DominoMeanie.java **DominoSet.java** **Domino.java**

When I grew up playing dominoes, there was a game we would play at the end of playing the domino game of 42. This game has many names, but for the purpose of this program we will call it Domino Meanie. The object of the game is to evenly distribute a randomized set of dominoes to all the players. Then, put the dominos back into the box from least to greatest, starting with [0|0] and continuing up to the [Max Pip|Max Pip]. This is done by going to each player consecutively, if the player has the desired domino, then the player places the domino in the box and play moves to the next player. If the player does not have the correct domino, then the player would choose a random player that is still playing and steal one of their dominos. If they steal the correct domino, then they can play it. To simplify this program, we will have the player steal a random domino from the next available player rather than choose a random player. If a player no longer has any dominos in their hand, they are no longer in the game, which is a good thing. The game ends when only one player remains, and that player is the Domino Meanie. You don't want to be the Domino Meanie, because that means you have to put the dominos up and clean up game area…my house was filled with competitive people who didn't like to clean up after other people, and we were vicious when it came to Domino Meanie.

EXPECTED INPUT:
The program will ask the user to input the number of players, the maximum pip value, and one random seed value. That seed value will be used in two locations: first when creating the Random instance in the shuffle method in Domino Set and second when creating another Random instance to be used in the main method when determining what domino to steal. Only instantiate each Random instance once and then use that Random instance throughout their respective method.

EXPECTED OUTPUT:
The programmer will output each round, first by stating what is being sought for the box and then printing out each player's hand. If it is that particular player's turn, print the hand, for all other players, print out the number of dominoes in their hand. If the player can play a domino from their hand then print out the domino being played. If not, print out the domino the player stole and the player they stole it from and check to see if the domino can be played. If it can, print the domino the player played. Once there is only one player left in the game, proclaim that player the Domino Meanie and print the players hands as if it is the turn of the Domino Meanie.

EXPECTATIONS:
This program should incorporate the usage of Iterators and TreeSets. To do this, Domino must be made into a Comparable, and a compareTo method must written for it:

```
public class Domino implements Comparable<Domino>
```

The program will use standard input rather than read from a file. There are no files to read in this program.

The Domino class should have a toString method that will send back the data of the dominoes in the following manner: [6|2]. The class should have a method that could also rotate the domino values such that the toString could send back: [2|6]. When comparing two dominoes, the two largest pips should be compared to first, if they are the same, then return the comparison of the tow smallest pips. Math.max and Math.min will be helpful methods in this situation.

The DominoSet class should create the set of dominoes being used based on the maximum pip value. This means that a max pip value of six should create twenty eight dominoes in this order:
[0|0][0|1][0|2][0|3][0|4][0|5][0|6][1|1][1|2][1|3][1|4][1|5][1|6][2|2][2|3][2|4][2|5][2|6][3|3][3|4][3|5][3|6][4|4][4|5][4|6][5|5][5|6][6|6].

The DominoSet class will provide the programmer with a prewritten shuffle method. This is the only part of the code that is allowed to use indices:

```
public void shuffle(int randSeed)
{
      Random rand = new Random(randSeed);
      for(int i=0; i<setOfDominoes.size(); i++)
      {
            if(rand.nextBoolean())
                  setOfDominoes.get(i).rotate();
            int k = rand.nextInt(setOfDominoes.size());
            Domino temp = setOfDominoes.get(k);
            setOfDominoes.set(k,setOfDominoes.get(i));
            setOfDominoes.set(i, temp);
      }
}
```

The Domino Meanie class should have the following piece of code in the main method:

```
out.println("Enter in the random seed value:");
randSeed = input.nextInt();
randNum = new Random(randSeed);
```

This will create the randNum object that can be used to randomly choose a domino from each players hand.

```
SAMPLE RUN:
Enter in the number of players:
3
Enter in the max pip size:
3
Enter in the random seed value:
26
Looking for [0|0]
Player 1 hand: [[1|1], [2|1], [1|3], [3|3]]
Player 2 number of dominoes: 3
Player 3 number of dominoes: 3
```

```
player 1 grabbed [0|0] from player 2
[0|0] played by player 1
Looking for [0|1]
Player 1 number of dominoes: 4
Player 2 hand: [[0|1], [0|3]]
Player 3 number of dominoes: 3
[0|1] played by player 2
Looking for [0|2]
Player 1 number of dominoes: 4
Player 2 number of dominoes: 1
Player 3 hand: [[2|0], [2|2], [2|3]]
[0|2] played by player 3
Looking for [0|3]
Player 1 hand: [[1|1], [2|1], [1|3], [3|3]]
Player 2 number of dominoes: 1
Player 3 number of dominoes: 2
player 1 grabbed [0|3] from player 2
[0|3] played by player 1
Looking for [1|1]
Player 1 number of dominoes: 4
Player 2 number of dominoes: 0
Player 3 hand: [[2|2], [2|3]]
player 3 grabbed [1|3] from player 1
Looking for [1|1]
Player 1 hand: [[1|1], [2|1], [3|3]]
Player 2 number of dominoes: 0
Player 3 number of dominoes: 3
[1|1] played by player 1
Looking for [1|2]
Player 1 number of dominoes: 2
Player 2 number of dominoes: 0
Player 3 hand: [[2|2], [1|3], [2|3]]
player 3 grabbed [3|3] from player 1
Looking for [1|2]
Player 1 hand: [[2|1]]
Player 2 number of dominoes: 0
Player 3 number of dominoes: 4
[1|2] played by player 1
Player 3 is the Domino Meanie!
Player 1 number of dominoes: 0
Player 2 number of dominoes: 0
Player 3 hand: [[2|2], [1|3], [2|3], [3|3]]
```