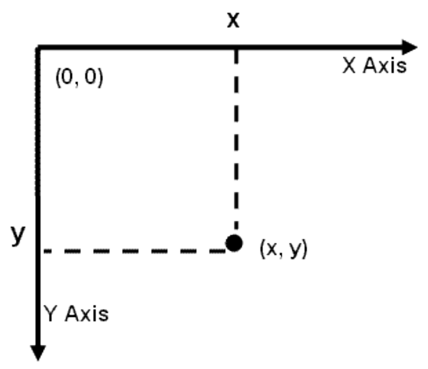
CSCI 2073 – Programming Assignment 1



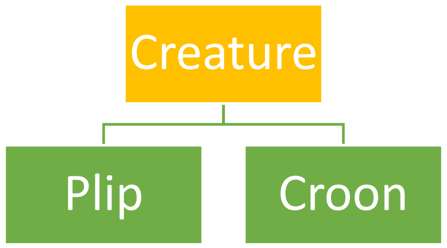
A distant world, which we will imagine as a matrix of cells with dimensions 20 x 20, is inhabited by a variety of creatures. We will assume each dimension is indexed from 0 to 19, so that coordinates (0, 0) represent the upper left (i.e., Northwest) corner of the world. Every creature has a location in terms of (x, y) coordinates and should respond to the `int` methods `getX()` and `getY()`.

Your task is to model two specific types of creatures: Plips and Croons. A Plip is a creature capable of absorbing the sun's energy. A Croon is a creature that lives by preying on Plips. Plips have a certain number of life units; Croons have a strength factor (a fraction between 0 and 1).

All creatures, including Plips and Croons, can move around the world by invoking the `void move()` method. A Plip moves east by one cell unless it is adjacent to the world's eastern boundary (x = 19), in which case it moves west by one cell. Future moves will cause it to continue moving west until it hits the world's western boundary, which causes it to resume future moves in an eastern direction. A Croon moves north by one cell unless it is adjacent to the world's northern boundary (x = 0), in which case it moves south by one cell. Future moves will cause it to move south until it hits the world's southern boundary, which causes it to resume future moves in a northern direction. A Plip loses two units of life when it moves (so it must have at least two life units before moving); a Croon loses 5% of its strength when it moves (and it must have a strength factor of at least .25 before moving).

In addition, a Plip can invoke method `void absorb()` in order to gain one life unit from the sun and method `int getLife()` to return the number of life units. A Croon can invoke method double `getStrength()` and method `boolean preyOn(Plip p)` when it wants to consume the Plip given as argument (for this to occur, they both must be at the same coordinates and the Plip must have at least one life unit). If successful, the Croon doubles its strength (subject to the 1.0 maximum strength), the Plip loses all of its life, and the Plip is incapable of further actions (move or absorb). The method returns true if the Croon is successful, false otherwise.

You are to model these creatures by implementing an abstract Creature class as well as Plip and Croon subclasses. For both subclasses, the contents returned by `toString` should be attractively labeled and include, at a minimum, the creature's location as well as either life units or strength. The subclasses are to provide respective constructors:



`Plip(int lifeUnits, int x, int y)`: creates a Plip with the given number of units of life at the given (x, y) coordinates (where 0 ≤ x ≤ 19 and 0 ≤ y ≤ 19).

`Croon(double strength, int x, int y)`: creates a Croon with the given strength factor at the given (x, y) coordinates (where 0 ≤ x ≤ 19 and 0 ≤ y ≤ 19).

To test the basic syntax and functionality of your classes, the *CreatureTest.java* program should be downloaded and stored in the same folder as your classes. Compile and execute *CreatureTest.* If this test program and your classes do not work well together, you need to modify your classes, not the test program. Once the test program works with your files as expected, submit the three files below (no .class files) to Mimir for final testing: Creature.java, Plip.java, and Croon.java

Be sure to document your code using Javadoc format.

Each of your classes must have the following documentation at the top of the .java file:

```
/**

   Date:

   Course:

   Description:

   On my honor, I have neither given nor received unauthorized help while
   completing this assignment.

   Name and CWID.

*/
```

Each of your methods must have the following documentation preceding the method header:

```
/**

   Method description
   @param arg description of each arg
   @return description of value returned

*/

public int sample(String arg)

{

}
```