CS116 Midterm Exam

Boris Glavic

May 10, 2019

1 Exam

1.1 Instructions

Download `ListStuff.java` from blackboard (there is an assignment Midterm). The exam consists of two parts. In the first part you are implementing methods in a given class. In the second part you are writing a new class. After you are done upload both `.java` files to Blackboard.

1.2 List Operations - Implement Methods in a Class (13 points)

In this task, you will extend the given class `ListStuff.java` which contains several functions whose body is missing. Implement these functions according to the specifications shown below. The `main` method of this class contains a few test cases for these methods.

1.2.1 Count Occurrences (4 points)

Write a function `countOccurences` that takes as input a list of integers `l` and a number `searchEl` and counts how many times the number appears in the list. For example, for a list `(1,1,4,1,2)` and search element `1` you should return `3` (the number `1` appears `3` times in this list).

1.2.2 Find the largest element (4 points)

Write a function `max` that takes as input a list of integers and returns the largest element of the list. For example, the largest elements of `(4,50,3,2)` is `50`.

1.2.3 Check whether a list is sorted (5 points)

Write a function `isInOrder` that takes as input a list of integers and returns `true` (`boolean`) if the list is sorted in increasing order. For example, `(1,2,5,10,14)` is sorted increasingly, but `(1,4, 6, 2, 9)` is not since `2` should appear before `4`.

1.3 Write a new Class (12 points)

1.3.1 Design the fields (4 points)

Write a class `Student`. Each object of this class stores the following information about a student:

- First Name
- Last Name
- Credit hours taken

1.3.2 Constructor (2 points)

The constructor of the class should take as input all information about a student as described above.

1.3.3 Getter and Setter Methods (3 points)

Add getter and setter methods to access the information about a student. For example, there should be a method `getFirstName`.

1.3.4 Additional Methods (3 points)

Write a method that returns the full name of a student which constists of the first name followed by a space followed by the last name.

1.3.5 Bonus (3 points)

Also store the **GPA** of the student and implement a method `registerGrade(int credits, double grade)` that updates the students GPA and credit hours taken based on a `grade` the student got in a course with a certain number of `credits`.

2 Solutions

```java
/**
 *
 */
package midterm;

import java.util.ArrayList;
import java.util.List;

/**
 * @author lord_pretzel
 *
 */
public class ListStuff {

	private static final List<Integer> l1 = new ArrayList<Integer>();
	private static final List<Integer> l2 = new ArrayList<Integer>();

	static {
		int[] l1a = {1,2,3,4,4,4,5,6,7};
		int[] l2a = {1,4,-1,1,3,4};

		for(int i = 0; i < l1a.length; i++)
			l1.add(l1a[i]);

		for(int i = 0; i < l2a.length; i++)
			l2.add(l2a[i]);
	}


	/**
	 * Count the number of occurrences of {@code searchEl} in list {@code
↪  l}.
	 *
	 * @param l input list
	 * @param searchEl the element to search for
	 * @return the number of occurrences of {@code searchEl}
	 */
	public static int countOccurances(List<Integer> l, int searchEl) {
		int count = 0;

		for(Integer i: l) {
			if(i == searchEl)
				count++;
```

```
                }

                return count;
        }

        /**
         * Given a list {@code l} return true if this list is in order. A list
↪   is in order if the elements of the list are sorted in increasing order.
         * For example, (1,2,5,10,14) is sorted increasingly, but (1,4, 6, 2, 9)
↪   is not since 2 should appear before 4.
         *
         * @param l
         * @return true if the list is sorted in increasing order
         */
        public static boolean isInOrder(List<Integer> l) {
                for(int i = 0; i < l.size() - 1; i++) {
                        if(l.get(i) > l.get(i+1)) // found element that is out
                        ↪  of order
                                return false;
                }
                return true;
        }

        /**
         * Return the greatest element from list {@code l}.
         *
         * @param l the input list
         * @return the greatest element.
         */
        public static int max(List<Integer> l) {
                int max = l.get(0);

                for(Integer i: l)
                        if (i > max)
                                max = i;
                return max;
        }

        public static void main (String[] args) {
                System.out.println("l1 contains this many occurances of 4: " +
                ↪  countOccurances(l1, 4));
                System.out.println("l2 contains this many occurances of 1: " +
                ↪  countOccurances(l2, 1));
                System.out.println("l1 is sorted: " + isInOrder(l1));
                System.out.println("l2 is sorted: " + isInOrder(l2));
```

```
                System.out.println("l1 max is:" + max(l1));
                System.out.println("l2 max is:" + max(l2));
        }


}
```

```java
/**
 *
 */
package midterm;

/**
 * @author lord_pretzel
 *
 */
public class Student {

	private String firstName;
	private String lastName;
	private int creditsTaken;
	private double gpa;

	public Student(String firstName, String lastName, int creditsTaken,
	↪  double gpa) {
		this.setFirstName(firstName);
		this.setLastName(lastName);
		this.setCreditsTaken(creditsTaken);
		this.setGpa(gpa);
	}

	public String getFullName() {
		return firstName + " " + lastName;
	}

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public int getCreditsTaken() {
```

```
        return creditsTaken;
    }

    public void setCreditsTaken(int creditsTaken) {
        this.creditsTaken = creditsTaken;
    }

    public double getGpa() {
        return gpa;
    }

    public void setGpa(double gpa) {
        this.gpa = gpa;
    }

    public void registerGrade(int credits, double grade) {
        gpa = (gpa * creditsTaken) + (credits * grade);
        creditsTaken += credits;
        gpa /= creditsTaken;
    }

}
```