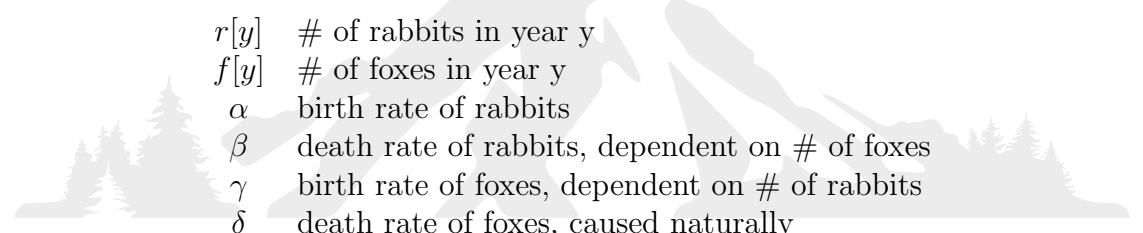


Lab 7: Rabbits & Foxes

You will write a program that calculates the populations of foxes and rabbits who live within a certain area of land, as that number changes from year to year. It will take the initial populations as inputs, and then proceed to calculate the year-by-year changes.

The following is derived from the “Lotka-Volterra equations”, which you may look up if you desire. Essentially, we will model the foxes and rabbits with 4 constants, and 2 arrays of year-by-year values. They are defined as follows:



$r[y]$ # of rabbits in year y
 $f[y]$ # of foxes in year y
 α birth rate of rabbits
 β death rate of rabbits, dependent on # of foxes
 γ birth rate of foxes, dependent on # of rabbits
 δ death rate of foxes, caused naturally

The formulas we will use are:

$$r[y] = r[y - 1] + (r[y - 1] \times \alpha) - (r[y - 1] \times f[y - 1] \times \beta)$$

$$f[y] = f[y - 1] + (f[y - 1] \times r[y - 1] \times \gamma) - (f[y - 1] \times \delta)$$

However, negative values are not allowed! If either formula results in a negative value, you should use 0 instead.

You will start out by asking the user how many years of simulation he or she wants, and what the initial numbers of rabbits and foxes are. Each of these needs to be held in its own int array. The first elements of each array holds the initial populations, and subsequent elements are calculated using the above formulas. (Remember that single-letter names are usually horrible. Don't call your rabbit and fox arrays r and f —those names are just for the equations above.)

We will use the values of $\alpha = 0.2$, $\beta = 0.005$, $\gamma = 0.001$, and $\delta = 0.2$. You should print out each year's populations as you calculate them, and then give a final report at the end with all of the calculations. Here is how your program should appear:

```
Welcome to the fox and rabbit program!

How many years of simulation do you want? 10
Please enter the initial rabbit population: 100
Please enter the initial fox population: 10

Year 0: 100 rabbits, 10 foxes
Year 1: 115 rabbits, 9 foxes
Year 2: 133 rabbits, 8 foxes
Year 3: 154 rabbits, 7 foxes
Year 4: 179 rabbits, 7 foxes
Year 5: 209 rabbits, 7 foxes
```

```
Year 6: 243 rabbits, 7 foxes
Year 7: 283 rabbits, 7 foxes
Year 8: 330 rabbits, 8 foxes
Year 9: 383 rabbits, 9 foxes
Year 10: 442 rabbits, 11 foxes
```

```
Rabbit populations: [100, 115, 133, 154, 179, 209, 243, 283, 330, 383,
442]
```

```
Fox populations: [10, 9, 8, 7, 7, 7, 7, 7, 8, 9, 11]
```

```
Goodbye!
```

Some hints:

- The values of α , β , γ , and δ should all be `public static final` class variables, with proper names.
- Since the fox and rabbit populations are ints, you will have to round them for the program to work. The easiest way to do this is to use the `Math.round()` method. However, this returns a long—so you'll have to cast it to an int.
- Be sure that your program doesn't allow for negative populations. If you start with 1 rabbit and 1000 foxes, the next year there should be 0 rabbits instead of -3 rabbits. (Or is that 3 anti-rabbits? What is an anti-rabbit?)
- You can use the `Arrays.toString()` method to make a nice readable output of an array. This takes an array as its argument, and returns a human-readable string.
- Note that the size of the arrays should be 1 more than the requested years of simulation!

Your final class should be called `RabbitsAndFoxes`.