Class Card

Method	Test Cases	Meaning
generateRandomUnusedNumber()		check for duplicate numbers. check for illegal values. ie numbers that are greater than 39 or less than 0
getPerson()		return the person
getGroup()		return the group
getPoints()		return the unmodified point value
toString()	8 if conditions	return "person" "group" "points" or "person" "group" "*" if the point value is special.

Class Guillotine

Method	Test Cases	Meaning
main(String[] args)	no args args less than 2, greater than 40 args is not int multiple indices for args	private static method. Tested by entering the following into the interactions pane: 1. run Guillotine 2. run Guillotine 1 3. run Guillotine 41 4. run Guillotine abc 5. run Guillotine a 3 6. run Guillotine 5 Cases 1 through 5 expects a cardList with 20 cards. Case 6 expects 5 cards.
calculatePoints()	0,1,many first middle last 5 if statements	The special cards are in the first, middle, or last position. There are 0, 1, or many special cards, each with 0, 1, or many "conditions" that change their point value. For example, a Heretic with 3 more church members will add 3 points to the total point value. There Heretic has a point value of 1, which is added later.
Tests on Buttons and TextAreas		When it was not the players turn, clicking the buttons did nothing. Clicking the buttons did the appropriate action: change the linked list and the display, and disabled the button if it was a one time use. If the button was Draw, the card was removed and added to the appropriate player display.

		When the game ends, clicking buttons will do nothing.
		The Score board displayed the correct point worth of each player. Displayed who's turn it was. And when the game was over, Declared a winner or a tie.

Class LinkedList

Method	Test Cases	Meaning
getFirstNode()		returns the first node
setFirstNode()		sets the first node
addToFront()	0, 1, many First Middle Last are not applicable	0 elements, 1 element, many elements in list
isEmpty()	0, 1, many	0 elements, 1 element, many elements in list
removeFromFront()	0, 1, many	0 elements, 1 element, many elements in list
length()	0, 1, many	0 elements, 1 element, many elements
moveBack(int n)	0, 1, many	move back 0, 1, many with 0 elements, 1 element, many elements in list
moveFirstToLast()	0, 1, many	move with 0 elements, 1 element, many elements in list
moveLastToFirst()	0, 1, many	move with 0 elements, 1 element, many elements in list
reverseList()	0, 1, many	reverse with 0 elements, 1 element, many elements in list
reverseFirstK(int k)	0, 1, many	reverse first 0, 1, many elements with 0 elements, 1 element, many elements in list
toString()	0, 1, many	0 elements, 1 element, many elements in list
contains(T element)	not used	
iterator()		returns a LinkedListIterator.

Class LinkedListIterator

Method	Test Cases	Meaning
hasNext()	0, 1, many	0 elements in the list, 1 element in the list, many elements in the list

next()	0, 1, many	0 elements in the list, 1 element in the list, many elements in the list

Class LLNode

Method	Test Cases	Meaning
getElement()		returns the element
getNext()		returns the next node
setElement()		sets the element
setNext()		sets the next node
lengthFromHere()	not used	
insertAfter(T element)	not used	
deleteNext()	not used	