Quiz navigation

i i 1 2 i 3
4 i 5 6 i 7
8 i 9 10 i 11
i 12 13 14 15 16
17 18

Show one page at a time

Finish review

Started on	Saturday, February 18, 2017, 8:30 PM
State	Finished
Completed on	Tuesday, February 21, 2017, 7:15 PM
Time taken	2 days 22 hours
Grade	**92.00** out of 100.00

Information

Flag question

The next set of questions list code and ask you to describe the contents of the chain headed by firstNode and the value of the node referred to be currentNode.

To write the contents of the chain, use the method of connecting the data with arrows, starting with the data contained in firstNode. For example: 4 -> 5 -> 8 -> 3

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node c = new Node(7, b);
Node firstNode = new Node(2, c);
Node currentNode = firstNode.next;
```

Question 1

Correct

2.00 points out of 2.00

Flag question

What is the contents of the chain headed by firstNode?

Answer: 2 -> 7 -> 3 -> 5 ✓

The correct answer is: 2->7->3->5

Question 2

Correct

2.00 points out of 2.00

Flag question

What is the data held in the node referred to by currentNode?

Answer: 7 ✓

The correct answer is: 7

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node c = new Node(7, b);
Node firstNode = new Node(2, c);
Node current = firstNode;
current.next.next.data = 8;
```

Question 3

Correct

What is the contents of the chain headed by firstNode?

Answer: 2 -> 7 -> 8 -> 5 ✓

2.00 points out of 2.00

Flag question

The correct answer is: 2->7->8->5

Question 4

Correct

2.00 points out of 2.00

Flag question

What is the data held in the node referred to by current?

Answer: 2

The correct answer is: 2

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node c = new Node(7, b);
Node firstNode = new Node(2, c);
Node currentNode = firstNode;
currentNode.data = 15;
```

Question 5

Correct

2.00 points out of 2.00

Flag question

What is the contents of the chain headed by firstNode?

Answer: 15 -> 7 -> 3 -> 5

The correct answer is: 15->7->3->5

Question 6

Correct

2.00 points out of 2.00

Flag question

What is the data held in the node referred to by currentNode?

Answer: 15

The correct answer is: 15

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node c = new Node(7, b);
Node firstNode = new Node(2, c);
Node currentNode = firstNode;
currentNode = currentNode.next;
```

Question 7

Correct

2.00 points out of 2.00

Flag question

What is the contents of the chain headed by firstNode?

Answer: 2 -> 7 -> 3 -> 5

The correct answer is: 2->7->3->5

Question 8

Correct

2.00 points out of 2.00

Flag question

What is the data held in the node referred to by currentNode?

Answer: 7

The correct answer is: 7

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node c = new Node(7, b);
Node firstNode = new Node(2, c);
Node currentNode = firstNode;
currentNode.next = currentNode.next.next;
```

Question 9

Correct

2.00 points out of 2.00

Flag question

What is the contents of the chain headed by firstNode?

Answer: 2 -> 3 -> 5 ✓

The correct answer is: 2->3->5

Question 10

Correct

2.00 points out of 2.00

Flag question

What is the data held in the node referred to by currentNode?

Answer: 2 ✓

The correct answer is: 2

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node firstNode = new Node(2,b);
Node currentNode = firstNode;
currentNode = currentNode.next;
currentNode = currentNode.next;
System.out.println(currentNode.next);
```

Question 11

Correct

2.00 points out of 2.00

Flag question

What is the result of the println?

Answer: null ✓

The correct answer is: null

Information

Flag question

```
Node a = new Node(5);
Node b = new Node(3, a);
Node firstNode = new Node(2,b);
Node currentNode = firstNode;
currentNode = currentNode.next;
currentNode = currentNode.next;
System.out.println(currentNode.next.data);
```

Question 12

Correct

2.00 points out of 2.00

What is the result of the println?

Answer: error ✓

Flag question

The correct answer is: error

Question 13

Complete

8.00 points out of 12.00

Flag question

Write a method to determine if a linked chain of Integers are all divisible by some number. The method header is:

```
public boolean isDivisibleBy(Node<Integer> firstNode, int divisor)
```

Examples:

- Invoking the method with 3 -> 9 -> 6-> 12 and 3 returns true.
- Invoking with 3 -> 9 -> 15 -> 8 -> 6 and 3 returns false.

```
	public static boolean isDivisibleBy(Node<Integer> firstNode, int divisor)
{
		Node<Integer> current = firstNode;
		boolean isDivisible = false;

		while(current != null){
			Integer currentInteger = current.data;
			if((currentInteger % divisor) == 0 && isDivisible==false){
				isDivisible = true;
			}else{
				isDivisible = false;
			}
			current = current.next;
		}
		return isDivisible;
	}
```

Comment:
good work!! you covered all the key mechanics

one worry is, that the method only returns the divisibility status of the very last element

-4 will return true or false based only on the last element in the chain; need to break out of the chain when you find an element that is not divisible

what happens after isDivisible is set to true, how does the IF part behave?

Question 14

Complete

12.00 points out of 12.00

Flag question

Write a method to print every other node in the chain, starting with the first node. The method header is:

```
public void printEveryOther(Node firstNode)
```

```
	public static void printEveryOther(Node firstNode) {
		Node current = firstNode;
		int counter = 1;

		while(current != null){
			if(counter % 2 == 0){ //if counter is even
				int uselessData = (int) current.data;
			}else{
				int currentData = (int) current.data;
				System.out.println(currentData);
			}
			current = current.next;
			counter++;
		}

	}
```

Comment:

consider how you could accomplish the task without a counter- but only using checks against null- this is the standard approach to working with nodes

Question 15

Complete

12.00 points out of 13.00

Flag question

Write a `getMin` method for the `LinkedBag` class. This method returns the smallest object in the bag.

- Assume the bag contains objects whose classes implement Comparable.
  - This means you can invoke compareTo on your objects of type T.
  - See the note on the opening page of the homework about how to change the LinkedBag code to reflect this.
- For full credit, do **not** invoke the toArray() method and do not use a separate ArrayList, LinkedList, etc. type object.

```
    public T getMin(){
            Node currentNode = firstNode;
            T minData = currentNode.data;

            while(currentNode != null){
                    if(minData.compareTo(currentNode.data) > 0){
                            minData = currentNode.data;
                    }
                    currentNode = currentNode.next;
            }

    return minData;
}
```

Comment:
-1 will crash on an empty bag because firstNode is null

Question 16

Complete

13.00 points out of 13.00

Flag question

Write a `union` method for the `LinkedBag` class.

- The *union* of two bags is the combined contents of both bags. Unions are explained in more detail in Chapter 1, #5.
- A union might contain duplicates.
- The method should **not** alter either bag. The current bag and the bag sent in as a parameter should be the same when the method ends.
- For full credit, do **not** invoke the toArray() method and do not use a separate ArrayList, LinkedList, etc. type object.
- The method header is:

```
public BagInterface<T> union(LinkedBag<T> anotherBag)
```

See Homework W04: Array-Based Bags and Lists #5 for examples of how unions work.

```
    public BagInterface<T> union(LinkedBag<T> anotherBag){
            BagInterface<T> result = new LinkedBag <T>();

            Node myBag = this.firstNode;
            Node otherBag = anotherBag.firstNode;

            while(myBag != null){
                    result.add(myBag.data);
                    myBag = myBag.next;
            }
            while(otherBag != null){
                    result.add(otherBag.data);
                    otherBag = otherBag.next;
            }

            return result;

    }
```

Comment:

Question 17

Complete

10.00 points out of 13.00

Flag question

Write an `addAll` method for the `LList` class.

- The method adds an array of items to the end of the current list.
- The method header is: `public void addAll(T[] items)`
- For full credit, do **not** invoke the toArray() method and do not use a separate ArrayList, LinkedList, etc. type object.

```
public void addAll(T[] items){
        T[] arrayItems = items;

        for(T element : arrayItems){
                this.add(element);
        }

}
```

Comment:
-3 repeated calls to add within a loop is very inefficient- take advantage of access to the node structure to only link through the chain one time

Question 18

Complete

13.00 points out of 13.00

Flag question

Write a `getPosition` method for the `LList` class.

- The method returns the position of a given object in the list.
- The method header is: `public int getPosition(T anObject)`
- For full credit, do **not** invoke the toArray() method and do not use a separate ArrayList, LinkedList, etc. type object.

```
//The method returns the position of a given object in the list.
 public int getPosition(T anObject){
         Node currentNode = firstNode;
         boolean found = false;
         int position = 0;

        while(currentNode != null && !found){
                position++; //increase position for every while loop
                T currentObject = currentNode.data;
                currentNode = currentNode.next;

                if(currentObject.equals(anObject)){ //equals an object
                        found = true; //return true if the object is found
                        return position;
                }
        }
         return -1; //return -1 if not found
 }//end of getPosition
```

Comment:

Finish review

You are logged in as Faustina Nyaung (Log out)

CS111C-META-SPRING-2017