10h May 2020

**3. assignment/6. Task**

**ARBAB ALI**

arbabniz123@gmail.com

# Task

We are simulating the animals of the tundra. There are colonies of prey and predator animals. The number of animals in a colony affect the number of animals in other colonies. There are three predator species: the snowy owl, the arctic fox and the wolf. There are three kinds of prey: the lemming, the arctic hare and the gopher. If the number of prey animals increase, predators can reproduce more quickly. If the number of prey is very large, most of them will wander away because they cannot find enough food. If the number of predators is large, the number of the prey decreases quicker as they are preyed upon. Each colony has a name, a species, and the number of animals in the colony. The prey species are affected by the different predator species as follows. The number of animals in their own colony changes first, then they influence the predators.

**Lemming:** If they are preyed upon by a predator colony, the number of animals in their colony decreases by four times the number of animals in the predator colony. The number of animals in their colony doubles every second turn. If there are less than 20 animals in the colony, all predator colonies have 20% less offspring’s. If there are more than 200 animals in the colony, the number of animals in the colony decreases to 30

**Hare:** If they are preyed upon by a predator colony, the number of animals in their colony decreases by double the number of animals in the predator colony. The number of animals in their colony grows by 5

0 percent (to one and a half times their previous number) every second turn. If there are less than 10 animals in the colony, every predator colonies have20%less offspring s. If there are more than 100 animals in the colony, the number of animals in the colony decreases to 20

**Gopher:** If they are preyed upon by a predator colony, the number of animals in their colony decreases by double the number of animals in the predator colony. The number of animals in their colony doubles every fourth turn. If there are less than 30 animals in the colony, every predator colonies have20%less offspring. If there are more than 200 animals in the colony, the number of animals in the colony decreases to 40.

Predators choose and attack a prey colony randomly in each turn. If there are not enough animals in the attacked colony (for example, there are not four times the number of predators in a lemming colony), the number of predators also decreases: every fourth predator out of the ones who didn't get prey perishes. Predators have offsprings every eighth turn. Normally, the snow owls have 1 offspring per 4 animals, the foxes have 3 offsprings per 4 animals, and the wolves have 2 offsprings per 4 animals. If the number of all prey animals is more than 10 times the number of predator animals, predators have 1 more offsprings: owls have 2 per 4 animals, foxes have 4 per 4 animals, and wolves have 3 per 4 animals. The program should read the colonies from a text file. The first line contains the number of prey and predator colonies separated by a space. Each of the next lines contains the data of one colony separated by space: their name, their species, their starting number of animals. **The species can be: o -owl, f -fox, w -wolf, l -lemming, h -hare, g -gopher.**

Simulate the process until each of the prey colonies becomes extinct or the number of prey animals quadruples compared to its starting value. Print the data of each colony in each turn. The program should ask for the name of the input file and display its contents. You can assume that the input file is correct.

A possible input:

3 3

lem1 l 86

lem2 l 90

hare1 h 26

go g 12

hungry w 12

feathery o 6

# Analysis

# LEMMING:

|  |  |
| --- | --- |
| conditions | changes |
| Every 2nd attack | NoOfAnimals\*2 |
| Each attack | NoOfAnimals := NoOfAniamls – PredatorAnimals \* 4 |
| NoOfAnimals>200 | NoOfAnimals := 30 |
| NoOfAnimals<20 | PredatorAnimals := predatorAnimals\*0.8  (20% less) |

**HARE:**

|  |  |
| --- | --- |
| conditions | changes |
| Every 2nd attack | NoOfAnimals\* 1.5 |
| Each attack | NoOfAnimals := NoOfAniamls – PredatorAnimals \* 2 |
| NoOfAnimals>100 | NoOfAnimals := 30 |
| NoOfAnimals<10 | PredatorAnimals := predatorAnimals\*0.8  (20% less) |

**GOPHER:**

|  |  |
| --- | --- |
| conditions | changes |
| Every 4th attack | NoOfAnimals\*2 |
| Each attack | NoOfAnimals := NoOfAniamls – PredatorAnimals \* 2 |
| NoOfAnimals>200 | NoOfAnimals := 40 |
| NoOfAnimals<30 | PredatorAnimals := predatorAnimals\*0.8  (20% less) |

**SNOW OWLS**:

|  |  |
| --- | --- |
| conditions | changes |
| Every 8th attack and NofAnimals<preyAnimals\*10 | noOfAnimals := NoOfAnimals\*1.25 |
| Every 8th attack and NofAnimals>preyAnimals\*10 | noOfAnimals := NoOfAnimals\*1.50 |
| NoOfAnimals<4\*preyAnimals | NoOfAnimals := NoOfAnimals\*0.75 |

**ARCTIC FOXES**:

|  |  |
| --- | --- |
| conditions | changes |
| Every 8th attack and NofAnimals<preyAnimals\*10 | NoOfAnimals := NoOfAnimals\*1.75 |
| Every 8th attack and NofAnimals>preyAnimals\*10 | NoOfAnimals := NoOfAnimals\*2 |
| NoOfAnimals<4\*preyAnimals | NoOfAnimals := NoOfAnimals\*0.75 |

**THE WOLF**:

|  |  |
| --- | --- |
| conditions | Changes |
| Every 8th attack and NofAnimals<preyAnimals\*10 | NoOfAnimals := NoOfAnimals\*1.50 |
| Every 8th attack and NofAnimals>preyAnimals\*10 | NoOfAnimals := NoOfAnimals\*1.75 |
| NoOfAnimals<4\*preyAnimals | NoOfAnimals := NoOfAnimals\*0.75 |

# Plan

To describe the Preys and Predators, 9 classes are introduced: base class *Colony* to describe the general properties and 2 children for the concrete species of each : Lemming , Hare , Gopher and Snow Owl , The Wolf ,The Arctic Fox . Regardless the type of the creatures, they have several common properties, like the name (*name*) and the specie (*speice*), the getter of its name (*name()*), if it is Prey or Predator (*getSpecie()* ) and it can be examined what happens when predator attacks a prey . This latter operation (*transmute()*) modifies the number of animals of the creature and transmutes the number of predators . Operations getSpecie() ,get*name() ,getNumberOfAnimals()* may be implemented in the base class(Colony) already, but *transmute()* just on the level of the concrete classes(Prey and Predator) as its effect depends on the species of the each other. Therefore, the general class *Colony* is going to be abstract, as method *transmute()* is abstract in two other abstract classes Prey and Predator and we do not wish to instantiate such class.

General description of the Predators is done the base class *Predator* from which concrete species are inherited: *Snow Owl, The Arctic Wolf* , and *The Fox* . Prey has three class inherited class , each of that show how number of Animals of Lemming, Hare and Gopher changes after being attacked by predators. Objects are referred by pointers.

The special Colony classes initialize the name and number of Animals through the constructor of the base class and Two other abstract class Prey and Predetors override the operation *transmute()* in a unique way for each child class. According to the tables of analysis , in method *transmute()*, conditionals have to be used in which the type of the prey is being attacked Though, the conditionals are not effective if the program might be extended by new predator types, as all of the methods *transmute()* in all of the concrete classes(Prey and Predator) have to be modified. To avoid it, design pattern Prey is applied where the predator classes are going to have the role of the attacker on prey.

Methods *transmute()* of the Prey and Predator expect a object of each other as an input parameter as a visitor respectively and calls the methods which corresponds to the species of the colony.

A picture containing screenshot

Description automatically generated

We read information about Animals of Tundra from the given file and create appropriate objects. On each simulation, we calculate the number of animals affected in both objects of prey and predator and determine what number of animals are remain and count the number of attacks as well.

ALGORITHM

SPECIFICATION:

A ( pred:Predator, prey:Prey, L :Bool)

Pre = Prey0 ∩ Predator0

Post = (

( l,elem,t) = SEARCHeEt cond(e) )

ANALOGY

OPTIMISTICL LINEAR SEARCH

|  |  |
| --- | --- |
| ENOR(e) | 1..n |
| F(e) | Transmute(e) |
| cond (e) | (preys[j]->getNoOFAnimals()<(4\* preys[j]->getFirstPopulation()) && preys[j] - >getNoOFAnimals()>0) |
|  | |

INNER SEARCH

|  |  |  |
| --- | --- | --- |
| l := true | | |
| i := 1..n ^ l | | |
|  | (preys[j]->getNoOFAnimals()<(4\* preys[j]->getFirstPopulation()) && preys[j] - >getNoOFAnimals()>0) | |
| l := False | SKIP |

OUTTER SEARCH

|  |  |
| --- | --- |
| i := 1, l := false | |
| l := <> | |
|  | preys[i]->transmute(\*predators[i]); |
| i := i +1  i :=i mode predators.size();  l = INNNERSEARCH |

**Testing**

*-Examination of function transmute () according to their conditions*

3 different cases depending on the prey.

*-READING AND CONSTRUCTION OF DATA CHECKED*

*-Exception handling*

Wrong filename

*SUMULATION TESTING:*

*-* whenEACH OF THE PREY COLONIES BECOMES EXTINCT.

- WHEN NUMBER of PREY ANIMALS quadruples compared to its starting value*.*

*-OUTTER LOOP*

*1.length-based*

*-One Prey and Many Predator*

*-More Prey and Many Predator*

*-Inner Loop*

*1.Length-based*

*-One Prey and One Predator transmutes properly*

*-One Prey and Many Predator*

*2.FIRST and Last*

*-First Predator transmutes properly depending on the last species of Prey*

*-Last Predator Transmute properly depending on the first species of Prey*