**P9 – Algorithm Choices**

**Algorithm 1**

I chose to highlight this particular algorithm from the Fantasy Game, as I felt that it was an pulled in a lot of interesting elements. Not only does it pull information from its own class, but it also calls in aspects of other classes that are passed in. I am also particularly pleased with the way we got the algorithm to interact with an enum, and the .toString() and .toLowercase() additions went through several iterations until I found the combination that worked.

For me, this particular piece of code was also when I began to really enjoy and have fun with Java, and I think that shows through in this.

**Threading Algorithm**

This algorithm involves a couple of screenshots in order to see the full extent of what is going on. Essentially, we have three different functions – across 2 different classes and involving inheritance – that run on timers.

I have chosen to show this code off, as it took me several days to really dig into the methodology and understand what was required in order to get them all working in the correct way.

The functions relating to the dinosaurs dig in and control just carnivores on the increase hunger function, but then for the rampage function we can implement this on both carnivores and herbivores, and the randomising element means that we simply do not know which dinosaur is going to escape each time it runs.

The same is true of the visitor codes, adding in the random element allows for simulation of real time park movement rather than a set pattern of 4 people moving in a set direction at a set time. The randomising and timed elements of all of these functions makes the overall program a little bit more realistic and lifelike – or as realistic as you can get when dealing with a fictional dinosaur theme park!