Maven

SonarQube

MySQL (server working or not)

Java Practise handson

Subjective test week 4

Method Reference examples

UML (if required)

GIT (Working or not)

Wave 2 Day 5

PRACTICE QUESTIONS DAY 5 | WAVE II

* [**Armstrong Numbers**](https://practice.geeksforgeeks.org/problems/armstrong-numbers2727/1?page=1&difficulty%5b%5d=-2&category%5b%5d=Mathematical&category%5b%5d=Tree&category%5b%5d=Searching&sortBy=submissions)
* [**Replace all 0's with 5**](https://practice.geeksforgeeks.org/problems/replace-all-0-with-5-in-an-input-integer/1?page=1&difficulty%5b%5d=-2&category%5b%5d=Mathematical&category%5b%5d=Tree&category%5b%5d=Searching&sortBy=submissions)
* [**Swap two numbers**](https://practice.geeksforgeeks.org/problems/swap-two-numbers3844/1?page=1&difficulty%5b%5d=-2&category%5b%5d=Mathematical&category%5b%5d=Tree&category%5b%5d=Searching&sortBy=submissions)

Completed the challenge discussion.

Handson Task:

Java Mock 27 Req # 5

import java.util.Date;

import java.util.List;

public class Song {

private String title;

private String genre;

private Date duration;

private Double rating;

private List<Artist> artistList;

public Song() {}

public Song(String title, String genre, Date duration, Double rating, List<Artist> artistList) {

super();

this.title = title;

this.genre = genre;

this.duration = duration;

this.rating = rating;

this.artistList = artistList;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

public Date getDuration() {

return duration;

}

public void setDuration(Date duration) {

this.duration = duration;

}

public Double getRating() {

return rating;

}

public void setRating(Double rating) {

this.rating = rating;

}

public List<Artist> getArtistList() {

return artistList;

}

public void setArtistList(List<Artist> artistList) {

this.artistList = artistList;

}

}

//-----------------------------------

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public class Artist {

private String name;

private String role;

private String gender;

private Date dob;

private List<Song> songList;

public Artist() {}

public Artist(String name, String role, String gender, Date dob, List<Song> songList) {

super();

this.name = name;

this.role = role;

this.gender = gender;

this.dob = dob;

this.songList = songList;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

public Date getDob() {

return dob;

}

public void setDob(Date dob) {

this.dob = dob;

}

public List<Song> getSongList() {

return songList;

}

public void setSongList(List<Song> songList) {

this.songList = songList;

}

public static List<Artist> prefill() throws ParseException{

SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

List<Artist> list = new ArrayList<Artist>();

list.add(new Artist("Ed Sheeran","Singer","Male",sdf.parse("17-02-1991"),new ArrayList<Song>()));

list.add(new Artist("Steve Mac","Musician","Male",sdf.parse("15-01-1972"),new ArrayList<Song>()));

list.add(new Artist("Burruss","Singer","Female",sdf.parse("17-05-1976"),new ArrayList<Song>()));

list.add(new Artist("Beyonce","Singer","Female",sdf.parse("04-09-1981"),new ArrayList<Song>()));

list.add(new Artist("Zoey","Musician","Female",sdf.parse("10-10-1994"),new ArrayList<Song>()));

list.add(new Artist("Shawn Mendes","Singer","Male",sdf.parse("08-10-1998"),new ArrayList<Song>()));

list.add(new Artist("Frankie Dunne","Musician","Male",sdf.parse("10-09-1978"),new ArrayList<Song>()));

list.add(new Artist("Luis Fonsi","Singer","Male",sdf.parse("15-04-1978"),new ArrayList<Song>()));

list.add(new Artist("Daddy Yankee","Singer","Male",sdf.parse("03-02-1977"),new ArrayList<Song>()));

list.add(new Artist("Zuleyka","Dancer","Female",sdf.parse("03-10-1987"),new ArrayList<Song>()));

list.add(new Artist("Dan Reynolds","Singer","Male",sdf.parse("14-07-1987"),new ArrayList<Song>()));

list.add(new Artist("Wayne Sermon","Musician","Male",sdf.parse("15-06-1984"),new ArrayList<Song>()));

list.add(new Artist("Andrew","Musician","Male",sdf.parse("03-02-1978"),new ArrayList<Song>()));

list.add(new Artist("Chris Martin","Singer","Male",sdf.parse("02-03-1977"),new ArrayList<Song>()));

list.add(new Artist("Alan Walker","Lyricist","Male",sdf.parse("24-08-1997"),new ArrayList<Song>()));

list.add(new Artist("Jesper","Musician","Male",sdf.parse("16-09-1988"),new ArrayList<Song>()));

list.add(new Artist("Rob","Singer","Male",sdf.parse("20-01-1979"),new ArrayList<Song>()));

list.add(new Artist("Joe","Musician","Male",sdf.parse("15-03-1977"),new ArrayList<Song>()));

list.add(new Artist("Sia","Musician","Female",sdf.parse("18-12-1975"),new ArrayList<Song>()));

list.add(new Artist("Maddie","Dancer","Female",sdf.parse("30-09-2002"),new ArrayList<Song>()));

list.add(new Artist("David Guetta","Lyricist","Male",sdf.parse("07-11-1967"),new ArrayList<Song>()));

list.add(new Artist("Sam Martin","Musician","Male",sdf.parse("07-02-1983"),new ArrayList<Song>()));

return list;

}

static Artist bestSinger(String gender,List<Artist> artistList)

{

//write the code here

return null;

}

}

//-----------------------------

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class App {

public static void main(String[] args) throws ParseException {

Scanner sc=new Scanner(System.in);

List<Artist> allArtists = Artist.prefill();

System.out.println("Enter the number of songs:");

int noOfSongs=sc.nextInt();

List<Song> songList=new ArrayList<>();

for(int i=0;i<noOfSongs;i++)

{

String detail=sc.nextLine();

if(detail.equals(""))

detail=sc.nextLine();

String[] arr = detail.split(",");

// Shape of You,Pop,03:53,4.9,Ed Sheeran-Steve Mac-Burruss

Song song=new Song();

song.setTitle(arr[0]);

song.setGenre(arr[1]);

SimpleDateFormat sdf=new SimpleDateFormat("hh:mm");

song.setDuration(sdf.parse(arr[2]));

song.setRating(Double.parseDouble(arr[3]));

String[] aarr = arr[4].split("-");

List<Artist> artistList=new ArrayList<>();

for(String aName:aarr)

{

//find artist object from prefill using aName

for(Artist a : allArtists)

{

if(a.getName().equals(aName))

artistList.add(a);

}

}

song.setArtistList(artistList);

songList.add(song);

}

System.out.println("Enter the gender of singer:");

String gender=sc.next();

Artist result = Artist.bestSinger(gender, allArtists);

System.out.println("The best "+gender+" singer is "+result.getName());

}

}

Complete the Artist class bestSinger method.

And run all test cases.

Maven:

---------

What is Maven?

Is a Building tool

Project build tool.

Have we used any jar file in our project?

Jar= Java Archive similar to .rar or .zip

We add jar files to our project, so that we can make use of the classes, interfaces, methods available in that .jar (library)

If we add lot of jars to the project:

Size of the project is increased

Maintaining the jars when there is a change in version

In every project (ex: hibernate) is used, a copy of those jar files are present in each project

If I use maven,

The size of the project is not increased because, my project folder will not contain the jar files.

Maven has a remote repository where all the jar files are present.

When you create project and add dependency entry in pom.xml, maven checks if these jars are present in local repository or not. If not, then only it downloads.

Group id: com.ust.demo

Artifact id: project-demo-1

Version:

GIT

What is GIT?

Is a version control platform

When multiple programmers work on same project, they contribute code to the same project, but still we should be able to trace, track who made what changes.

Git allow us to collaborate and merge the code.

Mysql server folder and to cmd prompt

nysqd --console

this will start mysql server

JDBC with mysql:

Driver name (automatically chosen)

URL jdbc:mysql://localhost:3306/schema

Username

Password

In pom.xml use this mysql dependency:

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.27</version>

</dependency>

1. Create a maven project
2. In pom.xml add the above dependency

**package** com.ust.demo;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

/\*\*

\* Hello world!

\*

\*/

**public** **class** App

{

**public** **static** **void** main( String[] args ) **throws** SQLException

{

Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/bank","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery("SELECT \* FROM BRANCH");

**while**(rs.next())

{

System.***out***.printf("%-8s\n%-25s%-20s",rs.getString(1),rs.getString(2),rs.getString(3));

}

con.close();

}

}

Lets start sonarqube demo: