Take Assessment: Practical Quiz 7

Please answer the following question(s).
If the assessment includes multiple-choice questions, click the "Submit Answers" button when you have completed those questions.

You have 120 minutes to take this assessment.
Please complete this assessment by Thu Dec 20 2007 22:18:35 GMT+0800.

1. Go to bottom of question.

Using Java Collections

Prerequisites, Goals, and Outcomes

Prerequisites: Before you begin this exercise, you need:

- *Java collections*
  - Knowledge of class ArrayList
    - Declaring array lists
    - Initializing array lists
    - Accessing array list elements
    - Adding elements to an array list
    - Removing elements from an array list

- *Iterators and for-each loop*:
  - Know how to traverse and manipulate a collection using iterators and for-each loops.

Goals: Reinforce your ability to use class ArrayList, iterators and for-each loops.

Outcomes: You will demonstrate mastery of the following:

- Using Java collections
  - Creating a collection:
    - From a set of objects
    - From an array

  - Processing collections
    - Finding an element with a specific

characteristic
        - Counting the number of elements with a specific characteristic
        - Analyzing the contents of a collection
    - Removing elements from a collection:
        - Removing all objects with a specific characteristic

    - Displaying the contents of a collection
        - Obtaining the string representation of a collection

Background

This assignment asks you to implement a set of methods to process Java collections.

Description

In this assignment, you will finish the implementation of StudentArrayList, a class with methods for processing an *ArrayList* of Student objects. iCarnegie provides a test driver and the class Student.

Class Student

A complete implementation of this class is included in the student archive *student-files.zip*. Stop *now* and review its documentation:

- *Student.html*. Documentation of class Student

Class StudentArrayList

A partial implementation of this class is included in the student archive *student-files.zip*. You should complete the implementation of each method.

Class TestStudentArrayList

This class is a test driver for class StudentArrayList. It contains test cases for each method in the class. A complete implementation is included in the student archive *student-files.zip*. You should use this

class to test your implementation of class StudentArrayList.

Files

The following files are needed to complete this assignment:

- *student-files.zip*. Download this file. This archive contains the following:
  - *Student.java*. A complete implementation
  - *StudentArrayList.java*. Use this template to complete your implementation.
  - *TestStudentArrayList.java*. A complete implementation

Tasks

Implement all methods in class StudentArrayList. Follow Sun's code conventions. The following steps will guide you through this assignment. Work incrementally and test each increment. Save often.

1. **Extract** the files by issuing the following command at the command prompt:

   C:\>unzip student-files.zip

2. **Test** each method as soon as you finish writing it by issuing the following command at the command prompt:

   C:\>java TestStudentArrayList

3. **Implement** the method makeArrayList:

- *public static ArrayList<Student> makeArrayList(Student first, Student second, Student third)*. This method takes three Student objects and returns an array list with three elements. The first element of the array list contains a reference to the first argument; the second element contains a reference to the second argument; and the third element contains a reference to the third argument.

For example, consider the following three Student objects:

- Student[328,Galileo Galilei,80]
- Student[123,Albert Einstein,100]

- Student[96, Isaac Newton, 90]

If these objects are passed to makeArrayList in the indicated order, makeArrayList will return an array list with three Student objects:

```
ArrayList{Student[328, Galileo Galilei, 80],
          Student[123, Albert Einstein, 100],
          Student[96, Isaac Newton, 90]}
```

Note: Student[*ID*, *name*, *grade*] is a representation of a Student object.

4. **Implement** the method makeArrayListFromArray:

- *public static ArrayList<Student> makeArrayListFromArray(Student[] array)*. This method takes an array of Student objects and returns an array list with the same elements in the same order.

For example, consider the following array with three Student objects:

```
{Student[328, Galileo Galilei, 80],
 Student[123, Albert Einstein, 100],
 Student[96, Isaac Newton, 90]}
```

If makeArrayListFromArray is passed this array, it will return an array list with three Student objects:

```
ArrayList{Student[328, Galileo Galilei, 80],
          Student[123, Albert Einstein, 100],
          Student[96, Isaac Newton, 90]}
```

5. **Implement** the method hasStudent. Use a for-each loop to implement this method.

- *public static boolean hasStudent(ArrayList<Student> arrayList, int id)*. This method takes an array list of Student objects and a student ID, returns true if the specified array list contains a student whose ID matches the specified ID.

For example, consider the following array list:

```
ArrayList{Student[328, Galileo Galilei, 80],
          Student[123, Albert Einstein, 100],
          Student[96, Isaac Newton, 90]}
```

If hasStudent is passed this array list and the id 123, it will return true.

If hasStudent is passed this array list and the id 321, it will return false.

6. **Implement** the method countGradeGreaterOrEqual. Use a for-each loop to implement this method.

- *public static int countGradeGreaterOrEqual(ArrayList<Student> arrayList, int grade)*. This method takes an array list of Student objects and a grade, and returns the number of students in the specified array list whose grade is greater than or equal to the specified grade.

For example, consider the following array list:

```
ArrayList{Student[328,Galileo Galilei,80],
          Student[123,Albert Einstein,100],
          Student[96,Isaac Newton,90]}
```

If countGradeGreaterOEqual is passed this array list and the grade 100, it will return 1.

If countGradeGreaterOEqual is passed this array list and the grade 70, it will return 3.

7. **Implement** the method getMinGrade. Use an iterator to implement this method.

- *public static int getMinGrade(ArrayList<Student> arrayList)*. This method takes an array list of Student objects and returns the smallest grade of the students in the specified array list. To simplify the code, you may assume that the array list is not empty.

For example, consider the following array list:

```
ArrayList{Student[328,Galileo Galilei,80],
          Student[123,Albert Einstein,100],
          Student[96,Isaac Newton,90]}
```

If getMinGrade is passed this array list, it will return 80.

8. **Implement** the method getGradeAverage. Use a for-each loop to implement this method.

- *public static double getGradeAverage(ArrayList<Student> arrayList)*. This method takes an array list of Student objects and returns the average grade of the students in the specified array list.

For example, consider the following array list:

```
ArrayList{Student[328,Galileo Galilei,80],
          Student[123,Albert Einstein,100],
          Student[96,Isaac Newton,90]}
```

If getGradeAverage is passed this array list, it will return 90.0.

9. **Implement** the method removeGradeLess. Use an iterator to implement this method.

- *public static void removeGradeLess(ArrayList<Student> arrayList, int grade)*. This method takes an array list of Student objects and a grade, and removes all students in the specified array list whose grade is less than the specified grade.

For example, consider the following array list:

```
ArrayList{Student[328,Galileo Galilei,80],
          Student[123,Albert Einstein,100],
          Student[96,Isaac Newton,90]}
```

If removeGradeLess is passed this array list and the grade 100, the array list will be modified as follows:

```
ArrayList{Student[123,Albert Einstein,100]}
```

10. **Implement** the method displayAll. Use an iterator to implement this method.

- *public static String displayAll(ArrayList<Student> arrayList)*. This method takes an array list of Student objects and returns a string representation of the specified array list. To simplify the code, you may assume that every element in the specified array list contains a valid reference to a Student object.

Use the method toString in the class Student to obtain the string

representation of each object in the array list. A new line character (\n) should separate the string representations of the objects.

For example, consider the following array list:

```
ArrayList{Student[328,Galileo Galilei,80],
          Student[123,Albert Einstein,100]}
```

If this array list is passed to displayAll, the following string will be returned:

```
"Student[328,Galileo Galilei,80]\nStudent[123,Albert Einstein,100]"
```

Note that the string does *not* end with a new line character (\n).

Submission

Upon completion, submit **only** the following:

1. StudentArrayList.java

Go to top of question.

File to submit: Upload File | Forward File | Refresh | Ready for Grading

---

Go to top of assessment.