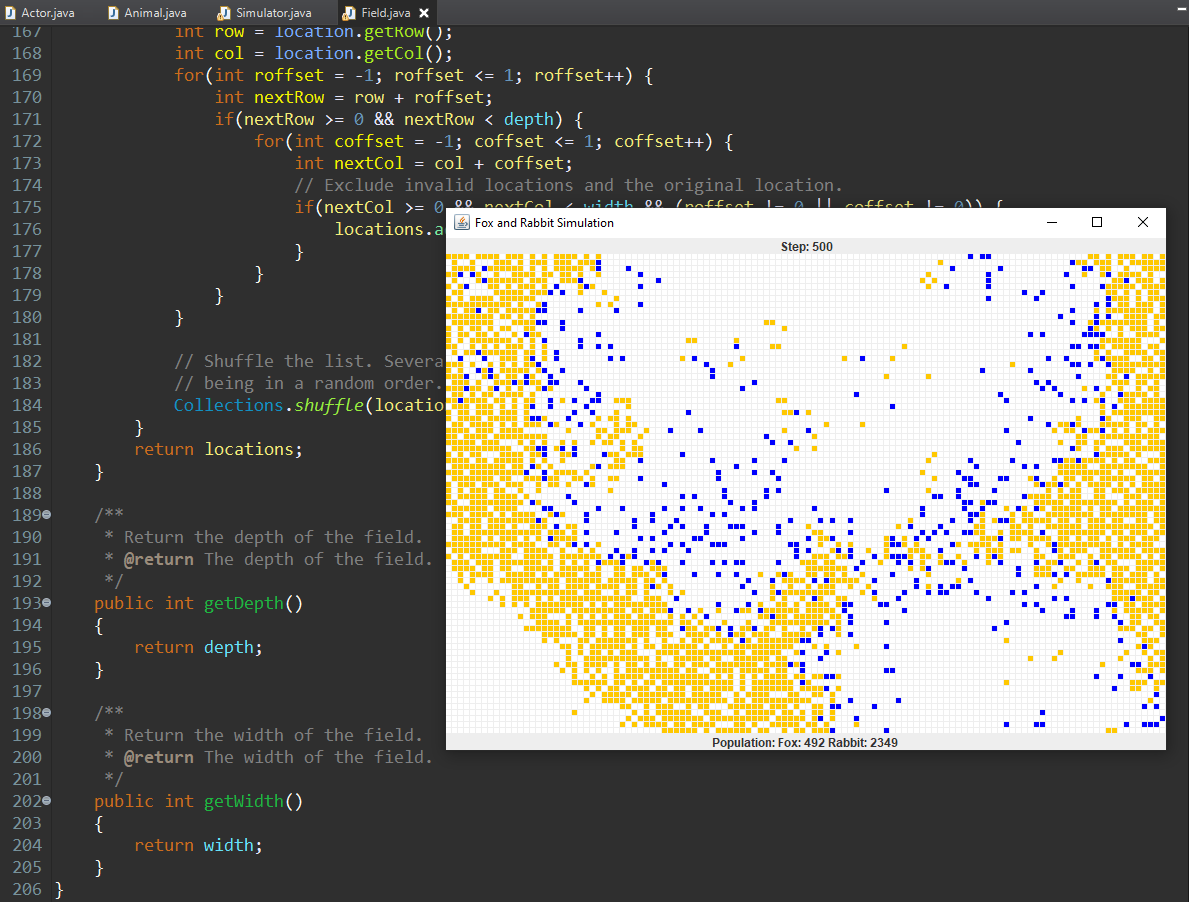
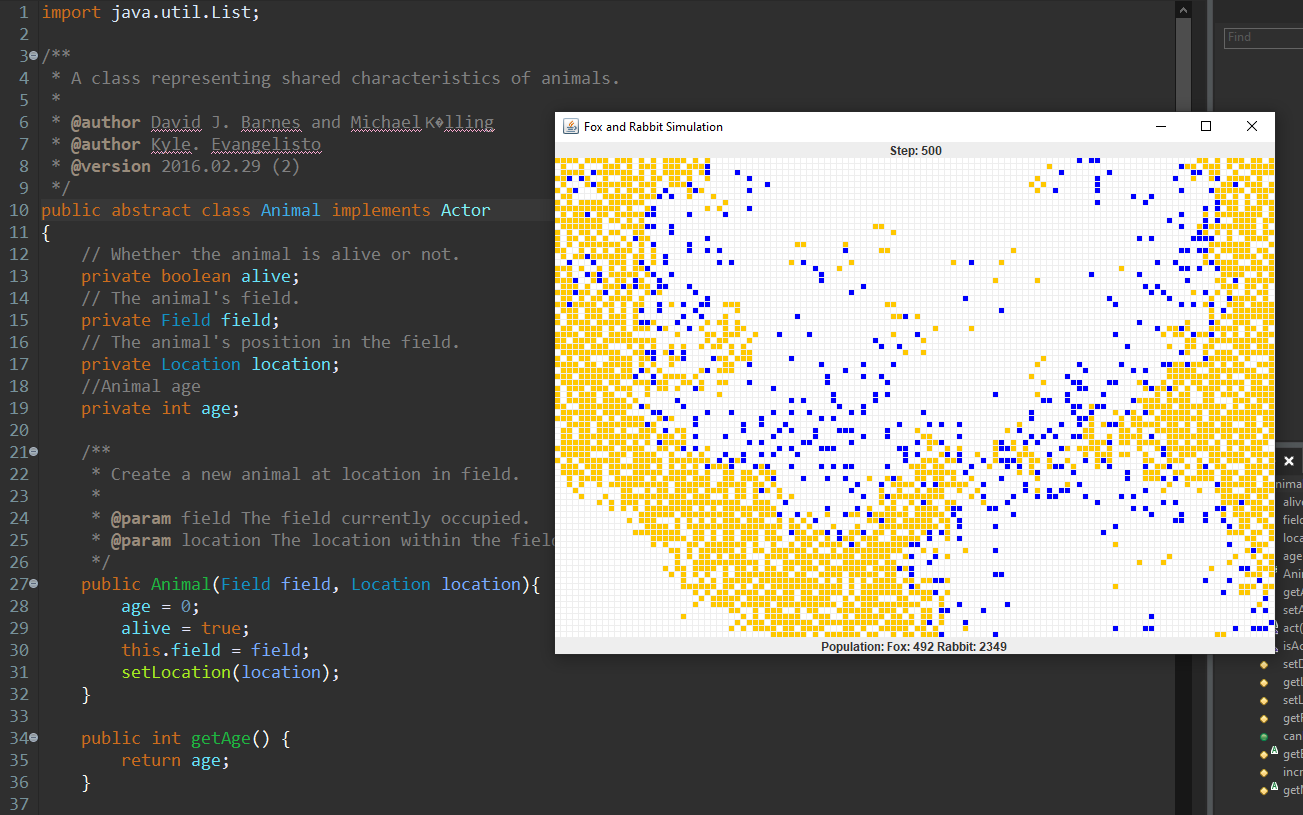
Kyle Evangelisto

Professor Weissman

Hw06-1

07 October 2020

* Created abstract class actor. Implemented java.util.\*;
* Animal now extends Actor -- act(List<Animal> newAnimals) now throws error
* Refactored Simulator.java and simulateOneStep() to utilize a List<Actor>
* Renamed isAlive() to isActive()
* Refactored abstract public void act(List<Animal> newAnimals); to List<Actor>
* Checked the class field for anything to refactor and didn’t notice anything substantial.
* 
* Changed actor from abstract class to interface
* Animal now uses actor interface
* 

10a ) The Actor interface helps us achieve 100% abstraction by hiding all the implementation.

10b) I don’t think making Animal an interface rather than an abstract class is the best idea in this situation here. Yes, 100% abstraction would be achieved but there are a lot of fields, accessors, mutators and other methods used that wouldn’t translate well into an interface.