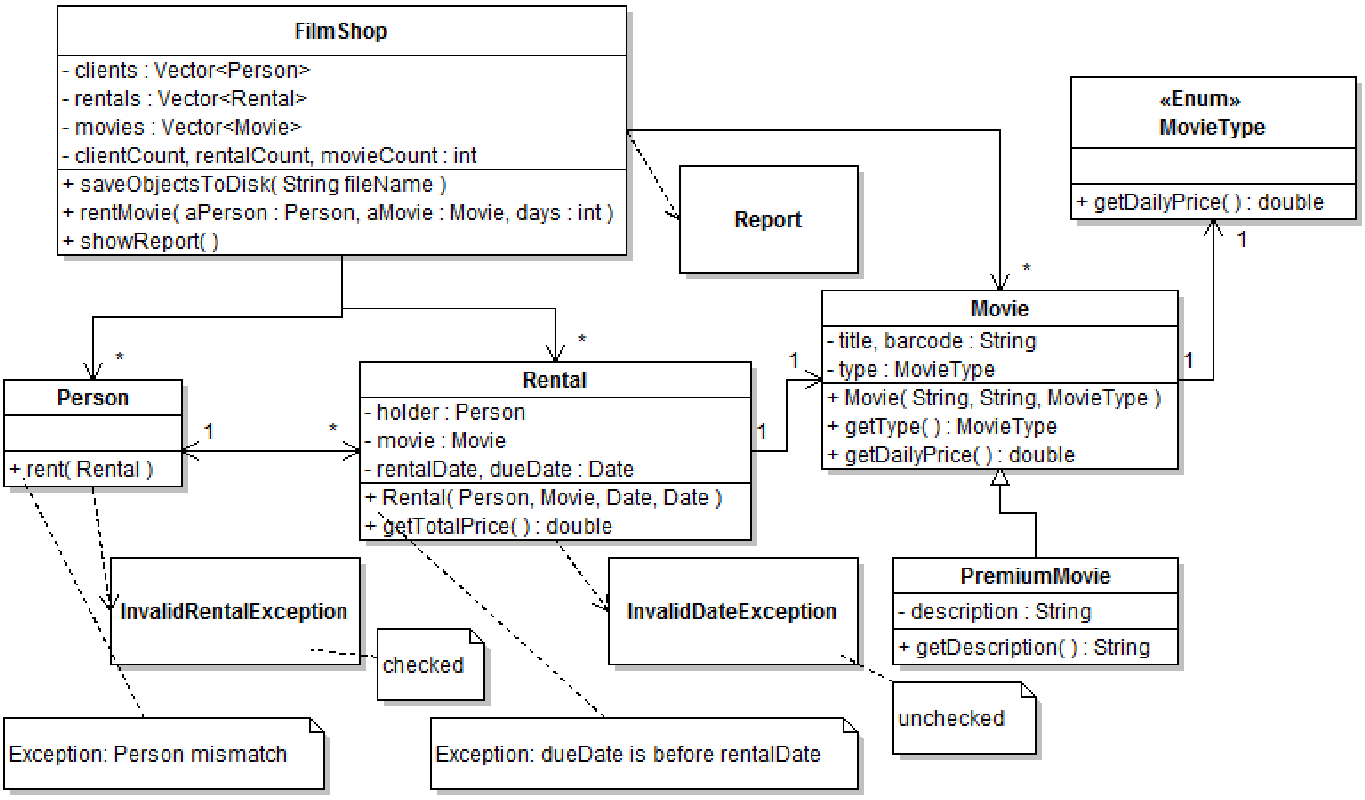
BLM2562 OBJECT ORIENTED PROG., Section 1&2, FINAL EXAM (15/06/2017)

Duration:	90 mins.			**Score:**					**Student Nr:**	**Signature:**
Grading:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Name, Surname:**	
	10	10	10	10	15	15	30			

QUESTIONS



FilmShop
- clients : Vector<Person>
- rentals : Vector<Rental>
- movies : Vector<Movie>
- clientCount, rentalCount, movieCount : int
+ saveObjectsToDisk(String fileName)
+ rentMovie(aPerson : Person, aMovie : Movie, days : int)
+ showReport( )
Report
«Enum»
MovieType
+ getDailyPrice( ) : double
1
*
Movie
- title, barcode : String
- type : MovieType
+ Movie(String, String, MovieType)
+ getType( ) : MovieType
+ getDailyPrice( ) : double
1
1
*
Person
+ rent(Rental)
1
*
*
Rental
- holder : Person
- movie : Movie
- rentalDate, dueDate : Date
+ Rental(Person, Movie, Date, Date)
+ getTotalPrice( ) : double
1
PremiumMovie
- description : String
+ getDescription( ) : String
InvalidRentalException
checked
InvalidDateException
unchecked
Exception: Person mismatch
Exception: dueDate is before rentalDate

Answer these questions according to the UML class schema given above. You may need to extract hidden information from the schema and add necessary code while answering the questions. You don’t have to consider inconsistencies other than the ones related with the exceptions in the class schema.

Question 1: Write the source code of enum MovieType. There are three movie types: Regular, New release and Children’s. The daily rental price of these movie types are TL 1.5, 2.0 and 1.0, respectively.

Question 2: Write the source code of class PremiumMovie. Movies having extended features such as Blu-Ray disc, 3D viewing, etc. are considered premium and their daily rental price are multiplied by 1.5. The description field tells what the extended feature is.

Question 3: Write the source codes of classes InvalidRentalException and InvalidDateException.

Question 4: Write the source code of the constructor method of class Rental. This method throws an InvalidDateException if the due date is earlier than the rental date.

Question 5: Write the source code of the rentMovie method of class FilmShop. This method must call the rent method of class Person. The Person.rent method throws an InvalidRentalException if that person is not the same person as the rental’s holder. You don’t need to write the body of Person.rent method.

Question 6: Write the source code of saveObjectsToDisk method of class FilmShop.

Question 7: Write the source code of showReport method of class FilmShop and the entire Report class. The showReport method creates an instance of the Report class in a new thread. Code the report class so that it lists the regular movies first and then the premium movies. Regular movies’ titles will be listed but premium movies’ titles and descriptions will be listed.

BLM2562 OBJECT ORIENTED PROG., Section 1&2, FINAL EXAM (15/06/2017)

Question 1: Write the source code of enum MovieType.

```
public enum MovieType implements java.io.Serializable {
    REGULAR(1.50), NEW_RELEASE(2.0), CHILDRENS(1.0);
    private double dailyPrice;
    private MovieType( double dailyPrice ) { this.dailyPrice = dailyPrice; }
    public double getDailyPrice() { return dailyPrice; }
}
```

Question 2: Write the source code of class PremiumMovie.

```
public class PremiumMovie extends Movie {
    private String description;

    public PremiumMovie(String title, String barcode, MovieType type, String description) {
        super(title, barcode, type);
        this.description = description;
    }
    public String getDescription() {
        return description;
    }
    public double getDailyPrice( ) {
        return super.getDailyPrice() * 1.5;     //OR: return getType().getDailyPrice()* 1.5;
    }
}
```

Question 3: Write the source codes of classes InvalidRentalException and InvalidDateException.

```
public class InvalidRentalException extends java.io.IOException {
    private static final long serialVersionUID = 1L; //Notlandırılmayacak
    public InvalidRentalException(String msg) { super(msg); }
}
@SuppressWarnings("serial") //Notlandırılmayacak
public class InvalidDateException extends RuntimeException {
    public InvalidDateException(String msg) { super(msg); }
}
```

Question 4: Write the source code of the constructor method of class Rental

```
    public Rental(Person holder, Movie movie, Date rentalDate, Date dueDate) {
        this.holder = holder;                   this.movie = movie;
        this.rentalDate = rentalDate;           this.dueDate = dueDate;
        if( dueDate.before(rentalDate) )
            throw new InvalidDateException("dueDate is "+dueDate
                    +" but rentalDate is "+rentalDate
                    +" for movie "+movie.getTitle()+
                    " and for person "+holder.getName());
    }
```

Question 5: Write the source code of the rentMovie method of class FilmShop.

```
    public void rentMovie( Person aPerson, Movie aMovie, int days ) {
        Date rentalDate = new Date();
        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DAY_OF_MONTH, days);
        Date dueDate = cal.getTime();
        Rental aRental = new Rental(aPerson, aMovie, rentalDate, dueDate);
        try {
            aPerson.rent(aRental);
            rentals.add(aRental);
        }
        catch (InvalidRentalException e) {
            e.printStackTrace();
        }
    }
```

Question 6: Write the source code of saveObjectsToDisk method of class FilmShop.

```
    public void saveObjectsToDisk( String fileName ) {
        try {
            ObjectOutputStream stream = new ObjectOutputStream(
                    new FileOutputStream(fileName));
            stream.writeObject(clients);
            stream.writeObject(rentals);
            stream.writeObject(movies);
            stream.close();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
```

Question 7: Write the source code of showReport method of class FilmShop and the entire Report class.

```
    public void showReport() {
        Thread t = new Thread(new Report(movies));
        t.start();
    }

import java.util.*;
public class Report implements Runnable {
    private Vector<Movie> movies;

    public Report(Vector<Movie> movies) {
        this.movies = movies;
    }
    public void run() {
        for( Movie movie : movies )
            if( !(movie instanceof PremiumMovie) )
                System.out.println( movie.getTitle() );
        for( Movie movie : movies )
            if( movie instanceof PremiumMovie )
                System.out.println( movie.getTitle() +
                        ((PremiumMovie) movie).getDescription());
    }
}
```