

Class:
Plant

```
Int locationX = random, locationY = random;
Double energyLevel = 0;
Boolean eaten;
Boolean inShade;

Public void setLocation();
Public boolean reproduce();
Public boolean eaten();
Public int getLocation();

// method to simulate a day for a plant, used to
// determine if enough energy has been absorbed
// to support reproduction
Public dayPass(int dayCount);
```

Class:
Herbivore

```
Int locationX = random, locationY = random;
Double energyLevel = 0;
Boolean eaten;
Boolean starved;

Public void setStartLocation();

// use for loop to run through each plant object
// and compare location with this specific
// herbivore, if a match the plant is eaten
Public Boolean compareLocation();

Public int dayWalk(Plant[] array, int numOfPlant);
returns number of plants eaten that day

Public boolean reproduce();
Public boolean eaten();
Public Boolean starved();
Public int getLocation();
```

Class:
Carnivore

```
Int locationX = random, locationY = random;
Double energyLevel = 0;
Boolean starved;

Public void setStartLocation();

// use for loop to run through each herbivore object
// and compare location with this specific
// carnivore, if a match the plant is eaten
Public Boolean compareLocation();

Public int dayHunt(Herbivore[] array, int
numOfHerbivores); returns number of herbivores
eaten that day

Public boolean reproduce();
Public Boolean starved();
```

Class:
WildLand

```
Int startQuantity; (for all object types, defined by user)
Double reproductionEnergy; (determines Plant lifecycle time, defined by
user)
Double totals; (totals of each object type, how many were eaten and
how many starved)
Int runtime = 100; (simulates a 100 days)

Public static void main(String[] args){;
Public static void setStartQuantities();

// create initial starting objects using for loop and array
For(object : startQuantities){
Object[] organism = new Organism[startnum];
}
Public static void fullRun(Organism[] array, int count){
// use nested for loops to run through each object on each day of the
run
For(organism[] : full run time){
For(organism[] : organism count){
Plant[]{ dayPass();
Eaten(); (if eaten, remove from simulation)
Reproduce(); (if reproduce successful, create new plant object)}
Herbivore[]{ dayWalk();
Eaten(); (if eaten, remove from simulation)
Starved(); (if starved, remove from simulation)
Reproduce(); (if reproduce successful, create new herbivore object)}
Carnivore[]{dayHunt();
Starved(); (if starved, remove from simulation)
Reproduce(); (if reproduce successful, create new carnivore object)}
}}
}

// display tracked stats at the end of the full run (i.e. final organism
totals, death totals, and percentages)
Public static void output();
}
```