COSC2391/2401 Software Architecture: Design and Implementation

TuteLab5 - JCF & Generics

1. What is the advantage of using the Java *Collection* or *Map* classes (as opposed to building your own)?

2. What are the main interfaces in Java Collection Framework (JCF)?

3. Does the sub-interface `Set` introduce any additional methods to `Collection` interface? Does it introduce any additional constraints?

4. Examine the class/interface structure of the `java.util.ArrayList<E>` class in the Java API documentation and explain how abstract classes and interfaces have been used to facilitate reusability and extensibility. What about generics, how and why are they used?

5. When is it more suitable to use a `LinkedList` over an `ArrayList`?

6. Which collection(s) will be suitable for:

(i) List of Goods received. New entries to be added at the end reflecting the date and time of goods receipt.

(ii) A text editor storing each line as an element in a list. Users are allowed to add, delete or alter any part of the document.

7. Why does the `Vector` class provide methods such as `addElement(Object o)` in addition to `add(Object o)` defined in the interface Collection?

8. What is the difference between **Map** and **Collection** classes. What are the constraints imposed by the `Map` interface?

9. What will be the output of the program below?

```
public class MapTest
{
 @SuppressWarnings({ "rawtypes", "unchecked" })
 public static void main(String[] args)
 {
       // Hash map that maps employee number to accounts
       Map accountsMap = new HashMap();
       accountsMap.put("E123", "Charles");
       accountsMap.put("E174", "Matt");
       accountsMap.put("E542", "Brendan");
       accountsMap.put("E174", "Tobin");
       System.out.println(accountsMap);
 }
}
```

10. Given `Book` is a subclass of `GenericMedia` which of the following will result in an error?

```
Library<GenericMedia> myMedias = new Library<GenericMedia>();
Library<Book> myBooks = new Library<Book>();
myBooks.add(new GenericMedia());
myBooks.add(new Book());
myMedias.add(new GenericMedia());
myMedias.add(new Book());
```

11. Given `Book` is a subclass of `AbstractMedia` which implements `Media` interface, which of the following will result in an error ?

```
Library<Media> myMedias = new Library<Media>();
Library<Book> myBooks = new Library<Book>();
myBooks.add(new Media());
myBooks.add(new Book());
myMedias.add(new Media());
myMedias.add(new Book());
```

12. Given the declarations below:

```
Library<Object> myObjects = new Library<Object>();
Library<Media> myMedias = new Library<Media>();
Library<Book> myBooks = new Library<Book>();
```

Which of the following will result in an error ?

```
Library<Object> my1 = (Library<Object>) myMedias;
Library<Book> my2 = (Library<Book>) myObjects;
Library<Media> my3 = (Library<Media>) myBooks;
```

13. Test the output of the code below .. and explain!

```
class Gen<E>
{
 private static int n = 1;

 public Gen()
 {
       n++;
 }

 public void print()
 {
       System.out.println("n is now " + n);
 }
}
```

```
public class StaticGenerics
{
  @SuppressWarnings("unused")
  public static void main(String[] args)
  {
        Gen<Object> myObjects = new Gen<Object>();
        Gen<String> myStrings = new Gen<String>();
        Gen<Integer> myInts = new Gen<Integer>();
        myInts.print();
  }
}
```

LAB EXERCISE

1. Code and test the examples from this tutorial and experiment further
2. If necessary complete your Aircraft Fleet Exercise from the previous tutelab exercise
3. Update the Fleet Solution to use the JCF (using parameterised collections)
4. Continue to work on your assignment