Programming Assignment 2: Towers of Hanoi

Description:

Based on Chapter 3, Programming Problem 12

The game *Towers of Hanoi* consists of three pegs and a collection of rings that stack on the pegs. The rings are different sizes. The initial configuration for a five-ring game is shown here, with the first tower having rings from one inch (on the top) to five inches (on the bottom).

Initial configuration for a five-ring game of Towers of Hanoi

The rings are stacked in decreasing order of their size, and the second and third towers are initially empty. During the game, the child may transfer rings one at a time from the top of one peg to the top of another. The goal is to move all the rings from the first peg to the second peg. The difficulty is that the child may not place a ring on top of one with a smaller diameter. There is the one extra peg to hold rings temporarily, but the prohibition against a larger ring on a smaller ring applies to it as well as to the other two pegs.

Assignment:

Create a Towers class that implements the game Towers of Hanoi with three pegs.

Feature	Signature	Requirement
Constructors	Towers(n)	Precondition: 1 <= n <= 64; Postcondition: The towers have been initialized with n rings on the first peg and no rings on the other two pegs. The diameters of the first peg's rings are from one inch (on the top) to n inches (on the bottom).
	Towers()	Creates a default-sized Towers with n = 5
Methods	int getRingCount(int pegNumber)	Precondition: pegNumber is 1, 2, or 3. Postcondition: The return value is the number

		of rings on the specified peg.
	int getTopDiameter (int pegNumber)	Precondition: pegNumber is 1, 2, or 3. Postcondition: If getRingCount(pegNumber) > 0, then the return value is the diameter of the top ring on the specified peg; otherwise, the return value is zero.
	boolean move(int startPeg, int endPeg)	Precondition: startPeg is a peg number (1, 2, or 3), and getRingCount(startPeg) > 0; endPeg is a different peg number (not equal to startPeg), and if endPeg has at least one ring, then getTopDiameter(startPeg) is less than getTopDiameter(endPeg). Postcondition: If precondition is not true, return value is false and pegs are unchanged, otherwise the top ring has been moved from startPeg to endPeg and return value is true.

Specifications:

	Requirements
Project Submission	Your submission shall be an exported Eclipse project jar file The name of the project shall be in the form: *P2_AAAnnn*_TowersOfHanoi AAAnnn will serve as your “student identifier”, where AAA – your initials (two or three characters) nnn – the last 3 digits of your Metro State student ID The jar file must have the Java sources
Code	The Constructors and methods defined in the capabilities table will be public, spelled as specified, and with the return type as specified (void otherwise)

	Your class(es) cannot be in the default package. I would recommend putting them in package `edu.metrostate.ics240.p2.towers` Your submissions will be tested with an automated testing framework Your code will be free of compile-time errors Your code will comply with Java coding conventions