//Bertram  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
public class Member {  
 //fields  
 private String cpr;  
 private String firstName;  
 private String lastName;  
 private String phone;  
 private String address;  
 private boolean isActive;  
 private boolean isCompetitive;  
 private double balance;  
 private Subscription subscription;  
  
 //default constructor  
 public Member(){  
 cpr = "null";  
 firstName = "null";  
 lastName = "null";  
 phone = "null";  
 isActive = false;  
 isCompetitive = false;  
 balance = 0.0;  
 subscription = new Subscription();  
 }  
  
 //constructor  
 public Member(String cpr, String firstName, String lastName, String phone, String address, boolean isActive, boolean isCompetitive, double balance, ArrayList<Subscription> subscriptions) throws Exception {  
 this.cpr = cpr;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.phone = phone;  
 this.address = address;  
 this.isActive = isActive;  
 this.isCompetitive = isCompetitive;  
 this.balance = balance;  
 subscription = subscriptions.get(findSubscriptionId());  
 }  
  
 //getters  
 public String getCpr(){  
 return cpr;  
 }  
 public String getFirstName(){  
 return firstName;  
 }  
 public String getLastName(){  
 return lastName;  
 }  
 public String getPhone(){  
 return phone;  
 }  
 public String getAddress(){  
 return address;  
 }  
 public boolean getIsActive(){  
 return isActive;  
 }  
 public boolean getIsCompetitive(){  
 return isCompetitive;  
 }  
 public double getBalance(){  
 return balance;  
 }  
 public Subscription getSubscription(){  
 return subscription;  
 }  
  
 //setters  
 public void setCpr(String cpr){  
 this.cpr = cpr;  
 }  
 public void setFirstName(String firstName){  
 this.firstName = firstName;  
 }  
 public void setLastName(String lastName){  
 this.firstName = firstName;  
 }  
 public void setPhone(String phone){  
 this.phone = phone;  
 }  
 public void setAddress(String address){  
 this.address = address;  
 }  
 public void setIsActive(boolean isActive){  
 this.isActive = isActive;  
 }  
 public void setIsCompetitive(boolean isCompetitive){  
 this.isCompetitive = isCompetitive;  
 }  
 public void setBalance(double balance){  
 this.balance = balance;  
 }  
 public void setSubscription(Subscription subscription){  
 this.subscription = subscription;  
 }  
  
 //methods  
 public String toString(){  
 return cpr + ":" + firstName + ":" + lastName + ":" + phone + ":" + address + ":" + isActive + ":" + isCompetitive + ":" + balance;  
 }  
 //style this one  
 /\*  
 public String showMember(){  
 return "CPR: " + cpr + "\nFirst name: " + firstName + "\nLast name: " + lastName + "\nPhone: " + phone + "\nActive: " + isActive + "\nCompetitive: " + isCompetitive + "\nBalance: " + balance + "\nSubscription: " + subscription.getType();  
 }  
 \*/  
 public int getAge()throws Exception{  
 //declare the format of cpr (day day month month year year)  
 SimpleDateFormat dateFormat = new SimpleDateFormat("ddMMyy");  
 //create an instance of Calender called birth and parse the first 6 characters of cpr to it.  
 Calendar birth = Calendar.getInstance();  
 birth.setTime(dateFormat.parse(cpr.substring(0, 6)));  
  
 //create another Calendar instance called age (set by default to the systems time)  
 Calendar age = Calendar.getInstance();  
  
 //SimpleDateFormat is kinda wierd, when only specifying two numbers for years, it won't go back more than 80 years  
 //if the birthdate year is set in the future, subtract 100 years to get the right birthdate  
 if (birth.get(Calendar.YEAR) > age.get(Calendar.YEAR))  
 birth.add(Calendar.YEAR, -100);  
  
 //subtract birthdate from current date  
 age.add(Calendar.DAY\_OF\_MONTH, -birth.get(Calendar.DAY\_OF\_MONTH));  
 age.add(Calendar.MONTH, -birth.get(Calendar.MONTH));  
 age.add(Calendar.YEAR, -birth.get(Calendar.YEAR));  
  
 //now the age Calendar object is equal to the age of the Member if he was born 01-01-0000  
 //return the year  
 return age.get(Calendar.YEAR);  
 }  
  
 public String getGender(){  
 //get last digit from cpr  
 int genderInt = Integer.parseInt(cpr.substring(cpr.length()-1));  
 //even == female  
 if (genderInt % 2 == 0)  
 return "Female";  
 return "Male";  
 }  
  
 //there is only 4 types of subscription, using the members age and activity flag we return the id of the correct subscription  
 public int findSubscriptionId()throws Exception{  
 int age = getAge();  
 //if member is passive  
 if (!isActive)  
 return 3;  
 //if member is junior  
 if (age < 18) {  
 return 0;  
 }  
 //if member is senior  
 if (age > 18 && age < 60)  
 return 1;  
 //if none of the above was true, the member must be a veteran  
 return 2;  
 }  
}