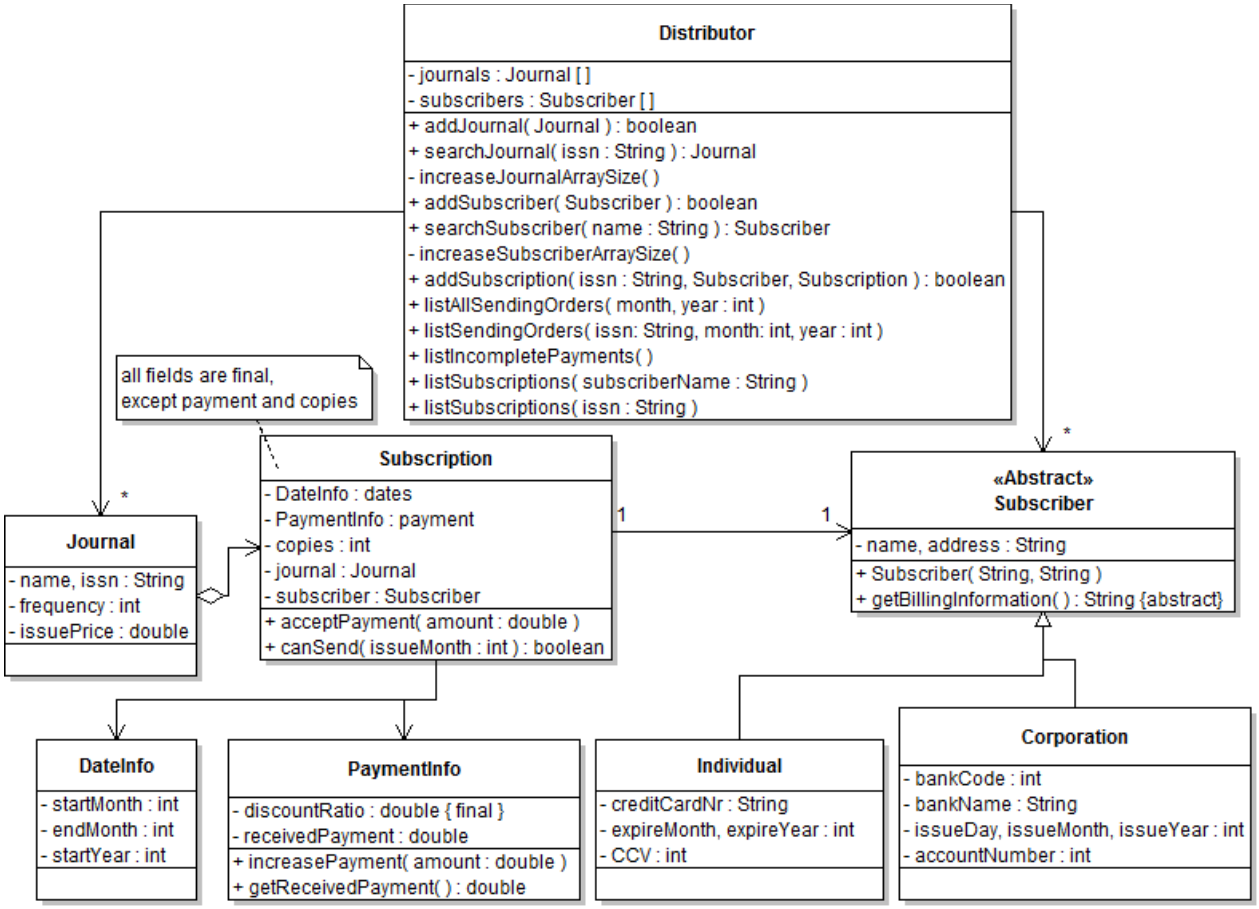
BLM 2911 OBJECT ORIENTED CONCEPTS, Gr.1&Gr.2, FINAL EXAM (05/01/2015)

Student Number:				**Name, Surname:**					**Signature:**
Scoring:	**1**	**2**	**3**	**4**	**5**	**6**	**Group Nr.**	**Duration:**	
	20	30	20	15	15			90mins.	

DESIGN AND PROBLEM DOMAIN



Attention: All the questions should be answered according to UML diagram above. Please note that **there is hidden information in the diagram but all class relations are shown**.

Problem Domain: You are writing software for a distributor company that sells journals to subscribers. Individuals pay for their subscription by credit card and corporations pay by bank cheques. A subscriber can pay in partial amounts; therefore the payment that has been received so far is kept in a PaymentInfo object. A subscriber can subscribe to multiple copies of the same journal. A subscription continues for one year but it can begin in any month. For example, a subscription may begin in March 2014 and this means that it will end in February 2015 (inclusive). The ISSN is a unique identifier of a journal. The frequency of a journal denotes how many issues are published within a year. Each journal has an issue price but subscribers can be granted a discount ratio.

Questions are given in the second paper. Do not forget to write your name, number and signature in all papers.

Question 1: Write the Java source code of classes Subscriber and Individual. The member fields of Subscriber are all related with billing information. The toString method of a subscriber instance returns a String that contains the following information about itself: Name, address and billing detail.

Question 2: Write the Java source code of the addSubscription method of class Distributor. A subscription for a non-existent subscriber or journal must not be created. If a subscription object for the given subscriber and journal exists, its copies field will be increased by one in this method.

Question 3: Write the Java source code of the listAllSendingOrders method of class Distributor. This method lists which journals will be sent to which subscribers for a given month and year. We must not send journals that have not been paid for. Use the result of the readily available canSend method shown in the UML diagram for this purpose.

Question 4: Write the Java source code of the addJournal method of class Distributor.

Question 5: Draw the sequence diagram of the searchJournal method of class Distributor.

Question 1: Write the Java source code of classes Subscriber and Individual.

```
public abstract class Subscriber {
   private String name, address;

   public Subscriber(String name, String address) {
      this.name = name; this.address = address;
   }
   public String getName() { return name; }
   public String getAddress() { return address; }
   public String toString() {
      String info = "Bill to: " + name + " " + address + "\n";
      info += getBillingInformation();
      return info;
   }
   public abstract String getBillingInformation( );
}
public class Individual extends Subscriber {
   private String creditCardNr;
   private int expireMonth, expireYear, CCV;

   public Individual(String name, String address) {
      super( name, address );
   }
   public void setCreditCardNr(String creditCardNr) { this.creditCardNr = creditCardNr; }
   public void setExpireMonth(int expireMonth) { this.expireMonth = expireMonth; }
   public void setExpireYear(int expireYear) { this.expireYear = expireYear; }
   public void setCCV(int cCV) { CCV = cCV; }
   public String getBillingInformation() {
      String info = "Card Nr: " + creditCardNr;
      info += " Exp: " + expireMonth + "/" + expireYear;
      info += " CCV: " + CCV;
      return info;
   }
}
```

Question 2: Write the Java source code of the addSubscription method of class Distributor.

```
public boolean addSubscription(String issn,String subscriberName,Subscription subscription){
   Journal journal = searchJournal(issn);
   Subscriber subscriber = searchSubscriber(subscriberName);
   if( journal == null || subscriber == null )
      return false;
   Subscription[] subscriptions = journal.getSubscriptions();
   for( int i=0; i<subscriptions.length; i++ )
      //i<journal.getSubscriptionCount derse null kontrolüne gerek kalmaz.
      if( subscriptions[i] != null && subscriptions[i].getSubscriber() == subscriber ) {
         subscriptions[i].addCopy();
         return true;
      }
   journal.addSubscription( subscription );
   return true;
}
```

Question 3: Write the Java source code of the listAllSendingOrders method of class Distributor.

UML sınıf şemasında Distributor ile Subscription arasında doğrudan ilişki olmadığına, Subscription nesnelerine ilgili Journal nesneleri üzerinden erişilmesi gerektiğine dikkatinizi çekerim.

```
   public void listAllSendingOrders( int month, int year ) {
      for( int j=0; j<journalCount; j++ ) {
         Subscription[] subscriptions = journals[j].getSubscriptions();
         for( int i=0; i<journals[i].getSubscriptionCount(); i++ ) {
            if( subscriptions[i].canSend(month) ) {
               System.out.println( subscriptions[i] );
            }
         }
      }
   }
```

Question 4: Write the Java source code of the addJournal method of class Distributor.

```
   public void addJournal( Journal journal ) {
      if( searchJournal(journal.getIssn()) != null )
         return;
      if( journalCount == journals.length )
         increaseJournalArraySize();
      journals[journalCount] = journal;
      journalCount++;
   }
```

Question 5: Draw the sequence diagram of the searchJournal method of class Distributor.