UNIVERSITY OF MICHIGAN
EECS 282

Assignment 3 – Monkey Business. 25 points

Due Date: Feb 2nd, 2010, 11:55 PM. Warning: this assignment can be tricky to get right and debug. The bugs can be subtle. Start early. Use plenty of asserts.

Overview

In this assignment, the goal is to represent information about monkeys. In particular, you will store the following information: year of birth, sex (M or F), and known parent-child relationships.

To do this, you will implement a single class, Monkey, in a file called Monkey.java. It will have the public interface that is given in the initial version of Monkey.java that is given to you (we have actually given you an Eclipse project, which contains Monkey.java in its source directory). The Monkey.java that is provided to you has some tests (the calls to which you should uncomment in the main program, when you are ready to run them).

Each monkey must have the following state, which must be stored in private variables:

- year of birth: integer value which should be >= 0.
- A unique ID. You should assign the ID using an auto-incrementing integer that starts at 0. If monkey m2 is created immediately after monkey m1, m2's ID should be one higher than m1's ID.
- sex: whether the monkey is male or female. use the enumerated type Sex that is in the file.
- mom: this is of type Monkey, and is a reference (pointer) to the mother monkey. This may be null, indicating that the Mom is unknown.
- dad: this is of type Monkey, and is a reference (pointer) to the father monkey. This may be null, indicating that the father is unknown.
- Monkey[] children: an array of children monkeys. Maximum size of this array is MAX_CHILDREN, a constant. In the sample code, MAX_CHILDREN is set to 4. But, your code should work even if this value was set at a lower or higher value. Note that, in Java, arrays only contains pointers to objects, not the actual objects.
- numChildren: number of children of this monkey.

There are several invariants on the state of monkeys, which must be maintained when doing updates on the state of the monkeys:

- If a monkey has a mom, then the monkey appears in the mom's children array.
- If a monkey has a dad, then the monkey appears in the dad's children array.
- If a monkey C appears in M's children's array, then M must be C's dad or mom.
- A monkey in this computer world cannot have more than MAX_CHILDREN children.
- A mom or dad monkey must be born in a year that is strictly earlier than any of its children.
- A mom must be female; a dad must be male.
- numChildren must be equal to the number of non-null monkeys in the children array.
- numChildren must be between 0 and MAX_CHILDREN (inclusive)

- ID must be greater than or equal to 0.
- year of birth must be greater than or equal to 0.

You will implement several public methods, the stubs of which are provided to you in the sample Monkey.java file. You can add additional helper methods, which should be kept private to the class. The public methods that you must implement are:

- void testInvariants(): This uses assert statements to test all the above invariants on the monkey's state. If any of the invariants is not true, testinvariants() should have an assertion failure. This function will consist of a series of assertions. To get you started, we have given the assertions for two of the invariants from the above list. You should add in the rest.
- Getters for year of birth, sex, ID, mom, dad, number of children, and the arrays of children. These are getYearofbirth, getSex(), getID(), getMom(), getDad(), getNumChildren(), getChildren(). If you are using Eclipse, Eclipse can generate the getters for you.
- Constructor, which takes the sex and year of birth as arguments
- void setMom(Monkey mom): sets/replaces the mom to the specified mom.
- void setDad(Monkey dad): sets /replaces the dad to the specified dad.
- void addChild(Monkey c): adds c to be a child of this monkey.
- void removeChild(Monkey c): removes c from the list of children
- void unsetMom(): sets the mom to null.
- void unsetDad(): sets the dad to null

You must be careful to maintain the invariants when doing updates. For example, when c.setMom(m) is called, if you simply set c's mom to m, you will fail to maintain the invariants. You must also make sure that you add c to m's children array. Furthermore, if c already had a previous mom, then you must first unset that mom before assigning a new one.

Furthermore, avoid clones of duplicates. A monkey should not be added twice to the children array.

If the user attempts to do an operation that will violate the invariants, make sure you check for that. In that case, the operation should be ignored. As an example, an attempt to set the mom to a male monkey should be ignored. As another example, an attempt to add a child that would cause numChildren to exceed MAX_CHILDREN should be ignored.

No setters need to be provided for sex, year of birth, or ID.

Important: In all your methods that modify the state of a monkey, invoke testInvariants() at the beginning of the method and before returning from the method. This is to make sure that all state modifications maintain the invariants. We have left these checks in some of the stubs to show you how to do that.

We have given you several test methods. These test methods are declared static, since they are not specific to a particular monkey. The test methods create a few monkeys, invoke some methods on them to update their relationships, and then invoke several assertion checks to help make sure that you have implemented the specifications correctly.

Start early on this one. The amount of code is not large, but you will have to think to make sure that invariants are maintained. The assignment also will require the use of recursive functions. To check whether a monkey is an ancestor or a descendant of another, you will need to use recursive functions.

You should add additional tests, if the tests that are provided to you are not adequate to test the specs.

What to submit:

1. Modified Monkey.java.

Note that we may run our own battery of tests on your Monkey class. So, do not modify the names or arguments of public methods – treat it as a conract between you and the 282 teaching staff. You are free to add any private methods or variables to the Monkey class.

You should use javadoc-style comments for all the methods. We have given the comments for some of the methods.

You are expected to add additional tests as well to test the specs for behaviors that are not captured by existing tests. See the comments in the code that explain what is expected.

2. A directory containing javadoc output. We should be able to click on index.html and see your javadoc documentation for the Monkey class. We expect to see all public methods commented.

3. Answers to the following questions:

    a. Why would it be risky to make the mom and dad field in the Monkey class public? Give an example of what a caller could do, which would violate the invariants.
    b. Why would it be risky to make the yearofbirth field as public? Give an example of what a caller could do, which would violate the invariants.
    c. Did you add a static variable to help assign unique IDs? If so, what would happen if that variable were not static?
    d. If your code does not completely work, have you narrowed down the bugs? Explain the status of the code.

4. Output from running the program. Cut and paste the output from the program into a text file, output.txt, and submit that. You can also use I/O redirection if you use java from command-line to capture the output into a file; that may be easier.