

ITIS Belluzzi a.s. 2013/14	<b>Verifica di Informatica</b> <b>29/10/13</b>	Classe: 4Ai	Nome e cognome:
-------------------------------	---	----------------	-----------------

## La videoteca

Si dispone di un file "videoteca.txt" organizzato in modo da avere in successione una riga con una stringa che rappresenta il titolo di un video, a seguire una riga con un intero che rappresenta la sua durata espressa in secondi e una riga con una stringa che rappresenta il genere (ad esempio: comico, giallo, musicale, ecc.), e poi così via per un certo numero di video non noto a priori.

Si intende realizzare una applicazione GUI destinata a gestire i dati della videoteca, secondo le consuete modalità operative si dispone della classe VideotecaGUIApp destinata al Control e VideotecaGUIView destinata alla gestione del form di interfaccia. In questo contesto si richiede:

- (1 punto) la definizione di una classe destinata a gestire in Model i dati di un singolo video (definire nome della classe e attributi, i relativi metodi getter e setter li diamo per scontati)
- (1 punto) la definizione di una classe destinata a gestire in Model i dati della intera videoteca (definire nome e attributi)
- (1 punto) la definizione di un attributo della classe VideotecaGUIApp destinato a un oggetto con i dati della videoteca

Inoltre si richiede a scelta tra i seguenti:

- (3 punti) *sviluppare* un metodo costruttore per la classe relativa alla videoteca che riceve come parametro un oggetto di classe File corrispondente al file con i dati, al termine dell'esecuzione del metodo l'oggetto costruito contiene tutti i dati letti dal file
- (2 punti) *sviluppare* un metodo per la classe relativa alla videoteca che, data una stringa indicante un genere, ritorna un oggetto con i dati del primo video trovato (se esiste) con quel genere
- (3 punti) si suppone che l'interfaccia abbia un textfield destinato all'inserimento di un genere e un bottone destinato all'avvio della ricerca. *Sviluppare* un metodo della classe control che verrà invocato dal metodo handler dell'evento click sul bottone, il metodo provvede alla ricerca del dato e alla visualizzazione del risultato. Definire la firma dei metodi di vista che si ritengono necessari per gli input e output opportuni.

## Svolgimento

### Quesito Uno

la definizione di una classe destinata a gestire in Model i dati di un singolo video (definire nome della classe e attributi, i relativi metodi getter e setter li diamo per scontati)

```
public class Video {
    String titolo;
    int durata; //secondi
    String genere;

    public Video(String titolo, int durata, String genere) {
        this.titolo = titolo;
        this.durata = durata;
        this.genere = genere;
    }
}
```

### Quesito Due

la definizione di una classe destinata a gestire in Model i dati della intera videoteca (definire nome e attributi)

```
public class Videoteca {
    Video archivio[];
    int numeroVideo=0;
    // . . .
}
```

### Quesito Tre

la definizione di un attributo della classe VideotecaGUIApp destinato a un oggetto con i dati della videoteca

```
public class VideotecaGUIApp extends SingleFrameApplication {
```

```
Videoteca video;
//. . .
}
```

### Quesito Quattro

sviluppare un metodo costruttore per la classe relativa alla videoteca che riceve come parametro un oggetto di classe File corrispondente al file con i dati, al termine dell'esecuzione del metodo l'oggetto costruito contiene tutti i dati letti dal file

```
public class Videoteca {
// . . .
    public Videoteca(File f) {
        archivio=new Video[200]; //meglio ancora definire una costante
        try{
            LettoreFileVideo lettore=new LettoreFileVideo(f);
            numeroVideo=0;
            while (numeroVideo<archivio.length && lettore.hasNext() ){
                archivio[numeroVideo]=lettore.leggiDatiVideo(new Video());
                if (archivio[numeroVideo]!=null){
                    numeroVideo++;
                }
            }
        }
        catch(Exception e){

        }
    }
// . . .
} // fine class Videoteca
```

```
public class LettoreFileVideo {
    Scanner lettore;

    public LettoreFileVideo(File f) throws Exception {
        lettore=new Scanner(f);
    }

    public Video leggiDatiVideo(Video v){
        if (!lettore.hasNext()){
            return null;
        }
        String titolo=lettore.nextLine();
        if (!lettore.hasNext()){
            return null;
        }
        int durata=lettore.nextInt();
        lettore.nextLine();
        if (!lettore.hasNext()){
            return null;
        }
        String genere=lettore.nextLine();
        v.setTitolo(titolo);
        v.setDurata(durata);
        v.setGenere(genere);
        return v;
    }

    public boolean hasNext(){
        return lettore.hasNext();
    }
}
```

### Quesito Cinque

sviluppare un metodo per la classe relativa alla videoteca che, data una stringa indicante un genere, ritorna un oggetto con i dati del primo video trovato (se esiste) con quel genere

```
public class Videoteca {
// . . .
    public Video cercaGenere(String genere) {
        for (int i=0;i<numeroVideo;i++){
            if (genere.equalsIgnoreCase(archivio[i].getGenere())){
                return archivio[i];
            }
        }
        return null;
    }
// . . .
} // fine class Videoteca
```

### Quesito Sei

si suppone che l'interfaccia abbia un textfield destinato all'inserimento di un genere e un bottone destinato all'avvio della ricerca. Sviluppare un metodo della classe control che verrà invocato dal metodo handler dell'evento click sul bottone, il metodo provvede alla ricerca del dato e alla visualizzazione del risultato. Definire la firma dei metodi di vista che si ritengono necessari per gli input e output opportuni

```
public class VideotecaGUIApp extends SingleFrameApplication {
    VideotecaGUIView vista;
    Videoteca video;
//. . .
    public void cercaGenere() {
        String genere= vista.getGenereDaCercare();
        Video v=video.cercaGenere(genere);
        if (v!=null){
            vista.mostraRisultato(v.toString());
        }
        else {
            vista.mostraStato("Genere non trovato");
        }
    }
//. . .
} //fine class VideotecaGUIApp

public class VideotecaGUIView extends FrameView {
//. . .

    public String getGenereDaCercare() {
        // . . .
    }

    public void mostraRisultato(String s) {
        // . . .
    }

    public void mostraStato(String s) {
        // . . .
    }
} //fine class VideotecaGUIView
```