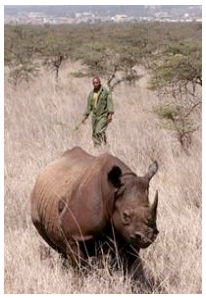
CS1110 Fall 2008 Assignment A3 Monitoring Rhinos 2
Due (submitted on the CMS) by Monday, 29 September



This assignment continues with A1. Start with your A1 classes Rhino and RhinoTester, which should be correct. You can group all your test cases for this assignment in one test procedure.

Keep track of the time you spend on this assignment. When you submit it, tell us how many hours you spent on it.

You may work in groups of 2, but you must do all the work together, sitting at the computer together. For example, for one person to write functions and the other person to write test cases for them, with no interaction, is dishonest. If you want to form a group, *do it immediately*. *Both* partners must take action in the CMS before the group will be formed.

The deadline is Monday, 29 September, so that it is completed before the first prelim. Because you have a lot of work to do, including studying for the prelim, we have considerably reduced the work involved in this assignment by removing 5-6 functions from previous years.

In assignment A1, if you made a mistake, we asked you to fix it and resubmit. In this assignment, if you make a mistake, points will be deducted. Points may be deducted for lateness. Your new functions must have suitable javadoc comments, there must be appropriate test cases, and all methods should be correct. *You can achieve all this most easily by writing and testing one method as a time*.

In this assignment, you will add to the class Rhino of A1: (1) a static variable and (2) more comparison methods, using boolean expressions.

The steps to perform in doing this assignment

Step 1. Static variable. Add to class `Rhino` a **private static** variable that contains the number of `Rhino` objects that have been created so far. WHENEVER A RHINO OBJECT IS CREATED, THIS FIELD SHOULD BE INCREASED BY 1. Provide suitable test cases.

Add a **static int** function `getPopulation`, with no parameters, that will return the value of the static variable. Change the constructors so that they properly maintain the value of the static variable. Finally, add test cases to class `RhinoTester` to test whether the static variable is properly maintained.

Step 2. Comparison functions. Write the following functions, all of which produce a boolean value. **Do one at a time**: (1) write the function, then (2) test it thoroughly with test cases in class `RhinoTester`. **Then** proceed to the next function.

Method	Description
isMotherOf(Rhino r)	= "r is not null, and this rhino is r's mother".
isFatherOf(Rhino r)	= "r is not null, and this rhino is r's father".
isParentOf(Rhino r)	= "r is not null, and this rhino is r's parent".
isBrother(Rhino r)	= "r is this rhino's brother". Precondition: `r` is not `null`. Note: r is Rhino A's brother if (1) r and A are two different objects, if r is male, and if r and A have at least one parent in common.

The names of your methods much match those listed above **exactly**, including capitalization. The number of parameters and their order and types must also match. The best way to ensure this is to copy and paste. Our testing will expect those method names and parameters, so any mismatch will fail during our testing. Parameter names will not be tested —you can change the parameter names if you want.

Each method **must** be preceded by an appropriate specification, as a Javadoc comment. The best way to ensure this is to copy and paste from this handout. After you have pasted, be sure to do any necessary editing.

Note carefully the definition of brother and sister in the specs. Use these definitions when writing the methods.

In testing whether two rhino's are the same, do not use the field that contains their names. Instead, use the names on the tabs of their objects.

Finally, isParentOf should be written to call isFatherOf and isMotherOf. We want you to begin seeing how a program can call methods already written instead of doing duplicate work themselves.

Your new method bodies should have no if statements or conditional expressions.

Directions

- Boolean expressions, using the operators `&&` (AND), `||` (OR), and ! (NOT), are sufficient to implement all four functions. You will lose points for using `if` statements or conditional expressions.
- Points will be deducted if isParentOf does not call isFatherOf and isMotherOf.
- Use method `.equals` to compare objects (including String objects) for equality, but when you want to test whether two objects are or are not actually not the same, use `==`.

Submitting the assignment

Check these points before you submit your assignment:

- Did you use an if statement or conditional expression in the new methods? Get rid of it.
- Is each function tested enough? For example, if an argument can be **`null`**, is there at least one test case that has a call on the function with that argument being **`null`**; If not, points will be deducted.
- Did you check your javadoc? Click the javadoc button in the DrJava navigation bar. This will cause the specification of the classes and methods of the classes to be extracted from your program and html pages to be created that contain the specs. Look at those specs carefully and make sure that they are suitable. Can you understand precisely what a method does based on the extracted spec? If not, fix the spec, generate the javadoc, and look at it again.
- ADD A COMMENT AT THE BEGINNING OF FILE RHINO.JAVA THAT SAYS HOW MANY HOURS YOU SPENT ON THIS ASSIGNMENT. We will report the min, average, mean, and max.

Submit only files that end with the `.java`. Be careful about this, because in the same place as your `.java` files you may also have files that end with `.class` or `.java~`. but otherwise have the same name.