UML Diagram

FileIO

File membersFile
File instructionFile
File reportFile
File reultFile
Club c
StringBuilder sbr

public void saveResult()
public void saveReport()
public void readInstruction()
public void readMember()

AMR

public static void main()

0..1   0..1

0..1   0..*

0..*

CustomDate

SimpleDateFormat dataFormat
String dateinString
Date date

public boolean isValid()
public String toString()

Member

String number;
String name;
String birthday;
String tier;
String address;
double mileage;
double points;

public String getNumber()
public void setNumber(String num)
public String getName()
public void setName(String na)
public String getBirthday()
public void setBirthday(String birth)
public String getTier()
public void setTier(String t)
public String getAddress():String
public void setAddress(String[] addr)
public double getMileage()
public void setMileage(double mile)
public double getPoints()
public void setPoint(double p)
public boolean isVaildToAdd()
public boolean VaildNumber()
public boolean VaildName()
public void earnPoints(double m)
public void redeemPoints(double po)
public String toString()

0..1

Club

ArrayList<Member> memberList

public void deleteMember(String s)
public void addMember(String num, String na)
public ArrayList<Member> getClub()
public void setClub(ArrayList<Member> memberlist)
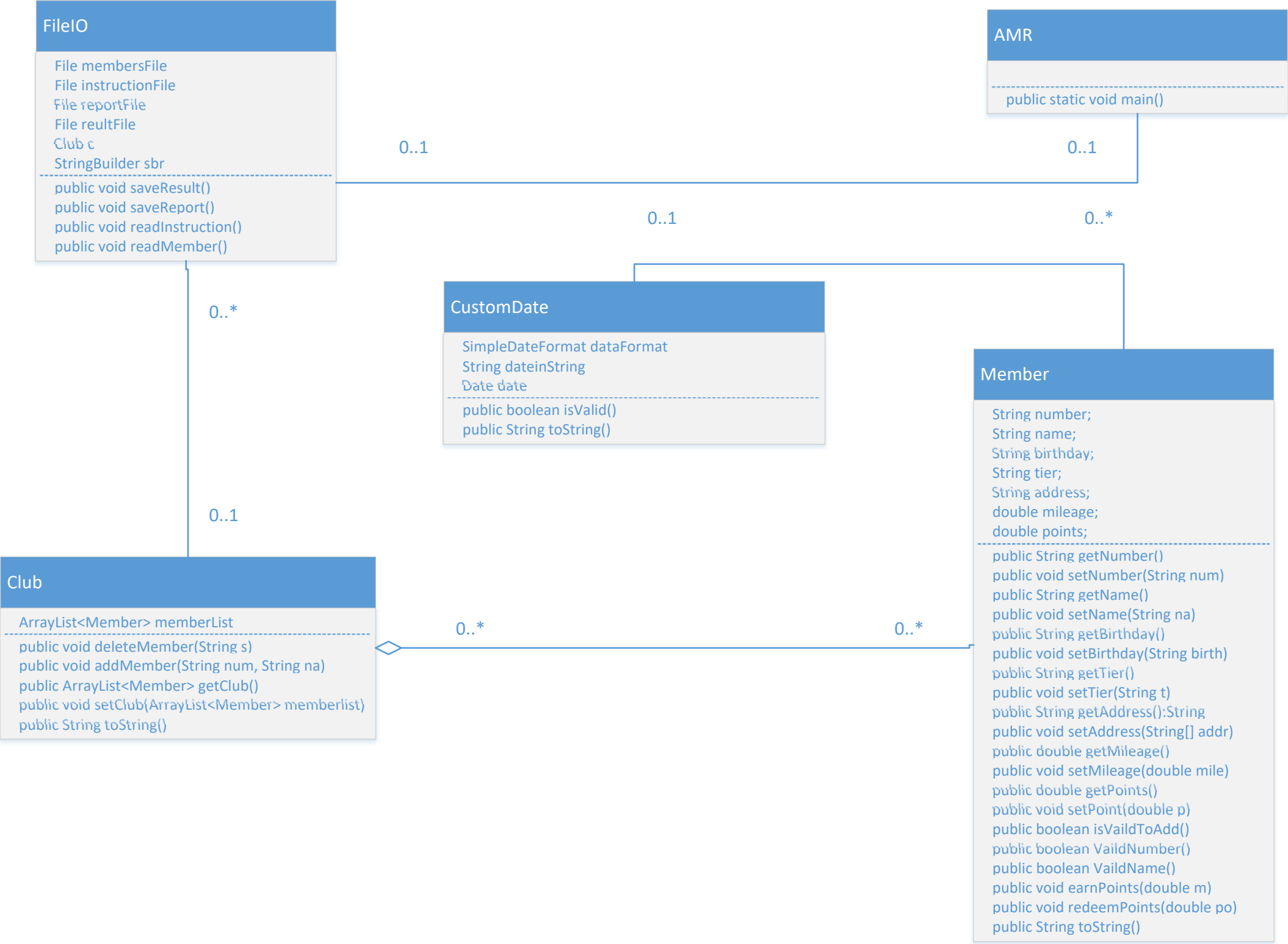public String toString()

0..*   0..*

Design Discription

I designed 5 classes whose names are Member, CustomDate, Club, FileIO, and AMR.

The Member class defines fields of a member and contains some getters and setters to get or set the value of each field. Besides, because the compulsory fields of a member are the number and the name, this class also provide separated validation checking methods to determine whether the number and the name of a member are valid. The validation checking part also shows that if all the compulsory fields of a member are valid, the member can be add to the club, which is an object of the Club class. As to the setters of other fields of a member, the validation checking parts have been designed as the conditions of if-else assignments in the methods. The earnPoints and redeemPoints methods are also in the Member class. There is also a toString method which uses StringBuilder to transfer the member into a customized String.

The CustomDate class can check whether a provided date is valid date and can transfer the date String into dd/mm/yyyy format.

The Club class defines an ArrayList to store different members. The addMember and deleteMember methods can add or delete members into or from the list. In the adding part, the parameters are two Strings which refer to the number and the name of the member. The deleting part only require one String parameter referring to a given name. Additionally, there are one setter and one getter to set and get the whole list. Just like the Member class, a toString method is designed to transfer the list into a customized String.

The FileIO class aims at reading input files and writing output files. The readInstruction method reads the instruction file and recognizes various operations. After determining the method of an operation, it can recognize the specific value of the method. Then modifications are done to the members in the club. The readMember method reads the member file and save valid members into the club. While adding a new member into the club, at first, the club will be gone through and to see whether there is member with the same number already existing. If there is no same-number member in the club, just adding a member simply has a valid number and a valid name. The tier field is "Silver", which is the default value. Other fields are empty. Then, if the member is provided valid value of other fields, using setters to set the values to the corresponded fields. If there is a same-number member already in the club, updating the existing same-number member with the valid values of fields of the new member in the member file. The querying parts are also in the FileIO class. The querying result are returned as a customized String. The saveResult method outputs the customized String, transferred from the club, into a txt file. The saveReport method outputs the querying result, into a txt file one by one.

The AMR class only contains a main method to run the program.