

(Interface) Spawn:

The interface will have 3 method headers namely:

-sprout

-sapling

-mature

The existing class Tree will implement this interface

Sprout:

public void sprout(){}

-Instantiate a sprout object ‘+’

-With 10% chance it’ll instantiate a Goomba object via the Application class dependency

-If an actor is existent, don’t instantiate Goomba

-A counter will be set for number of times this method is called. After count reaches 10 the sapling method is called.

Sapling:

Public void sapling(){}

-Instantiate a sapling object ‘t’

-With 10% chance it instantiates a Coin object.Add this coin object to wallet

-Similar to the sprout method, after count reaches 10, the Mature method is called.

Mature:

Public void nature(){}

-Instantiate a mature object ‘T’

-With 15% chance instantiate a Koopa object

-If actor is existent, don’t instantiate Koopa

-Another counter is set and this time, after every 5 turns AND if Dirt object is available,call the sprout method.

-With 20% chance, switch capabilitySet.addCapability status to DEAD and instantiate Dirt object.

Koopa:

Has the same attributes and methods as Goomba?

REQ2-

Add jump feature in the Actor class:

Needs to have an association with Ground so it can access the sapling sprout and mature methods within Tree class which is a subclass of Ground.

-If a wall exists, make the jump

-If jump fails get the current hit points and subtract 20

-Repeat process for sprouts,saplings and matures but with their corresponding success probabilities and hit point deduction.