Hacettepe University
Computer Science and Engineering Department

Advisers	:	Assoc. Prof. Dr. Ebru SEZER ebru@hacettepe.edu.tr
	:	RA Ali ÇAĞLAYAN alicaglayan@cs.hacettepe.edu.tr
Course	:	BBM104 Programming Lab. II
Experiment No	:	5
Subject	:	Inheritance, Access Modifiers
Programming Language	:	Java
Deadline	:	24.05.2013 17:00

Contents

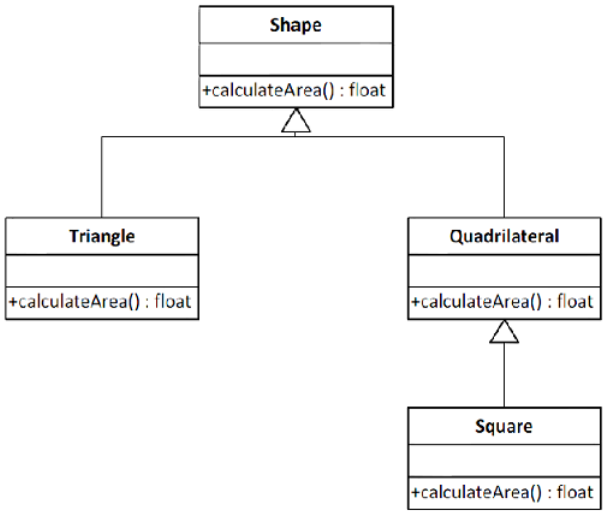


Figure 1: A hierarchy of shape classes

1 Introduction

Object-oriented programming has advantages such as modeling problems with less complexity and more code reuse. In this experiment, you will observe these advantages by using inheritance mechanism which is an important property of object-oriented programming. By the help of this experiment, you will learn the concept of inheritance, relationships among classes by using object references, control of multiple instances of classes, access modifiers in Java.

2 Useful Information

Under this section you will find useful information about project. They are at beginning level. For advanced information you need to make additional research.

2.1 Inheritance

Object-oriented programming (OOP) covers software in terms similar to those that people use to describe real-world objects. It takes advantage of class relationships, where objects of a certain class, such as a class of vehicles, have the same characteristics cars, trucks, little red wagons and roller skates have much in common. Inheritance is one of important property of OOP. OOP takes advantage of inheritance relationships, where new classes of objects are derived by absorbing characteristics of existing classes and adding unique characteristics of their own. In Java, a class(called the **derived class** or **subclass**) extends from another class(called the **base class** or **superclass**).

In Figure 1 a shape hierarchy is seen. Shape class is superclass of all the other classes. Square class is a subclass of Quadrilateral.

2.2 Method Overriding

When a class extend another class, the subclass can use the super class' methods. However sometimes the subclass should change behavior of a method which provided by superclass. The method implementation in the subclass overrides(replaces) the implementation in the superclass. The subclass method and superclass method have the same name, parameters and the same return type. That is called **method overriding**. Each subclass in Figure 1 overrides **calculateArea()** method, replacing its functionality from the superclass. The method behaves different in each subclass.

2.3 Method Overloading

Method overloading is another important concept of OOP. When programmers need more than one method with the same functionality, they don't have to declare new methods with different names for each one. By using method overloading feature, they declare each methods with same name but with different signatures(different argument list, argument types or orders). *System.out.println()* is an example of overloading method in Java. This method takes float, int, double or String as arguments.

2.4 Access Modifiers

In Java, there are four access modifiers which provide access levels for classess and members of classess: **private** (visible to the class only), **default** (visible to the package), **protected** (visible to the package and all subclassess) and **public** (visible to the everywhere).

3 Experiment

This section contains four subsections. The first one is problem definition, and the second one describes the content of your reports. The third one is **constraints** that you should obey. Otherwise your experiment will not be evaluated by your adviser. The last one is about submission.

3.1 Problem Definition

In this experiment, you are supposed to develop a simple Movie Database System similar to IMDB[1]. You are responsible for using inheritance mechanism and access modifiers in Java programming language. The system will process several data input files and will generate results of commands which will be read from a command input file. **All input files will be error free only syntactically.** The requirements and rules for the system are given below:

- There are two types data in this system: **films** and **people**.
- Each **Person** has **name**, **surname**, **country** and a unique **id**.
- A **Person** may have two types in this system. These are **Artist** and **User**.
- Each **User** has a unique **id**, **name**, **surname** and **country**
- There are three kinds of **Artist**: **Performer**, **Director** and **Writer**
- Each **Director** has a unique **id**, **name**, **surname**, **country** and **agent** where he/she works
- Each **Writer** has a unique **id**, **name**, **surname**, **country** and writing **style/type**
- There are also two types of **Performers** which are **Actor** and **Actress**
- Each **Actor** has a unique **id**, **name**, **surname**, **country** and **height**
- Each **Actress** has a unique **id**, **name**, **surname** and **country**
- There are four types of **films** in this system: **Feature Film**, **Short Film**, **Documentary** and **TV Series**
- Each film (Feature Film, Short Film, Documentary and TV Series) has a **rating** score which calculated from users' average rating scores for that film.
- A unique film **id**, film **title**, **language**, duration **length**, **country**, **directors** of a film and **cast**(actors and actresses of a movie) are common in all film types.
- **Feature Films** have a **release date**, **budget**, **writers** of movie and film **genre** in addition to the common data.

[1] www.imdb.com

- A **Short Film** has a **release date**, **writers** and **genre** in addition to the common data. A Short Film **length** should be less (or equal) than **40** min.
- **Documentaries** have only a **release date** in addition to the common film data.
- **TV Series** have **start date** and **end date** of series, number of **seasons**, number of **episodes**, **genre** of series and **writers** in addition to the common film data.
- A film may have more than one directors, writers, performers and genres in this system. A comma will be used to separate these data.

3.1.1 The Way of Execution

The program will be executed with four command line arguments:

<people_file> <films_file> <commands_file> <output_file>

Usage example:
>javac Main.java
>java Main people.txt films.txt commands.txt output.txt

There are three types of data input files and one output file. All the file names will be taken as program arguments. The format of each file is given below.

3.1.2 People File

There are five different recording samples in this file. These are:

DIRECTOR: *<tab> <ID> <tab> <NAME> <tab> <SURNAME>* <tab> <COUNTRY> <tab> <AGENT>

WRITER: *<tab> <ID> <tab> <NAME> <tab> <SURNAME>* <tab> <COUNTRY> <tab> <TYPE>

ACTOR: *<tab> <ID> <tab> <NAME> <tab> <SURNAME>* <tab> <COUNTRY> <tab> <HEIGHT>

ACTRESS: *<tab> <ID> <tab> <NAME> <tab> <SURNAME>* <tab> <COUNTRY>

USER: *<tab> <ID> <tab> <NAME> <tab> <SURNAME>* <tab> <COUNTRY>

A sample people file is shown in Figure 2.

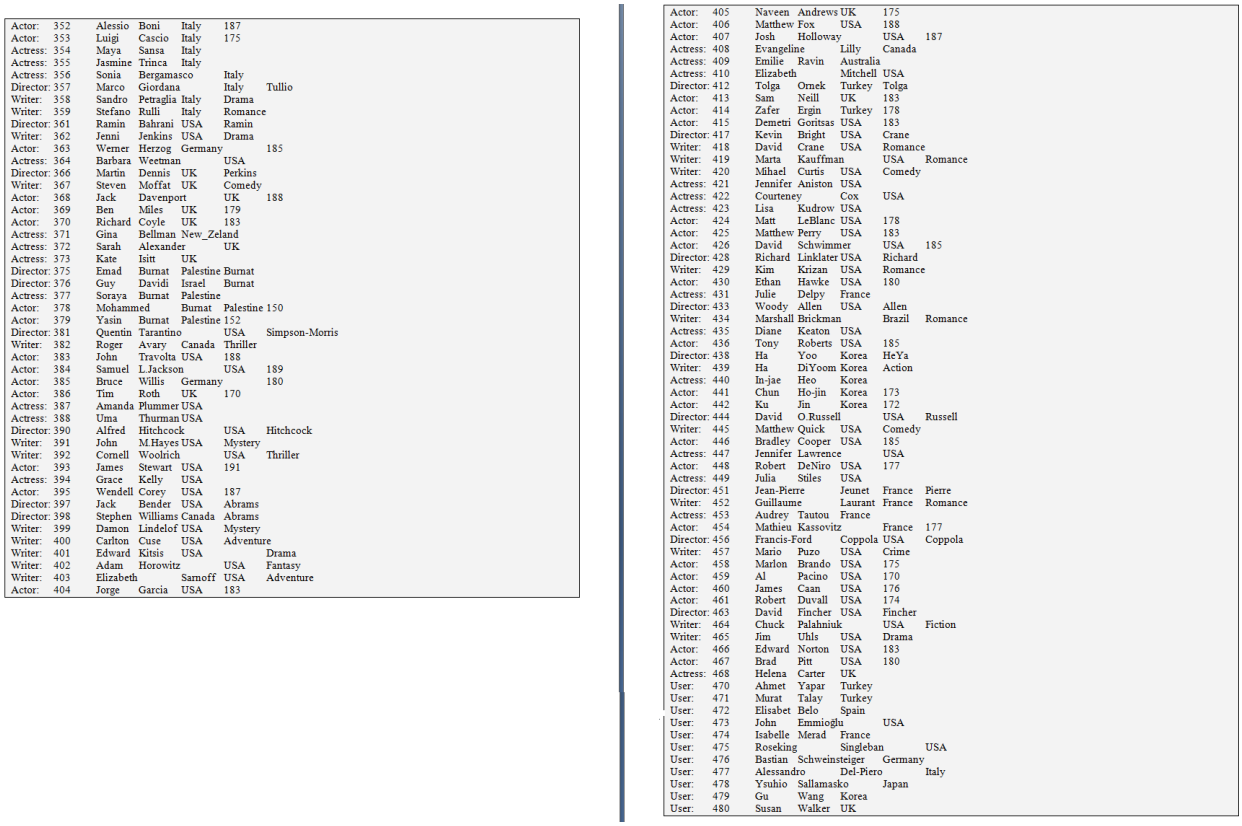
3.1.3 Films File

Since there are four different film types in this system, there are also four different record samples in this file.

FeatureFilm: *<tab> <ID> <tab> <TITLE> <tab> <LANGUAGE>* <tab> <DIRECTOR1_ID,...,DIRECTORn_ID>
<tab> <LENGTH> <tab> <COUNTRY> <tab> <PERFORMER1_ID,...,PERFORMERn_ID>
<tab> <GENRE1,...,GENREn> <tab> <RELEASE_DATE> <tab> <WRITER1_ID,...,WRITERn_ID>
<tab> <BUDGET>

ShortFilm: *<tab> <ID> <tab> <TITLE> <tab> <LANGUAGE>* <tab> <DIRECTOR1_ID,...,DIRECTORn_ID>
<tab> <LENGTH> <tab> <COUNTRY> <tab> <PERFORMER1_ID,...,PERFORMERn_ID>
<tab> <GENRE1,...,GENREn> <tab> <RELEASE_DATE> <tab> <WRITER1_ID,...,WRITERn_ID>

Documentary: *<tab> <ID> <tab> <TITLE> <tab> <LANGUAGE>* <tab> <DIRECTOR1_ID,...,DIRECTORn_ID>
<tab> <LENGTH> <tab> <COUNTRY> <tab> <PERFORMER1_ID,...,PERFORMERn_ID>
<tab> <RELEASE_DATE>



Actor:	352	Alessio	Boni	Italy	187
Actor:	353	Luigi	Cascio	Italy	175
Actress:	354	Maya	Sansa	Italy	
Actress:	355	Jasmine	Trinca	Italy	
Actress:	356	Sonia	Bergamasco	Italy	
Director:	357	Marco	Giordana	Italy	Tullio
Writer:	358	Sandro	Petraglia	Italy	Drama
Writer:	359	Stefano	Rulli	Italy	Romance
Director:	361	Ramin	Bahrani	USA	Ramin
Writer:	362	Jenni	Jenkins	USA	Drama
Actor:	363	Werner	Herzog	Germany	185
Actress:	364	Barbara	Weetman	USA	
Director:	366	Martin	Dennis	UK	Perkins
Writer:	367	Steven	Moffat	UK	Comedy
Actor:	368	Jack	Davenport	UK	188
Actor:	369	Ben	Miles	UK	179
Actor:	370	Richard	Coyle	UK	183
Actress:	371	Gina	Bellman	New_Zeland	
Actress:	372	Sarah	Alexander	UK	
Actress:	373	Kate	Isitt	UK	
Director:	375	Emad	Burnat	Palestine	Burnat
Director:	376	Guy	Davidi	Israel	Burnat
Actress:	377	Soraya	Burnat	Palestine	
Actor:	378	Mohammed	Burnat	Palestine	150
Actor:	379	Yasin	Burnat	Palestine	152
Director:	381	Quentin	Tarantino	USA	Simpson-Morris
Writer:	382	Roger	Avary	Canada	Thriller
Actor:	383	John	Travolta	USA	188
Actor:	384	Samuel	L.Jackson	USA	189
Actor:	385	Bruce	Willis	Germany	180
Actor:	386	Tim	Roth	UK	170
Actress:	387	Amanda	Plummer	USA	
Actress:	388	Uma	Thurman	USA	
Director:	390	Alfred	Hitchcock	USA	Hitchcock
Writer:	391	John	M.Hayes	USA	Mystery
Writer:	392	Cornell	Woolrich	USA	Thriller
Actor:	393	James	Stewart	USA	191
Actress:	394	Grace	Kelly	USA	
Actor:	395	Wendell	Corey	USA	187
Director:	397	Jack	Bender	USA	Abrams
Director:	398	Stephen	Williams	Canada	Abrams
Writer:	399	Damon	Lindelof	USA	Mystery
Writer:	400	Carlton	Cuse	USA	Adventure
Writer:	401	Edward	Kitsis	USA	Drama
Writer:	402	Adam	Horowitz	USA	Fantasy
Writer:	403	Elizabeth	Sarnoff	USA	Adventure
Actor:	404	Jorge	Garcia	USA	183

Actor:	405	Naveen	Andrews	UK	175
Actor:	406	Matthew	Fox	USA	188
Actor:	407	Josh	Holloway	USA	187
Actress:	408	Evangeline	Lilly	Canada	
Actress:	409	Emilie	Ravin	Australia	
Actress:	410	Elizabeth	Mitchell	USA	
Director:	412	Tolga	Ornek	Turkey	Tolga
Actor:	413	Sam	Neill	UK	183
Actor:	414	Zafer	Ergin	Turkey	178
Actor:	415	Demetri	Goritsas	USA	183
Director:	417	Kevin	Bright	USA	Crane
Writer:	418	David	Crane	USA	Romance
Writer:	419	Marta	Kauffman	USA	Romance
Writer:	420	Mihael	Curtis	USA	Comedy
Actress:	421	Jennifer	Aniston	USA	
Actress:	422	Courteney	Cox	USA	
Actress:	423	Lisa	Kudrow	USA	
Actor:	424	Matt	LeBlanc	USA	178
Actor:	425	Matthew	Perry	USA	183
Actor:	426	David	Schwimmer	USA	185
Director:	428	Richard	Linklater	USA	Richard
Writer:	429	Kim	Krizan	USA	Romance
Actor:	430	Ethan	Hawke	USA	180
Actress:	431	Julie	Delpy	France	
Director:	433	Woody	Allen	USA	Allen
Writer:	434	Marshall	Brickman	Brazil	Romance
Actress:	435	Diane	Keaton	USA	
Actor:	436	Tony	Roberts	USA	185
Director:	438	Ha	Yoo	Korea	HeYa
Writer:	439	Ha	DiYoom	Korea	Action
Actress:	440	In-jae	Heo	Korea	
Actor:	441	Chun	Ho-jin	Korea	173
Actor:	442	Ku	Jin	Korea	172
Director:	444	David	O.Russell	USA	Russell
Writer:	445	Matthew	Quick	USA	Comedy
Actor:	446	Bradley	Cooper	USA	185
Actress:	447	Jennifer	Lawrence	USA	
Actor:	448	Robert	DeNiro	USA	177
Actress:	449	Julia	Stiles	USA	
Director:	451	Jean-Pierre	Jeunet	France	Pierre
Writer:	452	Guillaume	Laurant	France	Romance
Actress:	453	Audrey	Tautou	France	
Actor:	454	Mathieu	Kassovitz	France	177
Director:	456	Francis-Ford	Coppola	USA	Coppola
Writer:	457	Mario	Puzo	USA	Crime
Actor:	458	Marlon	Brando	USA	175
Actor:	459	Al	Pacino	USA	170
Actor:	460	James	Caan	USA	176
Actor:	461	Robert	Duvall	USA	174
Director:	463	David	Fincher	USA	Fincher
Writer:	464	Chuck	Palahniuk	USA	Fiction
Writer:	465	Jim	Uhls	USA	Drama
Actor:	466	Edward	Norton	USA	183
Actor:	467	Brad	Pitt	USA	180
Actress:	468	Helena	Carter	UK	
User:	470	Ahmet	Yapar	Turkey	
User:	471	Murat	Talay	Turkey	
User:	472	Elisabet	Belo	Spain	
User:	473	John	Emmioglu	USA	
User:	474	Isabelle	Merad	France	
User:	475	Roseking	Singleban	USA	
User:	476	Bastian	Schweinsteiger	Germany	
User:	477	Alessandro	Del-Piero	Italy	
User:	478	Ysuhio	Sallamasko	Japan	
User:	479	Gu	Wang	Korea	
User:	480	Susan	Walker	UK	

Figure 2: A sample people file.

FeatureFilm:	100	La_meglio_gioventu		Italian	357	180	Italy	352,353,354,355,356		Drama,Romance	22.06.2003		358,359	240000				
ShortFilm:	101	Plastic_Bag	English	361	18	USA	363,364	Drama	07.09.2009		362							
TVSeries:	102	Coupling	English	366	30	UK	368,369,370,371,372,373		Comedy	367	12.05.2000	14.06.2004	4	28				
Documentary:	103	5_Broken_Cameras		Hebrew	375,376	94	Palestine	377,378,379		20.02.2013								
FeatureFilm:	104	Pulp_Fiction	English	381	154	USA	383,384,385,386,387,388		Crime,Thriller		14.04.1994	382	8000000					
FeatureFilm:	105	Rear_Window	English	390	112	USA	393,394,395		Mystery,Thriller		01.04.1956	391,392	1000000					
TVSeries:	106	Lost	English	397,398	42	USA	404,405,406,407,408,409,410		Adventure,Drama,Fantasy		399,400,401,402,403		22.09.2004	23.05.2010	6	121		
Documentary:	107	Gelibolu	Turkish	412	114	Turkey	413,414,415		18.03.2005									
TVSeries:	108	Friends	English	417	22	USA	421,422,423,424,425,426		Comedy,Romance		418,419,420		22.09.1994	06.05.2004	10	236		
FeatureFilm:	109	Before_Sunrise	English	428	105	USA	430,431		Drama,Romance		19.05.1995	429	2500000					
FeatureFilm:	110	Annie_Hall	English	433	93	USA	435,436		Comedy,Drama,Romance		20.04.1977	434	4000000					
FeatureFilm:	111	Biyeolhan_Geori	Korean	438	141	South_Korea	440,441,442		Action,Crime,Thriller		15.06.2006	439	4700000					
FeatureFilm:	112	Silver_Linings_Playbook	English	444	122	USA	446,447,448,449		Comedy,Drama,Romance		04.01.2013	445	21000000					
FeatureFilm:	113	Amelie	French	451	122	France	453,454		Comedy,Romance		25.04.2001	452	10000000					
FeatureFilm:	114	The_Godfather	English	456	175	USA	458,459,460,461		Crime,Drama		05.10.1973	457	6000000					

Figure 3: A sample films file.

TVSeries: <*tab*> <*ID*> <*tab*> <*TITLE*> <*tab*> <*LANGUAGE*> <tab> <DIRECTOR1_ID,...,DIRECTORn_ID> <tab> <LENGTH> <tab> <COUNTRY> <tab> <PERFORMER1_ID,...,PERFORMERn_ID> <tab> <GENRE1,...,GENREn> <tab> <WRITER1_ID,...,WRITERn_ID> <tab> <START_DATE> <tab> <END_DATE> <tab> <SEASONS> <tab> <EPISODES>

A sample films file is shown in Figure 3.

3.1.4 Commands File

All data input files will be processed according to the commands which will be given in a commands file. The command file contains 12 types of commands whose definitions and formats (in parenthesis) are given below.

1. A user can rate a film so that film will be saved to his/her rate list. Rating score must be between 1 and 10 integers.
   (***RATE*** <*tab*> <*USER_ID*> <*tab*> <*FILM_ID*> <*tab*> <*RATING_POINT*>)

```
RATE    470     113     9
ADD     FEATUREFILM     115     Fight_Club      English 463     139     USA     466,467,468     Drama   10.12.1999      464,465 63000000
VIEWFILM        115
RATE    470     115     9
RATE    470     108     8
LIST    USER    470     RATES
EDIT    RATE    470     115     10
REMOVE          RATE    470     108
LIST    USER    470     RATES
LIST    FILM    SERIES
LIST    FILMS   BY      COUNTRY         USA
LIST    FEATUREFILMS    BEFORE  1995
LIST    FEATUREFILMS    AFTER   1995
LIST    FILMS   BY      RATE    DEGREE
LIST    ARTISTS         FROM    USA
```

Figure 4: A sample commands file.

2. It's possible to add a new Feature Film to the system.
(***ADD*** <tab> ***FEATUREFILM*** <tab> <*ID*> <tab> <*TITLE*> <tab> <*LANGUAGE*> <tab> <DIRECTOR1_ID,...,DIRECTO
<tab> <LENGTH> <tab> <COUNTRY> <tab> <PERFORMER1_ID,...,PERFORMERn_ID>
<tab> <GENRE1,...,GENREn> <tab> <RELEASE_DATE> <tab> <WRITER1_ID,...,WRITERn_ID>
<tab> <BUDGET>)

3. Details of a film are displayed by using below command.
(***VIEWFILM*** <tab> <*FILM_ID*>)

4. A user can list all films which he/she rated so far.
(***LIST*** <tab> ***USER*** <tab> <*USER_ID*> <tab> ***RATES***)

5. A user can edit a film which he/she rated before.
(***EDIT*** <tab> ***RATE*** <tab> <*USER_ID*> <tab> <*FILM_ID*> <tab> <*NEW_RATING_POINT*>)

6. A user can remove one of his/her rated film.
(***REMOVE*** <tab> ***RATE*** <tab> <*USER_ID*> <tab> <*FILM_ID*>)

7. List all the TV Series in the system.
(***LIST*** <tab> ***FILM*** <tab> ***SERIES***)

8. List all the films from a specified country
(***LIST*** <tab> ***FILM*** <tab> ***BY*** <tab> ***COUNTRY*** <tab> <*COUNTRY*>)

9. List all the films released before a specified year
(***LIST*** <tab> ***FEATUREFILMS*** <tab> ***BEFORE*** <tab> <*YEAR*>)

10. List all the films released after a specified year
(***LIST*** <tab> ***FEATUREFILMS*** <tab> ***AFTER*** <tab> <*YEAR*>)

11. List all the films in descending order and categorized according to film rating degrees.
(***LIST*** <tab> ***FILMS*** <tab> ***BY*** <tab> ***RATE*** <tab> ***DEGREE***)

12. List all the artists from a specified country and display in categorized order
(***LIST*** <tab> ***ARTISTS*** <tab> ***FROM*** <tab> <*COUNTRY*>)

A sample commands file is shown in Figure 4.

3.1.5 Output File

The output of the commands will be printed to the specified output file. Each command's output will include the command itself **as read from the command file** and the result (error message if necessary) of its execution. The general format of the output file is shown below:

<COMMAND>
<NEW_LINE>
<RESULT>
<NEW_LINE>
<——>

Detailed format of <RESULT> (mentioned above in the general format) output for each command type is given below (WS represents Whitespace).

1. *Film rated successfully*
   Film type:<WS> <FILM_TYPE>
   Film title:<WS> <TITLE>

   If there is not any user or film with specified ID the <RESULT> should be as follows:

   Command Failed
   User ID:<WS> <USER_ID>
   Film ID:<WS> <FILM_ID>

   If the specified film was already rated by the given user, then there should be a warning message as follows:

   This film was earlier rated

2. *FeatureFilm added successfully*
   Film ID:<WS> <FILM_ID>
   Film title:<WS> <TITLE>

   If there is already a film with specified <FILM_ID> or if there is not any specified director, writer or performer in the system, the <RESULT> should be as follows:

   Command Failed
   Film ID:<WS> <FILM_ID>
   Film title:<WS> <TITLE>

3. If specified film is Feature Film or Short Film the result will be as follows:

   <TITLE> <WS> (<RELEASE_DATE>)
   <GENRE>
   Writers:<WS> <NAME> <WS> <SURNAME>
   Directors:<WS> <NAME> <WS> <SURNAME>
   Stars:<WS> <NAME> <WS> <SURNAME>
   <RATINGS> /10 from <VOTE_COUNT> users

   If specified film is Documentary; since a documentary doesn't have writers and genre in the system, the result will be as follows:

   <TITLE> <WS> (<RELEASE_DATE>)
   Directors:<WS> <NAME> <WS> <SURNAME>

Stars:<WS> <NAME> <WS> <SURNAME>
<RATINGS> /10 from <VOTE_COUNT> users

If specified film is TV Series the result will be as follows:

<TITLE> <WS> (<START_DATE> - <END_DATE>)
<SEASONS> <WS> seasons,<WS> <EPISODES> <WS> episodes
<GENRE>
Writers:<WS> <NAME> <WS> <SURNAME>
Directors:<WS> <NAME> <WS> <SURNAME>
Stars:<WS> <NAME> <WS> <SURNAME>
<RATINGS> /10 from <VOTE_COUNT> users

If there is not any film with specified <FILM_ID> the <RESULT> should be as follows:

Command Failed
Film ID:<WS> <FILM_ID>

If there is not any rating votes for that film, then below warning message should be printed:

Awaiting for votes

4. *<TITLE> : <WS> <RATING_SCORE>*

   If there is not any ratings of the specified user, then a warning message will be printed to the output file as follows:

   There is not any ratings so far

   If there is not any user with specified <USER_ID> the <RESULT> should be as follows:

   Command Failed
   User ID:<WS> <USER_ID>

5. *New ratings done successfully*
   Film title:<WS> <TITLE>
   Your rating:<WS> <NEW_RATING_SCORE>

   If there is not any user or film with specified IDs and if the user has no rating score for the specified film, then the <RESULT> should be as follows:

   Command Failed
   User ID:<WS> <USER_ID>
   Film ID:<WS> <FILM_ID>

6. *Your film rating was removed successfully*
   Film title:<WS> <TITLE>

   If there is not any user or film with specified IDs and if the user has no rating score for the specified film, then the <RESULT> should be as follows:

   Command Failed

User ID:<WS> <USER_ID>
Film ID:<WS> <FILM_ID>

7. *<TITLE> <WS> (<START_DATE> - <END_DATE>)*
   <SEASONS> <WS> seasons and <WS> <EPISODES> <WS> episodes

   If there is not any TV Series in the system, then a warning message will be printed to the output file as follows:

   No result

8. *Film title:<WS> <TITLE>*
   <LENGTH> <WS> min
   Language:<WS> <LANGUAGE>

   If there is not any film for specified country in the system, then a warning message will be printed to the output file as follows:

   No result

9. *Film title:<TITLE> <WS> (<RELEASE_DATE>)*
   <LENGTH> <WS> min
   Language:<WS> <LANGUAGE>

   If there is not any film released before specified date in the system, then a warning message will be printed to the output file as follows:

   No result

10. *Film title:<TITLE> <WS> (<RELEASE_DATE>)*
    <LENGTH> <WS> min
    Language:<WS> <LANGUAGE>

    If there is not any film released after specified date in the system, then a warning message will be printed to the output file as follows:

    No result

11. FeatureFilm:
    <TITLE> <WS> (<RELEASE_DATE>) Ratings: <WS> <RATINGS> /10 from <VOTE_COUNT> users

    ShortFilm:
    <TITLE> <WS> (<RELEASE_DATE>) Ratings: <WS> <RATINGS> /10 from <VOTE_COUNT> users

    Documentary:
    <TITLE> <WS> (<RELEASE_DATE>) Ratings: <WS> <RATINGS> /10 from <VOTE_COUNT> users

    TVSeries:
    <TITLE> <WS> (<START_DATE> - <END_DATE>) Ratings: <WS> <RATINGS> /10 from <VOTE_COUNT> users

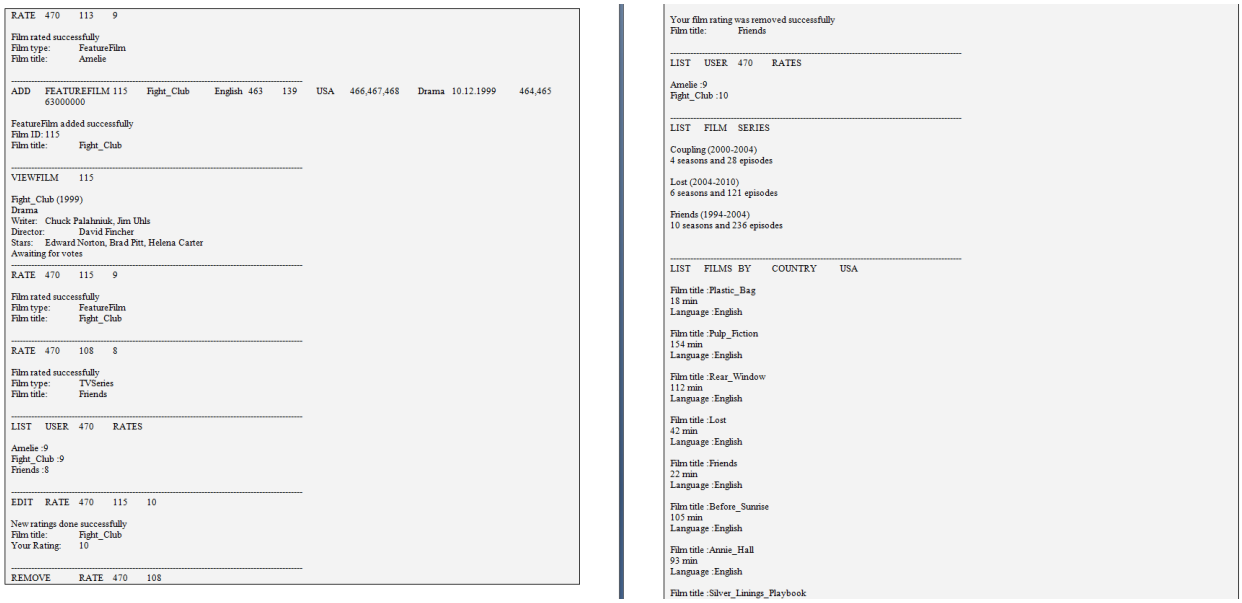


Figure 5: Sample results output file(part-1).

All the results should be printed in descending order.

If there is not any result for a category, then a warning message will be printed to the output file for that category as follows:

No result

12. Directors:
<NAME> <WS> <SURNAME> <WS> <AGENT>

Writers:
<NAME> <WS> <SURNAME> <WS> <TYPE>

Actors:
<NAME> <WS> <SURNAME> <WS> <HEIGHT> <WS> cm

Actresses:
<NAME> <WS> <SURNAME> <WS> <TYPE>

If there is not any result for a category, then a warning message will be printed to the output file for that category as follows:

No result

According to these definitions a sample output file is given in Figure 5, 6.

Number of white spaces does matter as long as there is at least one where appropriate. Further examples which give more details will be provided at the course's FTP site.

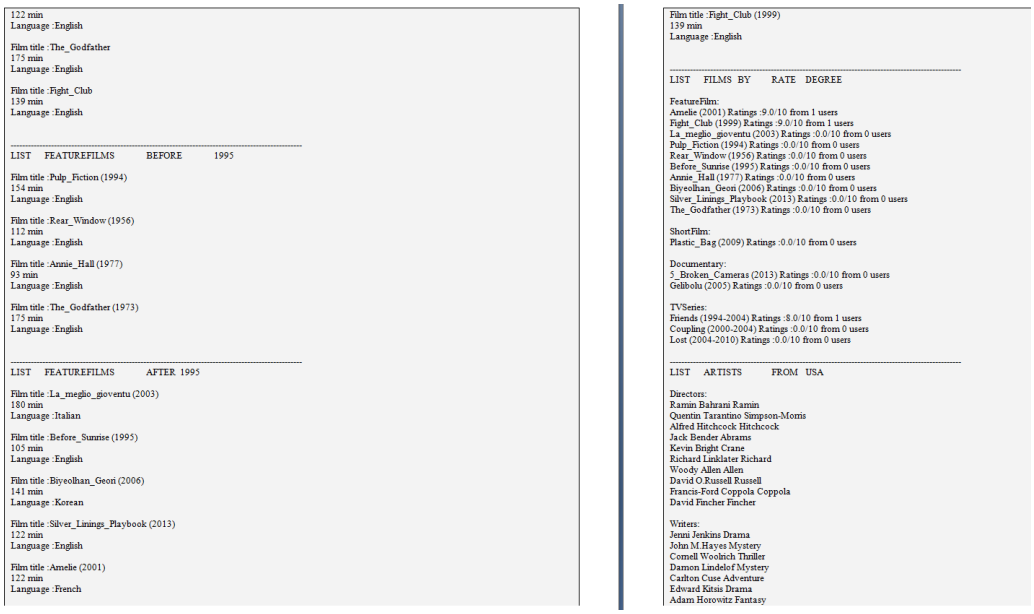


Figure 6: Sample results output file(part-2).

3.2 Report

The structure of report is described below:

- **Cover Page**
- **Class Diagram and Solution**, describe details of your solution, stating its advantages and disadvantages technically. Draw class diagram. Show attributes and method names of each class in your diagram.
- **Comments**, give feedback about problem, problem description, and solution constraints.
- **References**, give the references you used throughout your work at the end of your report.

3.3 Constraints

You should obey these constraints. Otherwise your experiment will not be evaluated.

1. The methods' and attributes' names should be satisfied the most common naming conventions in Java.
2. You should model entities of the system with classes.
3. You are responsible for a correct model design of the movies database system. Your design should be accurate.
4. You should use inheritance mechanism and correct access modifiers.
5. All the input files and output file will be taken as command line arguments.
6. **Take care about the output file format explained above. Your experiments will be evaluated automatically, so your output files should be comply exactly the same with the advisor's one.**

3.4 Submission

The submissions will be accepted only by `http://submit.cs.hacettepe.edu.tr`. The submit should be done until 24/05/2013 17:00 for both Wednesday and Friday section. Submission format is as follows:

<**student_id**>**.zip**(example: 21521512.zip)
[**src**]
–Main.java
–*.java
–*.zip
[**report**]
–report.pdf
–*.jpg or *.jpeg

4 Grading

Your Experiment results will be evaluated with the following rules. They are fixed. They will never be changed. Hence obey these rules. Before you contest your grade re-evaluate yourself with these rules.

- The execution of program have 75 points.
- The submit format have 1 points.
- The report have 24 points: 1 point for **Cover Page**, 20 points for **Class Diagram and Solution**, 2 points for **Comments** and 1 point for **References**. Your final report grade will be calculated as $(Execution * Report)/75$ and added to your execution grade for final score.

5 Notes

These notes are very important please read them and obey them.

- **No submissions except Submission System of Computer Engineering Department will be accepted. Do not send your experiment with emails.**
- **Do not miss the deadline. No submissions will be accepted after 24.05.2013 17:00.**
- Respect your adviser office hours: Wednesday, 10:00-12:00 (RA A. ÇAĞLAYAN).
- For every question or problem please contact with your adviser RA Ali ÇAĞLAYAN. Do not contact with teacher of experiment directly.
- You can send email to your adviser's email alicaglayan@cs.hacettepe.edu.tr.
- **You can ask your questions via BBM104 news group and you are responsible for the group.**
- **The assignment must be original, INDIVIDUAL work. Duplicate or very similar assignments are both going to be punished.**
- Use understandable names for your classes, attributes and procedures.
- Obey general software engineering principles.
- Save all your work until the experiment is graded

References

[1] http://docs.oracle.com/javase/tutorial/java/IandI/override.html

[2] http://docs.oracle.com/javase/tutorial/java/IandI/abstract.html

[3] http://en.wikipedia.org/wiki/Class_diagram

[4] http://en.wikipedia.org/wiki/Naming_convention_%28programming%29#Java

[5] http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html