Com S 228 Spring 2018

Project 1: Circle of Life (120 pts)

Due at **11:59pm**

Friday, February 2

1. Problem Description

This project simulates (although unrealistically) interactions among different forms of life in the Amazon jungle. The rainforest jungle is represented by an $N \times N$ grid that changes over a number of cycles. Within a cycle, each square is occupied by one of the following five life forms:

Jaguar (`J`), Puma (`P`), Deer (`D`), Grass (`G`), and Empty (`E`)

An Empty square means that it is not occupied by any life form.

Below is a jungle example as a 6×6 grid.

```
P5 E  E  P0 E  E
J3 P1 J0 D0 G  D0
D0 E  D2 J0 J2 G
J0 E  E  D1 P0 E
J1 E  E  G  E  D0
G  G  E  J0 D2 E
```

Both row and column indices start from 0. In the example, the (1, 1)th square is occupied by a 1-year-old Puma. It has a 3×3 neighborhood centered at the square:

```
P5 E  E
J3 P1 J0
D0 E  D2
```

The (0, 0)th square `P5` (a 5-year-old Puma) has only a 2×2 neighborhood:

```
P5 E
J3 P1
```

Meanwhile, the (2, 0)th square `D0` (a newborn Deer) has a 3×2 neighborhood:

```
J3 P1
D0 E
J0 E
```

Generally, the ***neighborhood*** of a square is a "3×3" grid which includes only those squares lying within the intersection of the jungle with a 3×3 window centered at the square. When a square is on the border, the dimension of its neighborhood reduces to 2×3, 3×2, or 2×2.

2. Survival Rules

The jungle evolves from one cycle to the next one. In the next cycle, the life form to reside on a square is decided by those life forms in the ***current cycle*** who live in the 3×3 neighborhood centered at the square, under a set of survival rules. These rules are specified according to the life form residing on the same square in the current cycle. Jaguars, pumas, and deers will grow one year older in the next cycle.

2.1 Jaguar

A jaguar dies of old age or hunger, or from an attack when jaguars are considerably outnumbered by pumas in the neighborhood. The life form on a Jaguar square in the next cycle will be

a) Empty if the Jaguar is at age 5 in the current cycle;
b) otherwise, Puma, if there are at least twice as many Pumas as Jaguars in the neighborhood;
c) otherwise, Empty, if Jaguars and Pumas together outnumber Deers in the neighborhood;
d) otherwise, Jaguar (the Jaguar will live on).

The new life form taking over the square, if a Puma, will have age 0 when the next cycle starts.

For example, in the following neighborhood of a Jaguar at age 2:

```
D0 G  D0
J0 J2 G
D1 P0 E
```

there are two Jaguars (including self), one Puma, and three Deers. Going down the rule list, rule d) applies. According to this rule, the central element (square) will still be `J` (Jaguar) --- one year older--- in the next cycle. In other words, `J2` will be replaced with `J3`.

2.2 Puma

A Puma dies of old age, hunger, or a Jaguar attack. The life form on a Puma square in the next cycle will be

a) Empty if the Puma is currently at age 4;
b) otherwise, Jaguar, if there are more Jaguars than Pumas in the neighborhood;
c) otherwise, Empty, if Jaguars and Pumas together outnumber Deers in the neighborhood;
d) otherwise, Puma (the Puma will live on).

The new life form, if a Jaguar, will have age 0 when the next cycle begins.

For example, in the following neighborhood of a Puma at age 1:

```
P5 E  E
J3 P1 J0
D0 E  D2
```

there are two Pumas, two Jaguars, and two Deers. Rule c) applies. The central square will become `E` in the next cycle.

2.3 Deer

A Deer dies of old age or hunger. It may also be eaten by a Jaguar or a Puma. More specifically, the life form on a Deer square in the next cycle will be

a) Empty if the Deer's current age is 6;
b) otherwise, Empty if there is no Grass in the neighborhood (the Deer needs food);
c) otherwise, Puma if in the neighborhood there are more Pumas and Jaguars together than Deers, and furthermore, if there are at least twice as many Pumas as Jaguars;
d) otherwise, Jaguar if there are more Pumas and Jaquars together than Deers, and if there are at least as many Jaguars as Pumas;
e) otherwise, Deer (the Deer will live on).

If the new life form is a Jaguar or Puma, it will have age 0 when the next cycle starts.

In the following neighborhood of a Deer at age 2:

```
P1 J0 D0
E  D2 J0
E  E  D1
```

lives two Jaguars, one Puma, and three Deers. Rule a) does not apply because the Deer is only 2-years old. Rule b) does since there is no Grass in the neighborhood. The central element (square) will be `E` in the next cycle according to this rule.

2.4 Grass

Grass may be eaten out by overcrowded Deers. Deers may also multiply fast enough to take over the Grass square. In the next cycle, the life form on a Grass square will be

a) Empty if at least three times as many Deers as Grasses in the neighborhood;
b) otherwise, Deer if there are at least four Deers in the neighborhood;
c) otherwise, Grass.

If the new life form is a Deer, it will be aged 0 when the next cycle starts.

For example, if the neighborhood of a Grass is

```
P0 E  E
D0 G  D0
J0 J2 G
```

the central element will be `G` in the next cycle under rule c).

2.5 Empty

Empty squares are competed by other life forms. More specifically, the life form on an Empty square in the next cycle will be

a) Deer, if more than one neighboring Deer;
b) otherwise, Puma, if more than one neighboring Puma;
c) otherwise, Jaguar, if more than one neighboring Jaguar;
d) otherwise, Grass, if at least one neighboring Grass;
e) otherwise, Empty.

If the new life form is a Jaguar, Puma, or Deer, it will have age 0 when the next cycle begins.

For example, suppose that an Empty square in the top row of a jungle has the following neighborhood:

```
P0 E  E
D0 G  D0
```

which includes two Deers. Thus, rule a) applies to change the central element to `D0` in the next cycle.

3. Task

You will implement an abstract class `Living` to represent a generic life form. It has three subclasses `Animal`, `Empty`, and `Grass`. The first subclass, implementing the interface `MyAge`, is abstract, and needs to be extended to three subclasses: `Jaguar`, `Puma`, and `Deer`. You also need to implement a `Jungle` class which has a public member `Living[][]` to represent a grid jungle.

The class `CircleOfLife` repeatedly simulates evolutions of input jungles, which are either randomly generated or read in from files. In each iteration, the class interacts with the user who chooses how the input jungle will be generated, and enters the number of cycles to simulate. The iteration prints out the initial jungle and the final jungle.

Your random jungle generator may follow the uniform probability distribution so that `Deer`, `Empty`, `Grass`, `Jaguar`, and `Puma` have equal likelihoods to occupy every square. Or you may use a different distribution, as long as no life form has zero chance to appear on a square.

Java provides a random number generator. To use it, you need to import the package `java.util.Random`. Next, declare and initiate a Random object like below

```
Random generator = new Random();
```

Then, every call below

```
generator.nextInt(5)
```

will generate a random number between 0 and 4 that corresponds to one of the five life forms.

Jungle creation from an input file will weigh more than random creation in our grading. When zero or a negative number of cycles is entered by the user, your code does nothing but waits for a positive input.

A new living form, if a Jaguar, Puma, or Deer, always has age 0 at its creation, whether initially by the class `Jungle` or later on under a survival rule. After surviving a cycle, its age increases by one.

Templates are provided for all classes. Be sure to use the package name `edu.iastate.cs228.hw1` for the project.

Below is a sample simulation scenario over three initial jungles. In the first iteration, the user entered 1 for a randomly generated jungle, 3 to specify the grid to be 3×3, and 1 to simulate just one cycle. The simulator printed out the initial and final jungles. The second iteration simulated a randomly generated 6×6 grid over 8 cycles. In the third iteration, the user typed 2 for a file input, entered the file name "`public3-10x10.txt`", and specified 6 cycles. (The file `public3-10x10.txt`, describling a 10 x 10 grid, resides in the same folder containing the `src` folder.) After the third iteration, the user typed 3 to end the simulation. (Any number other than 1 and 2 could have ended the simulation.)

```
Circle of Life in the Amazon Jungle
keys: 1 (random jungle)  2 (file input)  3 (exit)

Trial 1: 1
Random jungle
Enter grid width: 3
Enter the number of cycles: 1

Initial jungle:

G   J0  G
J0  D0  P0
J0  D0  E

Final jungle:

G   E   G
E   J0  P1
J1  E   D0

Trial 2: 1
Random jungle
Enter grid width: 6
Enter the number of cycles: 8

Initial jungle:

E   G   D0  D0  J0  P0
J0  D0  G   J0  P0  G
J0  G   D0  J0  P0  J0
J0  P0  J0  E   G   G
```

```
E  P0 E  P0 D0 E
J0 G  E  P0 J0 G

Final jungle:

G  G  D0 D0 G  G
G  G  E  D4 D5 G
G  G  D0 E  D5 D4
J0 J1 D2 E  D3 G
J0 J1 D1 D0 P3 G
G  G  G  P0 E  G

Trial 3: 2
Jungle input from a file
File name: public3-10x10.txt
Enter the number of cycles: 6

Initial jungle:

D0 P0 E  P0 E  P0 J0 J0 E  J0
G  G  G  J0 G  D0 P0 E  D0 G
J0 G  P0 D0 G  D0 G  D0 G  D0
P0 E  E  D0 D0 D0 G  D0 J0 D0
D0 D0 D0 P0 D0 J0 J0 P0 G  E
J0 J0 E  E  P0 J0 P0 E  G  D0
D0 E  P0 G  D0 E  D0 P0 D0 G
E  J0 D0 P0 E  G  P0 P0 D0 P0
P0 J0 D0 J0 P0 E  P0 P0 E  J0
D0 J0 J0 D0 P0 E  E  D0 G  J0

Final jungle:

D6 G  G  G  D1 P2 E  D0 J5 E
G  G  G  D3 D2 E  D0 E  D0 E
G  G  D0 D6 E  E  D0 E  D0 E
G  D5 D5 E  E  E  E  J5 E  E
J2 D0 D0 D0 D0 D0 D0 D0 D0 D0
E  E  E  E  E  E  P2 D0 E  E
J0 E  D0 D0 D2 D0 D1 D0 D1 D0
E  J0 J0 D0 D1 E  P0 P1 D1 G
E  J0 P0 E  P0 E  P0 P0 G  G
E  J0 P0 E  P0 E  E  E  G  G

Trial 4: 3
```

Your code should print out the same text messages for user interactions.

4. Input/Output Format

The format for jungles is shown in the sample runs above subject to the following rules.

a) Every square occupies two spaces starting with one of the letters '`D`', '`E`', '`G`', ‘`J`’, and '`P`'.
b) If the letter is ‘`D`’, '`J`', or '`P`', then it is followed by a digit representing the animal’s age; otherwise, it is followed by a blank.
c) There is exactly one blank between two squares, whether represented by a letter and a digit, or a letter and a blank. No blank lines.

You may assume all input files to be ***correctly formatted***, containing ‘`D`’, '`E`', '`G`', '`J`', '`P`', and digits up to 6 as the only non-blank characters. A digit will ***not*** exceed the lifespan of the animal represented by its preceding letter.

5. Junit Classes

JUnit classes include `AnimalTest`, `CircleOfLifeTest`, `DeerTest`, `EmptyTest`, `GrassTest`, `JaguarTest`, `JungleTest`, `LivingTest`, and `PumaTest`. ***You need to implement all these classes***.

6. Submission

Write your classes and JUnit test classes in the `edu.iastate.cs228.hw1` package. Also submit the text files with your JUnit tests. Turn in the zip file not your class files. Please follow the guideline posted under Documents & Links on Canvas.

Include the Javadoc tag `@author` in each class source file. Your zip file should be named `Firstname_Lastname_HW1.zip`.