Exercises

UML Diagrams

1. Specify a **use case diagram** to be included in a requirements analysis diagram for the application described below. Give also a detailed description of one of the most relevant use cases for this application.

   We want to develop an application to manage table reservations and client arrival/departure for a chain of restaurants. Clients can make their reservations personally and in some cases the maître can take care of them. For example, if a person goes into the restaurant and he has made no reservation and asks for a table, a reservation must be made in the system before assigning a table to him. In all the cases the person who makes a reservation can do it through an interactive system available through the Web or through the telephone reservation office of the chain of restaurants. This includes the maître if, for example, the Web connection fails at the time a client arrives without a reservation. In order to make a reservation the system must check if there is any table available for the number of guests requested. The allocation of a specific table is not made until the clients arrive to the restaurant. At this moment the maître performs the registration process, which involves, among other things, getting access to the reservation information and assigning an available table for the clients. Before the clients leave the restaurant, the maître must access again the reservation in order to get the data that are needed to calculate the bill and print it.

2. Specify the **sequence diagram** for the buy action of an online sales application, which is deFined in a class called *ShoppingCart*. The following assumptions must be made for the application:

   a. Objects of the ShoppingCart class keep the information of a web order represented as a list of all the products (instances of the *Product* class) the user has added to the cart and the size of the order for each of them.

   b. Before the purchase is confirmed the system has to check that all products are available. This will be done using a unique instance of the class *Warehouse*.

   c. If there are not enough goods, the system will perform a request of the necessary products associating to it the client code. The request will be made through a web service by means of the *Provider* class.

   d. If the order cannot be accomplished successfully, the transaction will end and an error message will be shown to the user. It will not be necessary to cancel previous requests to the provider.

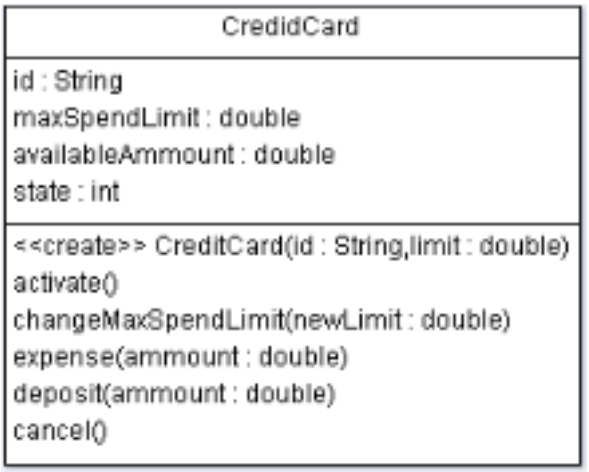
   e. Both if there are enough goods to order as to confirm the purchase and if the request specified in the previous step is successful, the payment will be made by means of a virtual TPV, an instance of the class *PaymentManager*, using the data stored in the *User* object that represents the user who has made the order.

   f. Upon collection, the order will be shipped to the store (Warehouse class) to proceed to its delivery.

3 Specify the sequence diagram for the following method:

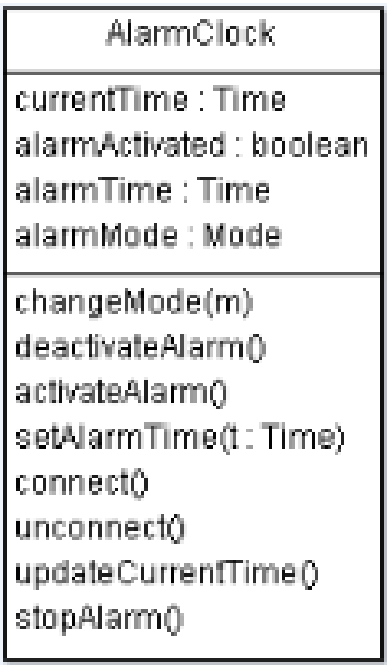
```
public class Puzzle {
   private Fragments parts;

   void solve()
   {
          Part   p1;
          Part   p2;
          Part p3;

          while(parts.length > 1)
          {
                 p1 = parts.select();
                 p2 = parts.select();

                 if (p1.match(p2))
                 {
                        p3 = new Part(p1, p2);
                        parts.drop(p1);
                        parts.drop(p2);
                        parts.add(p3);
                 }
          }
    }
}
```

4. Specify the **state transition diagram** that models the behavior of instances of the *CreditCard* class described below, and describe its main aspects.



When a credit card is manufactured it is deactivated and an identifier and a maximum spending limit are assigned to it. It is delivered to its owner, who can just cancel it or activate it. Once the holder activates the card, he can start using it and the maximum spending limit is available. For all expenses (purchases and ATM withdrawals) the available amount is decremented accordingly. Similarly, for all deposits (purchase returns and cash transfers) it is increased. At any moment, if the available amount is negative the card is only available for deposits until the available amount is positive. If there is an expense during this time the card is blocked. A blocked card cannot be used until the next monthly payment is performed. Monthly payments restore the available amount for the card to the maximum limit. The maximum limit of the card can be changed (increased or reduced) immediately after a monthly settlement but before the first expense is made. The card can only be canceled when its available amount matches the maximum limit and once it is canceled it cannot be reactivated.

5. Specify the **state transition diagram** that models the behavior of instances of the *AlarmClock* class described below, and describe its main aspects.



When an alarm clock is turned on its alarm is off and its alarm time is not set. While it is on, the user can set or change the alarm time unless the alarm is triggered. Once the alarm time is set the alarm is activated and the clock checks constantly if it is the time to fire the alarm. The system invokes automatically and periodically (at least every second) the method that updates the current time to allow the clock to take control of time evolution. Unless the alarm has been disabled, when the alarm time arrives the alarm will produce a signal (sound, light, or music) that depends on its mode, until the user stops it. Initially the alarm is in acoustic mode and it can be also in luminous and musical modes. The user can change the mode of the clock at any time except when the alarm is triggered and producing the corresponding signal. When the user stops the alarm, the clock continues working in the same way and with the same alarm time and the alarm stays activated until the alarm is reshoot the next day if no events are triggered before. If the alarm clock is disconnected, the alarm is automatically disabled (if it was active) and its alarm time is erased (if it was set). When the clock is connected again, it returns to the audible alarm mode and its alarm time is not set.

6. A sales online application includes the *ShoppingCart* class described below. Specify the **state transition diagram** that models the behavior of its instances, and describe its main aspects.

When shopping carts are created they are empty and no shipping address is assigned to them. When a product is added, the cart starts keeping track of the products it contains. While containing products, the corresponding amounts can be modified and products can be removed from the cart. Regardless of whether the cart is empty or not, a shipping address may be assigned at any moment. A basket that contains products and has a shipping address can be closed. At that moment, the total amount for the order including tax and shipping expenses is calculated. A closed cart can be reopened so that the products it contains and its shipping address can be changed. When a cart is closed the order can be paid and once payment has been successfully completed it is recorded as sent. Once it is received by the customer, the order is recorded as terminated.