Ex/IT/T/221/2017

B. E INFO. TECH2ND YEAR2ND SEMESTER EXAMINATION, 2017

(2ndSemester)

OBJECT ORIENTED SYSTEMS

Time: Three hours Full Marks: 100

Attempt **ALL**questions from GROUP – A and any **TWO**from Group – B

GROUP – A

1. Define three classes *Queue*, *Producer* and *Consumer*. The *Producer* thread puts a message (a *String* object which it gets by calling a method *getMesssage()*) into the queue and the *Consumer* thread displays the message. The last message that the *Producer* adds is "Exit" after which it terminates. The *Consumer* also terminates after displaying the message "Exit". To facilitate the tasks of the threads, the *Queue* class implements two methods: *add* and *remove. Add* method does not allow to insert messages in a full queue. *Remove* method does not allow the consumer to read from an empty queue. The add and remove methods of the queue class must be thread safe. The method *getMesssage()*need not be defined by you.

   Also, define a *main* method to create the *Producer*, *Consumer* and the *Queue* objects and start the threads. The main method must wait till the Producer and the Consumer threads complete their tasks.

   8+4+4+4

2. A *Document* class contains a *TextBox*. A *TextBox*object contains a *String* and three methods *cut*, *copy* and *paste*. You assume that the methods for *cut*, *copy* and *paste* inside the *TextBox* class are already defined. Recognize the *Document* class as the *Receiver* class of the Command pattern. Complete the implementation of the *Document*class.
   Implement necessary*Command* classes so that a command can be undone.
   Implement an *Invoker* class that has a stack of commands and implement two methods *ExecuteCommand* (Command x) and *undoCommand*() in that class.

   An example client is provided in the following for your understanding.

```
public class Example {
    privateInvokerinvoker = new Invoker();
    private Document document;
    publicstatic void main( ... ) {
            document = new Document();
    }
    private void cutMenuItem_Click(Event e){
            invoker.executeCommand(new cutCommand(document, e));
    }
    private void pasteMenuItem_Click(Event e)    {
            commandManager.ExecuteCommand(new pasteCommand(document, e));
    }
}
```

a) Draw the class diagram of the command pattern for this solution.

b) Define the classes as mentioned in the question in Java

6+14

3. Suppose that a company sells its products online through a website. Customers create orders and submit them. The company processes these orders and ships them to the customer.

The following requirements were found out.

- Customers can save an order and come back later (like many shopping carts) to complete it.
- When a customer submits an order, the company charges the customer (credit card, debit card, etc.) through a different billing service provider.
- The order is not shipped until the charge has been made successfully.
- Customers can track their orders by a tracking number.
- The company wants to know how many orders have been saved but not submitted, how many orders have been submitted but not shipped, how many orders are being processed, and how many have been delivered. You may assume that there is a database in which relevant information is stored for this processing.

a) Draw the use case diagram. Do not write details of any use case.

b) Draw the statechart diagram of the "Customer Order" object.

5 + 15

GROUP – B

4. Answer the following questions,
   a) What is the base class of all errors and exceptions in Java? What is the base class of all user defined exceptions in Java? Name 2 methods of this class.
   b) Define a *List*class that has a number of integers stored in an array. Implement the following methods.
      i. Suitable constructors.
      ii. clone
      iii. equals

   c) Which methods in a JAVA base class are not overridden?
   d) Which combination of try-catch-finally is syntactically correct if there is one try block?

(1+1+2)+(3+5+4)+2+2

5. Answer the following questions:

a) Define the following classes.

DefineRectangle class that contains the following.

i) Two instance variables width (double) and length (double).

ii) Three constructors. The no-arg constructor initializes the width and length to 1.0.

iii) Set Methods for all instance variables.

iv) Methods getArea() and getPerimeter().

v) Override the toString() method to return "A Rectangle with width=<its width> and length=<its length >"

Define a class called *Square*, as a subclass of *Rectangle*. It has no instance variable, but inherits the instance variables width and length from its superclass Rectangle. Override all necessary methods.

b) What is the difference between a String and a StringBuffer object?

(8+10)+2

6. Answer the following questions

a) Name two relationships between use cases. Give one example of each.

b) Give an example of an association class.

c) Consider the 4-queens problem where 4 queens are to be placed in a 4x4 chessboard so that no two queens may attack each other. Assume that the queen q_i is placed in the i-th column. The system must find out the rows in which each queen has to be placed. Draw the sequence diagram when an actor tries to place q_3 on the chessboard. Please note that before the actor tries to place $q_3$on the chessboard, q_1, q_2 have already been placed in the chessboard so that they do not attack each other. Once q_3 can be placed properly, a message will be sent to q_4 for its placement. In the Sequence Diagram, show all messages among all queen objects.

(2+2)+2+14

7. Answer the following questions.

a) A library has a number of subscribers and a catalogue of borrow-able items. There may be a number of copies of each borrowable item. Each copy of a book has a unique accession number. A borrow-able item may be a book or a CD or a micro-film etc. There are two types of subscribers in the system. One would be ordinary and the other one would be privileged. Each Subscriber has a number of tickets. A subscriber can borrow a book for a ticket. Ordinary subscribers have 5 tickets while privileged subscribers have 10 tickets. A ticket stores the return date when a subscriber borrows an item. Each subscriber has a unique number, name, address, the list of items borrowed etc. A catalogue contains a list of borrow-able items. Any borrow-able item has and a name. A book also has name(s) of the author(s) and ISBN numbers.
Draw a class diagram for the above description.

b) Show the structure of a *Visitor* Pattern using a class diagram and a sequence diagram

14+(3+3)