTest</a>	7	0	0	0.057s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.service.DayElapseSimulatorTest</a>	5	0	0	0.005s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.service.TextFileParserIntTest</a>	2	0	0	0.003s	100%
<a href="#">com.zlrx.progtech.vegetation.simulator.service.VegetationSimulatorTest</a>	2	0	0	0.029s	100%

# MainE2ETest

all > [com.zlrx.progtech.vegetation.simulator](#) > MainE2ETest

1  
tests

0  
failures

0  
ignored

0.523s  
duration

100%  
successful

Tests

Standard output

Test	Method name	Duration	Result
Run test of the system	testRun()	0.523s	passed

# PlantFactoryTest

all > [com.zlrx.progtech.vegetation.simulator.helper](#) > PlantFactoryTest

4  
tests

0  
failures

0  
ignored

0.027s  
duration

100%  
successful

Tests

Test	Method name	Duration	Result
When type is 'd' factory should return a Deltafa instance	testDeltafaFromFactory()	0.001s	passed
Factory should return empty() when type not in 'a,d,p'	testFactoryException()	0.001s	passed
When type is 'p' factory should return a Parabokor instance	testParabokorFactory()	0.024s	passed
When type is 'a' factory should return a Puffancs instance	testPuffancsFromFactory()	0.001s	passed

# DeltafaTest

all > [com.zlrx.progtech.vegetation.simulator.model.plants](#) > DeltafaTest

7  
tests

0  
failures

0  
ignored

0.026s  
duration

100%  
successful

Tests

Test	Method name	Duration	Result
Alpha radiation should decrease nutrients by 3	alphaRadiationShouldDecreaseNutrientsByThree()	0.001s	passed
Delta radiation should increase nutrients by 4	deltaRadiationShouldIncreaseNutrientsByFour()	0.002s	passed
Neutral radiation should decrease nutrients by 1	neutralRadiationShouldDecreaseNutrientsByOne()	0.001s	passed
Deltafa should die when nutrients reach 0	puffancsShouldDieWhenNutrientReachesZero()	0s	passed
Should return 4 delta when nutrients below five	shouldReturnFourDeltaWhenNutrientsBelowFive()	0.001s	passed
Should return neutral when nutrients above ten	shouldReturnNeutralWhenNutrientsAboveTen()	0.001s	passed
Should return 1 delta when nutrients between five and ten	shouldReturnOneDeltaWhenNutrientsBetweenFiveToTen()	0.020s	passed



## ParabokorTest

all > [com.zlrx.progtech.vegetation.simulator.model.plants](#) > ParabokorTest

4  
tests

0  
failures

0  
ignored

0.002s  
duration

100%  
successful

### Tests

Test	Method name	Duration	Result
Alpha radiation should increase nutrients by 1	alphaRadiationShouldIncreaseNutrientsByOne()	0s	passed
Delta radiation should increase nutrients by 1	deltaRadiationShouldDecreaseNutrientsByOne()	0.001s	passed
Neutral radiation should decrease nutrients by 1	neutralRadiationShouldDecreaseNutrientsByOne()	0.001s	passed
Parabokor should die when nutrients reach 0	parabokorShouldDieWhenNutrientsReachZero()	0s	passed

## PuffancsTest

all > [com.zlrx.progtech.vegetation.simulator.model.plants](#) > PuffancsTest

7  
tests

0  
failures

0  
ignored

0.057s  
duration

100%  
successful

### Tests

Test	Method name	Duration	Result
Alpha radiation should increase nutrients by 2	alphaRadiationShouldIncreaseNutrientsByTwo()	0.002s	passed
Dead Puffancs should return empty()	deadPuffancsShouldReturnEmpty()	0.001s	passed
Delta radiation should decrease nutrients by 2	deltaRadiationShouldDecreaseNutrientsByTwo()	0.003s	passed
Neutral radiation should decrease nutrients by 1	neutralRadiationShouldDecreaseNutrientsByOne()	0.003s	passed
Puffancs should die when nutrients reach 11	puffancsShouldDieWhenNutrientsReach()	0.046s	passed
Puffancs should die when nutrients reach 0	puffancsShouldDieWhenNutrientsReachZero()	0.001s	passed
Plant with negative nutrient should be still-born with 0 nutrient	stillBornTest()	0.001s	passed

## DayElapseSimulatorTest

all > [com.zlrx.progtech.vegetation.simulator.service](#) > DayElapseSimulatorTest

5  
tests

0  
failures

0  
ignored

0.005s  
duration

100%  
successful

### Tests

Test	Method name	Duration	Result
should return alpha when alpha is greater than delta at least by three	alphaResultTest()	0.002s	passed
should return delta when delta is greater than alpha at least by three	deltaResultTest()	0s	passed
should return neutral when alpha is greater then delta by two	neutralResultAlphaTest()	0.001s	passed
should return neutral when delta is greater then alpha by two	neutralResultDeltaTest()	0.001s	passed
should return neutral when all plants are dead	shouldReturnNeutralWhenAllPlantsAreDead()	0.001s	passed

## TextFileParserIntTest

all > [com.zlrx.progtech.vegetation.simulator.service](#) > TextFileParserIntTest

2	0	0	0.003s	100% successful
tests	failures	ignored	duration	

Tests Standard output

Test	Method name	Duration	Result
should return the the content of the file when path and content correct	testFileRead()	0.002s	passed
should return empty() when file not found on path	testInvalidFileRead()	0.001s	passed

## VegetationSimulatorTest

all > [com.zlrx.progtech.vegetation.simulator.service](#) > VegetationSimulatorTest

2	0	0	0.029s	100% successful
tests	failures	ignored	duration	

Tests Standard output

Test	Method name	Duration	Result
should simulate when file provided	dataProvidedTest()	0.028s	passed
should not call simulator and printer when no data provided	noDataProvidedTest()	0.001s	passed