COSC 190
Intermediate Programming

Simulation Test

Feb 09, 2024

Question 1

You have been provided with an Implementation of a generic Stack class. You must complete the following two method stubs in the generic Stack class.

```
public List<T> fetchMatching(Predicate<T> predicate)
```

This method will take in a supplied predicate and return all elements passing the predicate as a List. *[Feel free to add helper methods if needed]*

```
public ArrayList<T> convert(Function<T,T> function, Predicate <T>
                            predicate)
```

This method will generate a new array list composed of elements that (a) pass the given predicate test and (b) have had the transformation applied to them. For example, if I have a stack (lstSample) composed of Integers and called transform using the following syntax:

lstSample.transform(x -> x * 3, x -> x %2 == 1)

I would get back a array list composed of the cube of all odd numbers from the stack. *[Feel free to add helper methods if needed]*

Question 2

In Q2 you will find the following method stub:

```
public static LinkedList<File> fetchDifferentJavaFiles (String
                               sDirectoryPath)
```

This method is intended to be a ***recursive*** Java method for going through the given directory and related sub-directories and finding all those files which have a *java* suffix. This method will return a LinkedList of File objects relating to those files and there should not be any repeated file. *[Feel free to add helper methods if needed]*

COSC 190
Intermediate Programming

Simulation Test

Feb 09, 2024

Question 3

The CSV file **Trains.csv** provided to you lists information about various Trains. The csv file is broken up in the following fields:

- Train Type (String)
- Train Name (String)
- Capacity (Integer)
- Country (String)

This csv file does not need to be changed and should be copied to the ***files*** folder on your system.

Also provided to you is a Train class that could be used for storing information related to the CSV File. The existing code in this class may not be modified. Note that the constructor for this class is expecting a String array to be given to it.

Provided to you in Q3 are a series of method stubs that you will be expected to complete. ***Other than the LoadList, your methods must utilize a Stream-based approach to complete the requirements***.

The method signatures are listed as follows:

```
public static List<Train> loadList(String sFileName)
```

This method should return a List of Train objects. It is assumed that the given File name will reference the Train.csv file.

```
public static int countOfType(List<Train> lstTrain, String sType)
```

This returns a count of all Trains on the list that are of the indicated type.

```
public static List<Train> getSortedByCapacityCountry(List<Train>lstTrain,
                                String sCountry)
```

This method will return a sorted List (by descending capacity i.e. displacement) of all the Trains in the given country.

```
public static List<String> getTypes(List<Train> lstPorts)
```

This method will return a list of all the unique Train types from the list of Trains.

```
public static String [] getHighestByType(List <Train> lstPorts)
```

This method will return an array of Strings. Each entry in the array will consist of a colon separate list of values that consists of (i) The Train type, (2) The name of the Train with the maximum capacity for that type and (3) the capacity.

COSC 190
Intermediate Programming

Simulation Test

Feb 09, 2024

Question 4

Modify the provided Train Class file to allow records of Train to be written out to a Random-Access File. Specifically, you should implement the methods with the following signatures:

```
public void writeToRAF(RandomAccessFile obFile)
public void readFromRAF(RandomAccessFile obFile)
```

The members of the Train should be written to the random-access file in the order of capacity, country, type, and name of the train.

Remember that when writing out Random Access records, all the records should be the same size.