

Packages, Inheritance, and Polymorphism

1. Predator-Prey Simulation

40 points

In this assignment, we will incorporate packages and inheritance into our programs. This assignment will demonstrate two major principles of object oriented programming, inheritance and polymorphism.

Polymorphism means one name, many forms. The main idea of polymorphism is the ability to associate many meanings to one method, making the method execute differently depending on in what context the method is called. This assignment also gives us the opportunity to practice using appropriate access modifiers, the final modifier, and packages. Packages are an excellent way to keep pieces of code that are common to each other separate from other pieces of code. They allow us to better organize our code. The classes that are built into java are already in packages and we use the import statement to gain access to them. This was seen when we used the Scanner class to take keyboard input from the user.

Part 1

In this assignment you will create a simple 2D predator–prey simulation. In this simulation, insects are the prey, and birds are the predators. These creatures live in a world composed of a 20 by 20 grid of cells. Only one creature may occupy a cell at a time. The grid is enclosed, so a creature is not allowed to move off the edges of the grid. Time is simulated in moves. Each creature performs some action every time it moves. These actions can include breeding, eating, or starving.

The insects behave according to the following model:

- **Move**
Each move, an insect will look for an empty adjacent cell up, right, left, or down (in that order) and move to the first empty cell it finds. If the cell in the selected direction is occupied or would move the insect off the grid, then the insect looks in the next direction. If an empty cell is not found in either direction, the insect does not move.
- **Breed**
If an insect survives for three moves, then after the third move, the insect will breed. This is simulated by creating a new insect in an adjacent cell (up, down, left, or right – again in that order) if that cell is empty. If there is no empty cell available, no breeding occurs. Once an offspring is produced, the insect cannot produce an offspring until three more moves have happened.

The birds behave according to the following model:

- **Move**
Each move, if there is an adjacent cell (up, down, left, or right, in that order) occupied by an insect, then the bird will move to that cell and eat the insect. Birds cannot eat other birds or creatures, only insects. If an insect is not found to eat, the bird will move into the next adjacent empty cell by looking up, down, left, or right, in that order.
- **Breed**
If a bird survives for eight moves, then after the eighth move, the bird will breed. This is simulated by creating a new bird in an adjacent cell (up, right, left, or down – again in that order) if that cell is empty. If there is not an empty cell available, no breeding occurs. Once an offspring is produced, the bird cannot produce an offspring until eight more moves have happened.
- **Starve**
If a bird has not eaten an insect within the last three moves, then at the end of the third move, it will starve and die. The bird should then be removed from the grid of cells.

Create a super class named Creature that encapsulates basic data common to both insects and birds. This class should have a method named move that a Creature will use to move according to its rules. You

may need additional data structures to keep track of which creatures have moved... You will also need other methods that model all of the necessary Creature behaviors.

Create a class named World that will represent the environment and store the 20 by 20 grid that holds the Creatures. This class is responsible for looping through the grid to simulate a move and display the final grid. In each move, the World should loop through the Creatures and attempt to move each Creature once. Care must be taken that if a Creature is moved down or to the right, it is not moved a second time in that iteration. Once all Creatures have attempted to move once, the World should iterate through the grid and kill off any Creatures that have starved. Then finally, the World should look for those creatures that should breed and create its offspring. To display the resulting grid, the World will draw the grid using ASCII characters of "o" for an insect and "B" for a bird.

Your classes should use appropriate access modifiers and packages.

Write a program to implement this simulation using the classes you created. Initialize the world with 15 birds and 100 insects. The program should simulate one complete move, and then prompt the user to press Enter to simulate another move or Q to quit.

Part 2

Add a new Creature to the simulation. This Creature is a Cat. The cat is represented by a "C" when the grid is displayed.

The Cats behave according to the following model:

- **Move**
Each move, if there is an adjacent cell (up, down, left, or right, in that order) occupied by an bird, then the cat will move to that cell and eat the bird. Cats cannot eat other cats or creatures, only birds. If an bird is not found to eat, the cat will move into the next adjacent empty cell by looking up, down, left, or right, in that order.
- **Breed**
If a cat survives for ten moves, then after the tenth move, the cat will breed. This is simulated by creating a new cat in an adjacent cell (up, down, left, or right– again in that order) if that cell is empty. If there is no empty cell available, no breeding occurs. Once an offspring is produced, the cat cannot produce an offspring until ten more moves have happened.
- **Starve**
If a cat has not eaten an bird within the last twelve moves, then at the end of the twelfth move, it will starve and die. The cat should then be removed from the grid of cells.

Create the new Cat class and modify the simulation program to initialize the world with 15 birds, 100 insects and 10 cats.

2. Online Quiz

10 points

Complete the online quiz "Packages, Inheritance, and Polymorphism "in this week's assignment folder of Blackboard.

What to Submit

Submit a zip file containing the all of the application source files and directories. Your application must compile when unzipped. You only need to submit the final version of the program that runs the simulation using three Creatures.

This is a two part submission. In addition to submitting the zip file, you must **complete the online quiz**.