Classes and Interfaces (Money, Etc. Revisited)

Re-submit Assignment

Due Nov 25 by 11:59pm **Points** 42 **Submitting** a file upload **File Types** zip

Description

In this assignment, you will update the Money, Date, Bill, and ArrayList classes from previous homeworks using new ideas that we have discussed in class. Write JUnit test to check the functionality.

1. Fix any privacy (and other) errors that were noted in your comments for the previous iteration of this homework.
2. Modify the Money, Date, and Bill class to implement the Comparable interface. Remember that compareTo takes an Object parameter and you should check to make sure that the object class is actually the correct class for the comparison, as appropriate. The Java spec says that it should throw a ClassCastException if the objects are not of the same type.
3. Modify the Money, Date, and Bill classes to implement the Cloneable interface. Note that Money and Date can simply copy their private instance variables, since they store only primitive and immutable types. However, you will need to override the clone method, to make it public, since it is protected in the Object class. The Bill class will need to do more, since it incorporates the Money and Date classes, which are mutable. Note that it can (and should) use the clone methods of those classes. Be sure to remove any use of the copy constructor for Money, Date, and Bill in the rest of the code (the definition can exist, but don't use it in other classes; use the clone method instead).
4. Build a class ExpenseAccount that extends your ArrayList. Also implement Cloneable interface. You should remove the limit on the number of bills that can be placed in an account by making your ArrayList dynamically resize itself.
5. Modify Money, Bill, and Date to implement the Serializable interface.

Hints, Etc.

- Remember to follow the Coding Style Guidelines.
- **Make sure you comment all methods and the class with javadoc comments.** This includes constructors, getters, setters, etc. If you have files based on code someone else has written (e.g., is based on a skeleton), you have to javadoc comment the methods someone else wrote also.
- The description for this assignment is much briefer than in some previous assignments. In particular, not much detail is given regarding what methods to have, etc. It is purposely open-ended. By this point, you should be able to design an appropriate interface to your classes as well as create an appropriate driver filled with tests. My advice is not to scrimp on the interface; include all methods one would reasonably expect objects of that class would be able to do.

About This Document

Original assignment by Rob Nash, Autumn 2014. Minor edits and additions by Johnny Lin, February 2015.

Interfaces Rubric

Criteria	Ratings			Pts
Money is Comparable	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Bill is Comparable	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Date is Comparable	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Money is Cloneable	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Bill is Cloneable	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Date is Cloneable	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
ExpenseAcount subclass	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Serializable Interface for Money, Bill, & Date	**2.0 pts** **Full Marks**		**0.0 pts** **No Marks**	2.0 pts
Clear, Well-Written, and Complete Comments in Code	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Program Compiles and Properly Runs	**4.0 pts** **Full Marks**	**2.0 pts** **Partial Marks**	**0.0 pts** **No Marks**	4.0 pts
Extra Credit Up to 10 points.	**0.0 pts** **Full Marks**		**0.0 pts** **No Marks**	0.0 pts

Criteria	Ratings			Pts
JUnit test cases	**4.0 pts** **Full Marks**	**2.0 pts** **Partial marks**	**0.0 pts** **No Marks**	4.0 pts
				Total Points: 42.0