

CSC207: Javadoc and UML

To earn the credit for this lab, you must be present in the lab class. For this lab, and for the rest of the labs in this class, please make sure to submit by end of the lab class.

1 Overview

This week, you will implement two classes based on the UML diagrams. You'll also document the classes by providing doc comments for the Javadoc.

Also, ask your TAs questions about assignment 1!

2 Log in and get things set up.

Use MarkUs to form a group with your lab partner. The MarkUs item for this lab is Lab5. This will require `s1` to log in to MarkUs and invite `s2` to form a group. Then `s2` will need to log in and accept the invitation from `s1`. Take a note of the group MarkUs created for you. It will be of the form `group_nnnn`.

All repositories in this course will have a URL of the form:

```
http://markus.cdf.toronto.edu/svn/csc207-2016-05/repo-name
```

`s1` drives and `s2` navigates.

Now `s1` logs in and opens a new terminal window.

1. Change to `s1`'s home directory.
2. Check out your group's repository.
3. In your repository, you will find a newly created directory called Lab5.
4. Open up Eclipse. Point your workspace to the folder `repo-name`.
5. Using Eclipse, create a new Java project called Lab5. Make sure you open the Java perspective.
6. Create a new package called `lab5`.

3 Class Organism take two.

Please study the UML diagram and implement both classes shown there.

`Organism` is an abstract class. For testing purposes we will instantiate a subclasses of organism called `Arthropod`.

Again, an `Organism` has a name, a pair of coordinates (x,y) in the world, and a movement direction chosen among `north`, `south`, `east`, `west` stored in a static variable `VALID_DIRECTIONS`. This time we will assume the world is unbounded in all directions. So the `move` method can easily move the organism in the chosen direction with the desired speed without worrying!

Write the `toString` and have Eclipse generate getters and setters.

4 The class Arthropod.

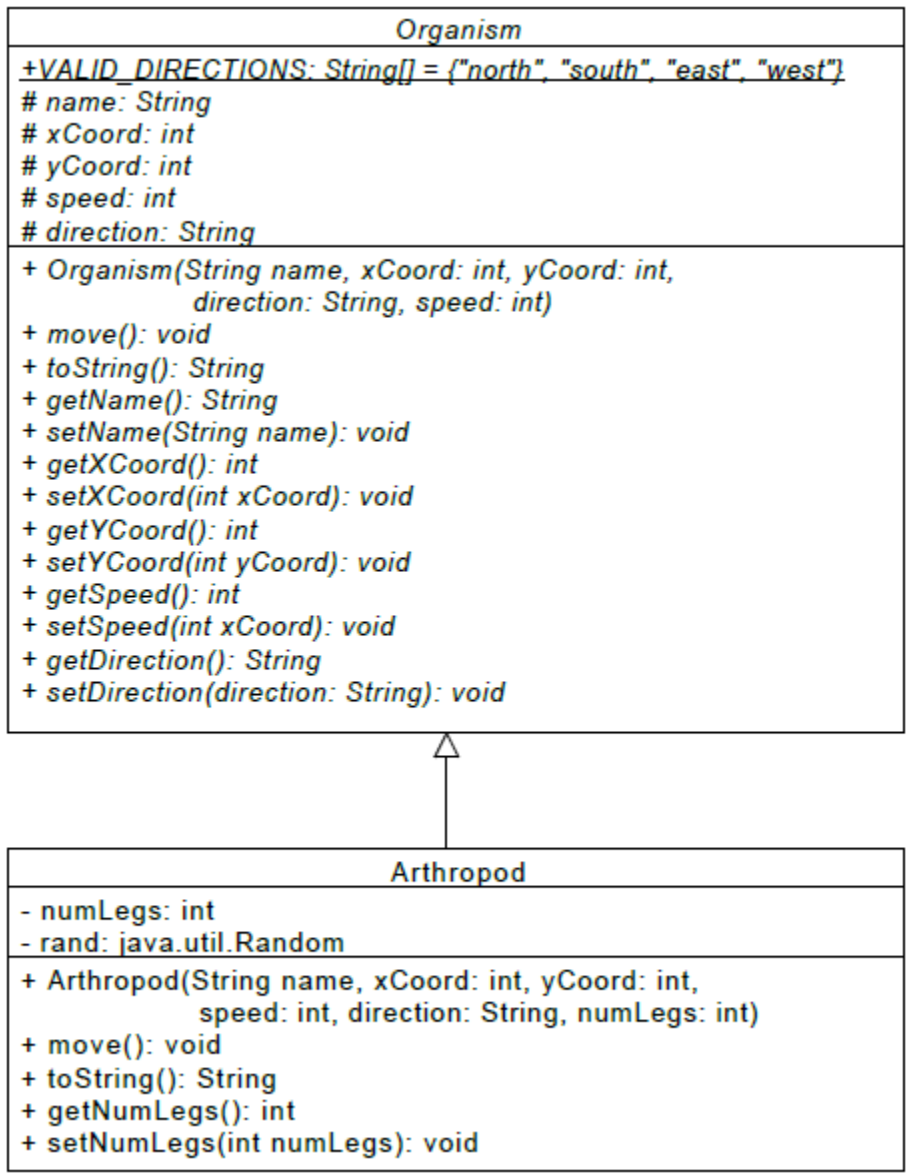
Switch roles!

Write the `Arthropod` class as shown in the UML diagram. Each `Arthropod` has a different number of legs. Try it! That is, create a test class, instantiate an `Arthropod`:

```

Organism lobster = new Arthropod("Homarus gammarus", 0, 0, 2, "north", 10);
System.out.println(lobster);
lobster.ove();
System.out.println(lobster);

```



5 Finishing up

- Show your work to your TA. Make sure to show them the Javadoc.
- Submit your work on Markus.
- If you need help about Assignment 1, don't hesitate to ask!