Lab week 10 OO Programming TU857

This lab will cover two things

1) A java feature called Generics – 0.6
2) A start to your assignment - 0.4

Generics in Java are a feature that allows you to write code that works with any data type They were introduced in Java 5 as part of the Java Collections Framework update.

Part 1 Create a class that manages a list of Strings 0.3

Create a class called **StringItemManager;** The purpose of the class is to simply manage a list of items of type String (e.g. list of Student names, list of country names).

Give it one attribute of **type List**.

Add a **constructor t**hat instantiates your List attribute to be an ArrayList

Implement three methods:

- **addItem –** which takes in a String and adds it to the list;
- **getAllItems** which returns all Strings in the list;
- **getItemAt** – which returns the String at the index (int) you send in to the method. This method should validate that a valid index value has been sent in (that doesn't exceed the size of the list) .

Control class : Then - as usual, set up a separate control class with a main method to use this class.

In the main method,

Instantiate your StringItemManager class.
Add 5 strings using your AddItem method.
Get all items in the list and print them all out/
Retrieve the items at a particular index :

A valid index (e.g. 3);
An invalid index (i.e. -1, 10,000)

Try using the code to add an **integer (**i.e. without the " ") to the list .

What happens?

Part 2 Create a class that manages a list of Integers 0.4

Write a class called **IntegerItemManager**.

It should have the same capability as the StringItemManager class you just wrote – but this time, you're storing integers (fastest thing is just make a copy your string class from part 1 and hack it to work for integers – just takes a few minutes).

Expand your **control class** from Part 1 so that it

- instantiates the IntegerItemManager class, adds 5 integers, and retrieves all items list to print them all out.

Part 3 Develop a generic class 0.6

Parts 1 and 2 illustrate how two different classes are needed for two data types. If you also needed classes to manage lists of objects of type Double, Student, Person, Animal etc.. you'd need more and more of the "same" class repeated again and again which doesn't make sense – and scores really badly on code reuse.

For part 3, use Generics to make one *generic* class called **ItemManager** that works for all data types It should work for storing any data type – and have the functionality already described (add, getall, getAtIndex, an attribute of List and a contructor to set up the ArrayList).

Then, expand your control class from parts 1 & 2 to:

Instantiate your ItemManager class to store Strings.

Add some strings to it. Use the methods.

Now, create another instance of ItemManager to store Integers.

Add integers to it etc.

This class will work for any object type you want to deal with – and you've only written one class.

Part 4 Your assignment – getting started

In Brightspace, find the theme that was assigned to you for your datasets/ application .

Generate the dataset (200 rows etc) following the specification in the assignment.

Produce the **frequency table**. Note: Just do this through excel or however you choose to sort/ count - there is no java coding involved for Part 4 of this lab. The frequency table, as explained in "a peek at machine learning" lecture allows you to see the total number of times each permutation of features occurs for per label value (i.e. Per "class", using machine learning terminology).

Keep a copy of this table, as you'll need to submit this table with your assignment.