Programming Principles II

Course Work 02

SE

UOW No: W1790334

IIT No: 20191193

Name: L.A.D.R. Gunawardana

YouTube Link: https://youtu.be/9Ca5428CmhI

Content

1. OOP Concepts (Object Oriented Programming)

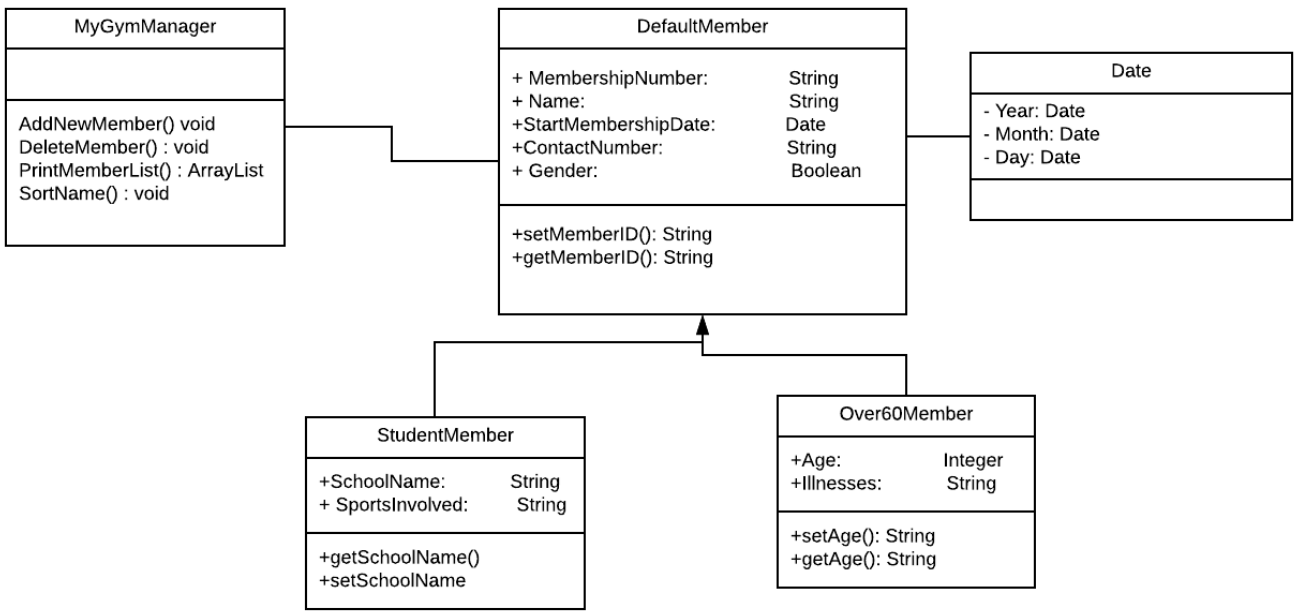
There are four main OOP concepts.(In Brief)

1. **Abstraction:** Data abstraction hide certain details and show only essential information.
2. **Encapsulation:** Hide the internal representation of an object from the outside.
3. **Inheritance:** Derives a class from another class for a hierarchy of classes that share a set of attributes and methods.
4. **Polymorphism:** Different classes can be used with same interface.

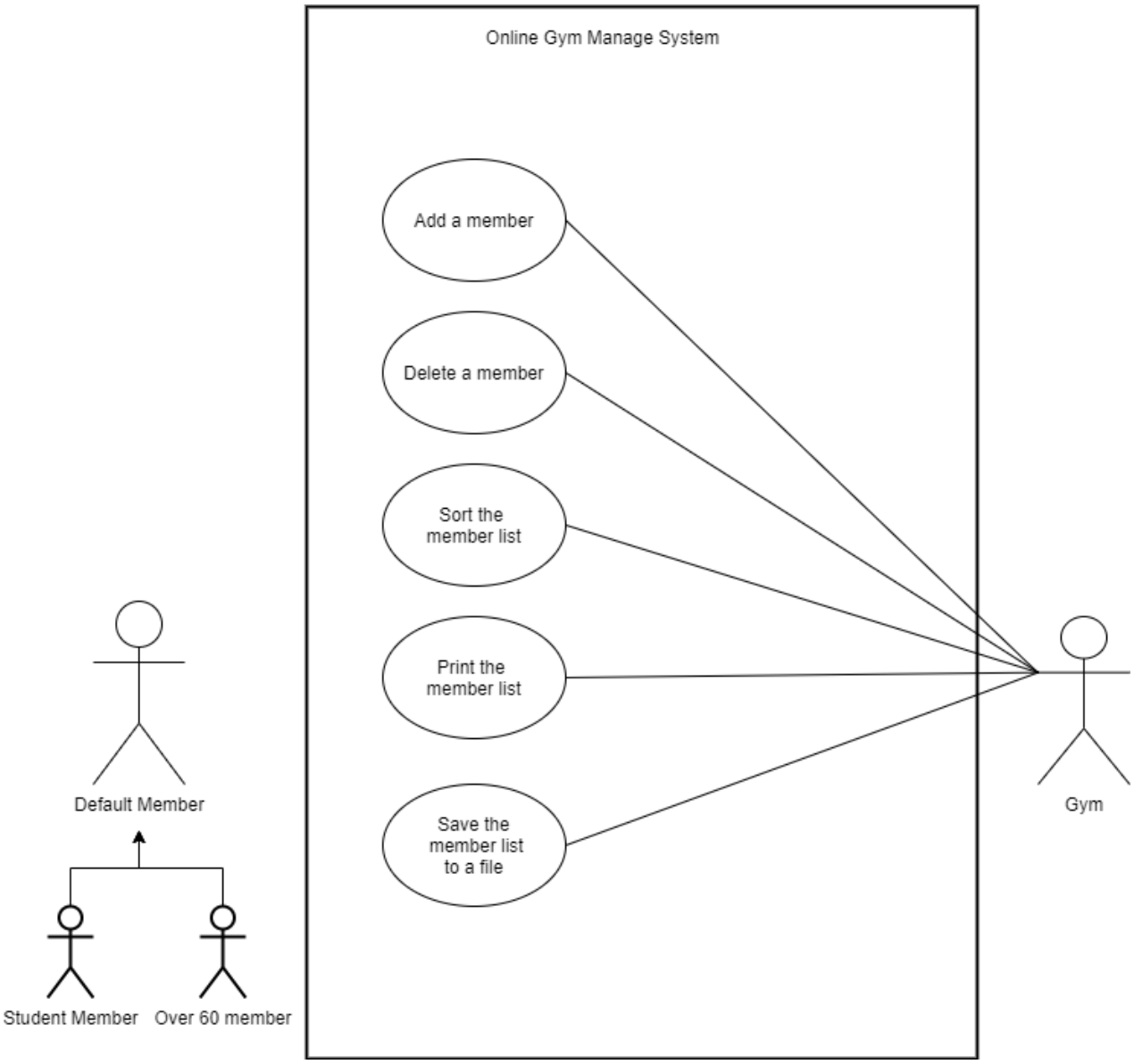
2. Functionalities of the Gym Management System

1. Add new member
2. Delete a member
3. Sort the member list
4. Print the member list
5. Save the member list
6. Open GUI

3. UML Class Diagram



4. Use Cases


Online Gym Manage System
Add a member
Delete a member
Sort the member list
Print the member list
Save the member list to a file
Default Member
Student Member
Over 60 member
Gym

5. Test Plan

No	Criteria	Correct Output
1	Class diagram (UML notation)	Done
2	Two or more use cases (UML notation)	Done
3	Use a super class: DefaultMember	Done
4	StudentMember class information and relative get/set methods + SchoolName	Done
5	Over60Member class information and relative get/set methods + Age	Done
6	Date Class	Incompl;r
7	GymManager Interface	
8	MyGymManager maintain the member list and provides all methods for gym manager.	Done
9	Add a new member	Done
10	Delete a member	Done
11	Sort the member list	Done
12	Print the member list	Done
13	Save the member list	
14	Open GUI	
15	GUI: Manager can visualize the member list by a table	
16	GUI: Manager can search members	
17	Test Plan	Done
18	Robustness of the code	To be measured

19	Quality of the code	To be measured
20	YouTube Video	Done

6. YouTube link: https://youtu.be/9Ca5428CmhI

7. Code

AddDefaultMember

```
package gymsystem;

import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.DBObject;
import java.util.Scanner;

// Adding a new default member

public class AddDefaultMember {
    public static Boolean TakenID;
    public static BasicDBObject basicDBObject = new BasicDBObject();
    public static void addDefaultMember() throws Exception {

        basicDBObject.clear();
        Scanner scanner1 = new Scanner(System.in);

        //Inserting the member ID s

        System.out.println("Type ID of the Member");

        while (true) {
            try {
                TakenID=false;

                String memberid = scanner1.nextLine();
                DefaultMember.setMemberID(memberid);
                SetDataBase.setDB();

                DBCollection table1 = SetDataBase.database.getCollection("Members");
                DBCursor findIterable = table1.find();
                for (DBObject totalCount : findIterable) {

                    String dataBaseID = (String) totalCount.get("Membership Number");
                    String selectedID = DefaultMember.getMemberID();
```

```
                    if (selectedID.equals(dataBaseID)) {
                        TakenID = true;
                        break;
                    }
                }

                if (TakenID){
                    System.out.println("ID already taken,please try another.");

                }else {
                    basicDBObject.put("Membership Number", DefaultMember.getMemberID());
                    System.out.println("You have successfully entered the data");
                    break;
                }

            }catch (notInRange exception){
                System.out.println("MemberID is not in Range");
            }
        }

        //Inserting the member names

        System.out.println("Type Name of the Member");

        while (true) {
            try {
                String name = scanner1.nextLine();
                DefaultMember.setName(name);
                SetDataBase.setDB();

                DBCollection table1 = SetDataBase.database.getCollection("Members");
                basicDBObject.put("Member Name", DefaultMember.getName());
                System.out.println("You have successfully entered the data");
                break;

            } catch (notInRange exception) {
                System.out.println("Member Name is not valid");


            }
        }

    }

}
```

AddNewMember

```
package gymsystem;

import com.mongodb.DBCollection;
import java.util.Scanner;

public class AddNewMember {
```

```
    public static void addNewMember() throws Exception{

        Scanner scanner1= new Scanner(System.in);

        //Instructions

        System.out.println("Choose the member category");
        System.out.println("To choose default member, type as 'default'");
        System.out.println("To choose student member, type as 'student'");
        System.out.println("To choose over 60 member, type as 'over60'");

        while (true){
            String input = scanner1.nextLine();
            SetDataBase.setDB();
            DBCollection table1 = SetDataBase.database.getCollection("Members");

            // default member detail method

            if (input.equals("default")){
                AddDefaultMember.addDefaultMember();
                AddDefaultMember.basicDBObject.put("Category","default");
                table1.insert(AddDefaultMember.basicDBObject);
                break;

                // default member detail method + student member detail method

            }else if (input.equals("student")){
                AddDefaultMember.addDefaultMember();
                AddStudentMember.addStudentMember();
                AddDefaultMember.basicDBObject.put("Category","student");
                table1.insert(AddDefaultMember.basicDBObject);
                break;

            }else if (input.equals("over60")){

                AddDefaultMember.addDefaultMember();
                AddOver60Member.addOver60Member();
                AddDefaultMember.basicDBObject.put("Category","over60");
                table1.insert(AddDefaultMember.basicDBObject);

                // default member detail method + over60 member detail method
                 break;
            }else{
               System.out.println("Invalid Input");
            }

        }

    }

}
```

AddOver60Member

```
package gymsystem;

import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
import javax.naming.InvalidNameException;
import java.util.Scanner;

//Adding over 60 members

public class AddOver60Member {
    public static void addOver60Member() {

        //Age is unique for this class

        Scanner scanner1 = new Scanner(System.in);
        System.out.println("Type Age of the Member");

        while (true) {
            try {

                String name1 = scanner1.nextLine();
                Over60Member.setAge(name1);
                SetDataBase.setDB();

                DBCollection table1 = SetDataBase.database.getCollection("Members");
                BasicDBObject basicDBObject = new BasicDBObject();
                AddDefaultMember.basicDBObject.put("Age", Over60Member.getAge());

                System.out.println("You have successfully entered the data");
                break;

            } catch (InvalidNameException exception) {
                System.out.println("Member Age is not in valid");


            }
        }
    }
}
```

AddStudentMember

```
package gymsystem;

import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
import javax.naming.InvalidNameException;
import java.util.Scanner;

//Adding student members

public class AddStudentMember {

    public static void addStudentMember() {
```

```
        //School name is unique for this class

        Scanner scanner1 = new Scanner(System.in);
        System.out.println("Type School Name of the Member");

        while (true) {
            try {

                String name1 = scanner1.nextLine();
                StudentMember.setName(name1);
                SetDataBase.setDB();

                DBCollection table1 = SetDataBase.database.getCollection("Members");
                BasicDBObject basicDBObject = new BasicDBObject();

                AddDefaultMember.basicDBObject.put("SchoolName",
StudentMember.getName());
                System.out.println("You have successfully entered the data");
                break;

            } catch (InvalidNameException exception) {
                System.out.println("Member Name is not in valid");


            }
        }
    }
}
```

DataBaseLink

```
package gymsystem;
import com.mongodb.MongoClient;
import com.mongodb.DB;

public class DataBaseLink {
    public static DB database;
    //local host
    public static MongoClient mongoClient= new MongoClient("localhost",27017);

    public static void setDB(){

        //Creating the database

        database = mongoClient.getDB("Gym");
        database.createCollection("MemberDetails",null);
    }
}
```

Date

```
package gymsystem;
```

```
public class Date {
}
```

DefaultMember

```
package gymsystem;

import java.util.Arrays;
import java.util.Collection;

class notInRange extends Exception{
}

class InvalidNameException extends Exception{
}

public class DefaultMember {
    private static String MemberID;
    private static String Name;

    //Checking the membership number length

    public static void setMemberID(String memberid) throws Exception {
        if (memberid.length() == 5) {
            MemberID = memberid;
        } else {
            throw new notInRange();
        }
    }

    public static String getMemberID() {return MemberID;};

    //Checking the Name

    public static void setName(String name) throws Exception {
        char[] chars = name.toCharArray();
        Collection<Character> uniqueCharacters = Arrays.asList(' ');
        for (char char1 : chars) {

            //Here the if condition is for; if character is a letter or a space the name
get set to the Member name variable.

            if (Character.isLetter(char1) || uniqueCharacters.contains(char1)) {
                Name = name;
            } else{
                throw new InvalidNameException();
        }

    }

}
    public static String getName(){
        return Name;
    }
```

```
}
```

DeleteMember

```
package gymsystem;

import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.DBObject;
import java.util.Scanner;

//Deleting a member

public class DeleteMember {
    public static Boolean TakenID;
    public static void delete() {

        Scanner scanner1 = new Scanner(System.in);
        System.out.println("Type ID of the Member to delete");

                //First we set the existing ID to false.

        while (true) {
            try {

                //When it gets false method will run

                TakenID = false;

                String memberid = scanner1.nextLine();
                DefaultMember.setMemberID(memberid);
                SetDataBase.setDB();

                DBCollection table1 = SetDataBase.database.getCollection("Members");
                DBCursor findIterable = table1.find();

                for (DBObject totalCount : findIterable) {
                    String dataBaseID = (String) totalCount.get("Membership Number");
                    String selectedID = DefaultMember.getMemberID();

                    if (dataBaseID.equals(selectedID)) {
                        TakenID = true;
                        break;
                    }
                }

                if (TakenID) {
                    BasicDBObject basicDBObject = new BasicDBObject();
                    basicDBObject.put("Membership Number", DefaultMember.getMemberID());
                    table1.findAndRemove(basicDBObject);
                    System.out.println("Data successfully deleted");
                } else {
```

```
                    System.out.println("Input a valid Membership Number");
                    break;

                }

            } catch (notInRange exception) {
                System.out.println("MemberID is not in Range");

            }catch (Exception r) {
                r.printStackTrace();

            }
        }
    }
}
```

MyGymManager

```
package gymsystem;

import javafx.application.Application;
import javafx.stage.Stage;
import java.util.Scanner;

public class MyGymManager extends Application {
    public static String sort;

    @Override
    public void start(Stage primaryStage) throws Exception {
        while (true){
            sort = "no";
        Scanner scanner1 = new Scanner(System.in);
        System.out.println("Instructions");
        System.out.println("A - add a new member");
        System.out.println("X - delete a member");
        System.out.println("R - sort the member list");
        System.out.println("P - print the member list");
        System.out.println("S - save the member list");
        System.out.println("T - open the saved file");
        System.out.println("G - open GUI");

        while (true) {
            String answer = scanner1.nextLine();

            if (answer.equals("A")) {
                AddNewMember.addNewMember();     //new adding
                break;

            } else if (answer.equals("X")) {
                DeleteMember.delete();          //Deleting a member
                break;

            } else if (answer.equals("R")) {
                break;                          //Sorting the member list
```

```
            } else if (answer.equals("P")) {
                Print.PrintList();            //printing the member list
                break;

            } else if (answer.equals("S")) {
                sort = "yes";                  //saving the member list
                break;

            } else if (answer.equals("T")) {
                break;                       //open the saved the member list

            } else if (answer.equals("G")) {
                break;                         //open the GUI

            } else {
                System.out.println("Invalid Input");

                }


            }


        }
    }
}
```

Over60Member

```
package gymsystem;

import javax.naming.InvalidNameException;
import java.util.Arrays;
import java.util.Collection;

//over 60 members

public class Over60Member {
    private static String Age ;

    public static void setAge(String age) throws InvalidNameException {

        char[] chars = age.toCharArray();
        Collection<Character> uniqueCharacters = Arrays.asList(' ');

        for (char char1 : chars) {

            //letter or space...

            if (Character.isLetter(char1) || uniqueCharacters.contains(char1)) {
                Age = age;

            } else {
                throw new InvalidNameException();
            }
```

```
        }

    }

    public static String getAge(){
        return Age;
    }
}
```

Print

```
package gymsystem;

import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.DBObject;

//Print the member list with all details

public class Print {
    public static DBCursor counter1;
    public static void PrintList(){

        BasicDBObject basicDBObject= new BasicDBObject("Name",1);
        SetDataBase.setDB();

        DBCollection table1 = SetDataBase.database.getCollection("Members");

        if (MyGymManager.sort.equals("yes")) {
            counter1 = table1.find().sort(basicDBObject);

        }else{
            counter1 = table1.find();

        }for (DBObject totalCount : counter1) {

            String MemberID = (String) totalCount.get("Membership Number");
            String name= (String) totalCount.get("Member Name");
            String category= (String) totalCount.get("Category");
            String SchoolName= (String) totalCount.get("SchoolName");
            String Age=(String)totalCount.get("Age");

            System.out.println("MemberID "+MemberID);
            System.out.println("Name "+name);
            System.out.println("Category "+category);

            //Since school name and age are unique for relevant classes,

            if (category.equals("Student")){
                System.out.println("SchoolName: "+SchoolName);

            }if (category.equals("Over60Member")){
                System.out.println("Age: "+Age);
```

```
            }

        }
    }
}
```

StudentMember

```
package gymsystem;

import javax.naming.InvalidNameException;
import java.util.Arrays;
import java.util.Collection;

//student member

public class StudentMember {
    private static String SchoolName;

    public static void setName(String sclName) throws InvalidNameException {
    char[] chars = sclName.toCharArray();

    Collection<Character> uniqueCharacters = Arrays.asList(' ');

        for (char char1 : chars) {

        //letter or space...

        if (Character.isLetter(char1) || uniqueCharacters.contains(char1)) {
            SchoolName = sclName;

        } else {
                throw new InvalidNameException();
            }

        }
    }

public static String getName(){return SchoolName;}
}
```

8. References

Lecture Notes

W3school : https://www.w3schools.com/

Draw.io : https://app.diagrams.net/

Lucid Chart: https://app.lucidchart.com/documents/edit/4962f48b-89e6-4335-858e-0243f097ba18/0_0?beaconFlowId=67791E19A19B3CE7

****