BITS, Pilani – KK Birla Goa Campus
Department of CS&IS

CS/IS F213 Object Oriented Programming
CS/IS C313 Object Oriented Programming and Design

Lab-7 *Dt.08.10.2014*

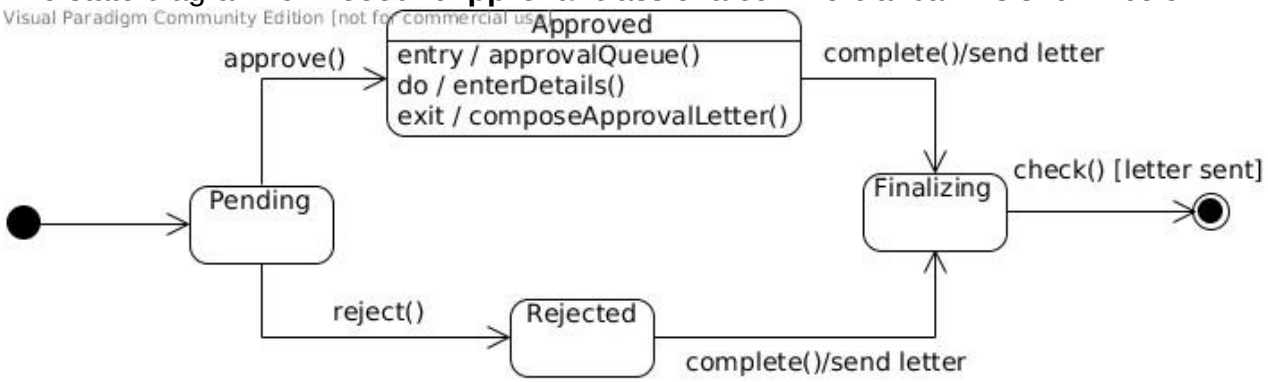
The pre-lab work for the lab is as follows.

1. Read **Ch 8: State Diagrams** from *UML Distilled, Martin Fowler with Kendall Scott, Pearson Education, 2nd Edition, 2000*.
   Refer to ***L17_state_machines.zip*** folder for coding conventions on state machines.

2. Read **Ch4: Interfaces, Ch5: Nested Classes and Interfaces** from *The Java Programming Language, Ken Arnold, James Gosling and David Holmes, 4th Edition / 3rd Edition*.

The in-lab work for seventh lab is as follows.

Question-1
The state diagram for **AccountApproval** class of a commercial bank is shown below.



(ref: Fig. 14-12, Learning UML 2.0, Russ Miles and Kim Hamilton, O'Reilly, 2006)

The *Pending, Rejected* and *Finalizing* states do not have any entry or exit activities. The in-state activities of each of the above mentioned three states should have the following statement.

```
System.out.println("state: " + state.toString() + "; phase: " + phase.toString());
```

The *Approved* state has activities for all three phases of the state. Each of these activities can print the following line.

```
System.out.println("state: " + state.toString() + "; phase: " + phase.toString());
```

The variable sendLetter is to be set by the class, where as letterSent variable is set by the users of this class.

Write Java code to implement the state diagram. Use the AccountApprovalTest class available on the course site to test your class. You can cross check your output with the sample output provided on the course site.

The post-lab work for seventh lab is given below. (to be check in next lab)

Question-2

a) Define a class **Department** (representing a department in university) with private attributes for Department ID, Department Name and HOD Name. Give appropriate get and set methods for the attributes mentioned above.

b) Now define an inner class **Course** within Department with private attributes CourseID, Course Name and Faculty (taught by). The convention to be followed for course ID is that it should begin with department ID, followed by an underscore ('_') and ending in an unique integer number.

c) Add methods to department class to add and remove courses from it and print all courses of a department. (You may assume a maximum number of courses allowed for a department if you wish so).

Question-3

Write an interface *Searchable* which defines a method *search* to search on Strings. Now implement this interface in department class defined in previous class. The method search should take any one of the course ID or course name or faculty name as an input parameter and search for corresponding entries within the department and print the course details of that course. If no course is found matching the input then print "NO COURSES FOUND" as output.

References:

Tutorials onInterfaces
http://tutorials.jenkov.com/java/interfaces.html
http://www.tutorialspoint.com/java/java_interfaces.htm

Interfaces on Nested Classes
http://www-plan.cs.colorado.edu/diwan/3308-05/book/TIJ310.htm