

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ - ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ ΣΤΙΣ ΕΠΙΧΕΙΡΗΣΕΙΣ
Α' Εξαμήνου Μεταπτυχιακού Προγράμματος του τμήματος Εφαρμοσμένης Πληροφορικής
Κατεύθυνση Επιχειρηματική Πληροφορική

Project 3: Ομαδοποίηση αντικειμένων (χρήση ArrayList) – Συσχέτιση κλάσεων

Μια αλυσίδα κινηματογράφων αποφάσισε να δημιουργήσει μια εφαρμογή για τη διαχείριση των στοιχείων των ταινιών τις οποίες θα προβάλει και την εκτύπωση διαφόρων λιστών, όπως για παράδειγμα λίστα ταινιών που θα προβληθούν, λίστα ταινιών που θα προβληθούν ένα συγκεκριμένο μήνα και λίστα ταινιών που είναι υποψήφιας για oscar. Στην παρούσα έκδοση η εφαρμογή αποτελείται από δύο κλάσεις: την κλάση **Film** για την αναπαράσταση των ταινιών και την κλάση **FilmProgram** για την αναπαράσταση της λίστας των ταινιών και τη διαχείρισή της.

Η κλάση με όνομα **Film** (όνομα αρχείου Film.java) χρησιμοποιείται για την αναπαράσταση μιας κινηματογραφικής ταινίας και περιλαμβάνει:

- 1) πεδία με όνομα title (String), director (String), playingTime (int), candidacy (boolean) και showDate (String) για την αποθήκευση του τίτλου της ταινίας, του ονόματος του σκηνοθέτη, της διάρκειας της ταινίας σε λεπτά, της πληροφορίας αν η συγκεκριμένη ταινία είναι υποψήφια για όσκαρ ή όχι και της ημερομηνίας πρώτης προβολής στη μορφή HH/MM/EEEE αντίστοιχα (δεν χρειάζεται να γίνει έλεγχος για την ορθότητα της ημερομηνίας).
- 2) ένα κατασκευαστή που θα δέχεται μέσω παραμέτρων τιμές για τα πεδία title, director, playingTime και showDate, ενώ θα θέτει στο πεδίο candidacy την τιμή false.
- 3) ένα κατασκευαστή που θα δέχεται μέσω παραμέτρου τιμή μόνο για το πεδίο title και θα αρχικοποιεί κατάλληλα τα υπόλοιπα πεδία.
- 4) μεθόδους μετάλλαξης (setters) για όλα τα πεδία.
- 5) μεθόδους πρόσβασης (getters) για όλα τα πεδία.
- 6) μέθοδο print που θα εμφανίζει τα στοιχεία μιας ταινίας ως εξής:

```
<τίτλος-ταινίας>, Director <όνομα σκηνοθέτη>  
<διάρκεια ταινίας> minutes  
<ημερομηνία πρώτης προβολής>  
Candidate for Oscar!
```

το μήνυμα "Candidate for Oscar!" θα εμφανίζεται μόνο στην περίπτωση που η ταινία είναι υποψήφια για όσκαρ.

Η κλάση **FilmProgram** (όνομα αρχείου FilmProgram.java) περιλαμβάνει ένα πεδίο τύπου ArrayList<Film> με όνομα list για την αποθήκευση όλων των ταινιών που πρόκειται να προβληθούν σε μια αλυσίδα κινηματογράφων, δηλαδή την αποθήκευση αναφορών σε αντικείμενα τύπου Film. Επίσης, η κλάση περιλαμβάνει:

- 7) κατασκευαστή που αρχικοποιεί κατάλληλα το πεδίο της κλάσης
- 8) μέθοδο με όνομα storeFilm, η οποία θα δέχεται ένα αντικείμενο τύπου Film και θα το αποθηκεύει στη συλλογή list.
- 9) μέθοδο πρόσβασης, έστω getList, που θα επιστρέφει το πεδίο list.
- 10) μέθοδο με υπογραφή public ArrayList<Film> candidateFilms(), η οποία θα διασχίζει το πεδίο list και θα δημιουργεί μια νέα συλλογή ArrayList<Film> που θα περιλαμβάνει μόνο τα αντικείμενα-ταινίες που είναι υποψήφιας για Όσκαρ, δηλαδή τα αντικείμενα-ταινίες που έχουν στο πεδίο candidacy την τιμή true. Η νέα ArrayList θα επιστρέφεται από τη μέθοδο.
- 11) μέθοδο με υπογραφή public ArrayList<Film> monthFilms(int month), η οποία θα διασχίζει το πεδίο list και θα δημιουργεί μια νέα συλλογή ArrayList<Film> που θα περιλαμβάνει μόνο τις ταινίες που η ημερομηνία πρώτης προβολής τους είναι μέσα στο μήνα month. Η νέα ArrayList θα επιστρέφεται από τη μέθοδο. *Δείτε στο τέλος της εκφώνησης τις υποδείξεις σχετικά με το χειρισμό ενός αλφαριθμητικού (αντιγραφή από την ημερομηνία των 2 χαρακτήρων που αναφέρονται στον μήνα και μετατροπή τους σε ακέραιο).*
- 12) μέθοδο με όνομα filmsMeanTime, η οποία θα υπολογίζει και θα επιστρέφει τη μέση διάρκεια σε λεπτά όλων των ταινιών που υπάρχουν στη συλλογή.
- 13) μέθοδο με όνομα findFilm, η οποία θα δέχεται τον τίτλο μιας ταινίας και θα κάνει αναζήτηση στη συλλογή list. Αν η ταινία βρεθεί στη συλλογή τότε θα καλείται γι' αυτή η μέθοδος print (που έχει

οριστεί στην κλάση Film) προκειμένου να εμφανιστούν όλα τα στοιχεία της. Διαφορετικά, δηλαδή αν δεν βρεθεί στη συλλογή ταινία με αυτό τον τίτλο, θα εμφανίζεται το μήνυμα:

The film <τίτλος-ταινίας> does not belong to the collection

- 14) μέθοδο με υπογραφή `public void showFilms(ArrayList<Film> alist)`, η οποία θα διασχίζει την `ArrayList alist` και θα καλεί για κάθε ταινία τη μέθοδο `print` (που έχει οριστεί στην κλάση `Film`) προκειμένου να εμφανιστούν όλα τα στοιχεία της.

Στη συνέχεια, προσθέστε κλάση με όνομα **Main** και ορίστε σε αυτή μέθοδο **main**, στην οποία:

- Θα δημιουργήσετε ένα αντικείμενο της κλάσης `FilmProgram` για τη δημιουργία και διαχείριση της λίστας των ταινιών που θα προβληθούν σε ένα κινηματογράφο.
- Θα δημιουργούνται 2 τουλάχιστον αντικείμενα ταινίες με τον παρακάτω τρόπο:
Θα δημιουργείται ένα αντικείμενο της κλάσης `Scanner` για την είσοδο των απαραίτητων δεδομένων από το πληκτρολόγιο και τη δημιουργία των ταινιών. Συγκεκριμένα:
 - Θα εμφανίζονται στον χρήστη προτρεπτικά μηνύματα για την είσοδο των δεδομένων των ταινιών από το πληκτρολόγιο (όπως φαίνεται στο παράδειγμα εκτέλεσης)
 - Θα δημιουργείται ένα αντικείμενο της κλάσης `Film` με τα δεδομένα που δόθηκαν
 - Θα καλείται η κατάλληλη μέθοδος μετάλλας του παραπάνω αντικείμενου για την ενημέρωση του πεδίου `candidacy` σχετικά με το αν η ταινία είναι υποψήφια για Oscar ή όχι
 - Θα προστίθεται η ταινία στη λίστα των ταινιών που δημιουργήθηκε στο ερώτημα (a) (αντικείμενο της κλάσης `FilmProgram`).
 - Θα ερωτάται ο χρήστης αν θέλει να συνεχίσει την είσοδο δεδομένων. Τα παραπάνω βήματα θα επαναλαμβάνονται όσο ο χρήστης δηλώνει ότι θέλει να συνεχίσει την εισαγωγή δεδομένων.
Δείτε στο τέλος της εκφώνησης τις υποδείξεις σχετικά με την είσοδο δεδομένων από το πληκτρολόγιο και τον κώδικα που σας δίνεται έτοιμος.
- Θα καλείται η μέθοδος `showFilms` με όρισμα την πλήρη λίστα των ταινιών του προγράμματος (η λίστα επιστρέφεται καλώντας τη μέθοδο `getList` του αντικείμενου της κλάσης `FilmProgram`).
- Θα καλείται η μέθοδος `showFilms` με όρισμα τη λίστα των ταινιών που επιστρέφει η μέθοδος `candidateFilms`.
- Θα καλείται η μέθοδος `showFilms` με όρισμα τη λίστα των ταινιών που επιστρέφει η μέθοδος `monthFilms` για ένα μήνα τον οποίο θα εισάγει ο χρήστης από το πληκτρολόγιο μετά από σχετικό προτρεπτικό μήνυμα που θα εμφανίζεται (να υπάρχει μία τουλάχιστον ταινία που θα προβληθεί αυτό το μήνα).
- Θα καλείται η μέθοδος `filmsMeanTime` και θα εμφανίζεται η τιμή που επιστρέφει με τον εξής τρόπο:
The mean playing time of all the films in the collection is <επιστρεφόμενη-τιμή>.
- Θα καλείται η μέθοδος `findFilm` δύο φορές με όρισμα το όνομα ταινίας που θα δίνει ο χρήστης από το πληκτρολόγιο μετά από σχετικό προτρεπτικό μήνυμα (καλό είναι να κληθεί η μέθοδος μία φορά για ταινία που υπάρχει και μία για ταινία που δεν υπάρχει).

Παρακάτω φαίνεται ένα *παράδειγμα εκτέλεσης*, στο οποίο σημειώνονται με έντονα και κόκκινα τα δεδομένα που εισάγονται από το πληκτρολόγιο. Χρησιμοποιήστε κενές γραμμές και επεξηγηματικά μηνύματα (στη `main`) όπως φαίνεται παρακάτω κατά την εμφάνιση των αποτελεσμάτων.

```
***** QUESTION b: read data for all the films *****
Give title? Mars
Give director? Nick Smith
Give playing time? 125
The film is candidate for Oscar? (true/false) true
Give show date? 25/11/2016
Continue? (y/n) y
Give title? Fiction
Give director? John Papadopoulos
Give playing time? 150
The film is candidate for Oscar? (true/false) false
Give show date? 13/12/2016
Continue? (y/n) n
```

```

***** QUESTION c: show the list of all the films *****
Mars, Director Nick Smith
    125 minutes
    25/11/2016
    Candidate for Oscar!
Fiction, Director John Papadopoulos
    150 minutes
    13/12/2016

***** QUESTION d: show the list of films that are candidate for OSCAR *****
Mars, Director Nick Smith
    125 minutes
    25/11/2016
    Candidate for Oscar!

***** QUESTION e: show the list of films shown on November *****
Give the month to see the program of films? 11
Mars, Director Nick Smith
    125 minutes
    25/11/2016
    Candidate for Oscar!

***** QUESTION f: mean time of all films *****
The mean playing time of all the films in the collection is 137.5

***** QUESTION g: search for films *****
Give the title of the film for searching? Holidays
The film Holidays does not belong to the collection
Give the title of the film for searching? Fiction
Fiction, Director John Papadopoulos
    150 minutes
    13/12/2016

```

Υποδείξεις:

- Για να αντιγράψετε ένα τμήμα ενός αλφαριθμητικού θα χρησιμοποιήσετε τη μέθοδο **substring(δείκτης αρχής, δείκτης τέλους)** στην οποία περνάμε ως πρώτο όρισμα τον δείκτη από τον οποίο θα ξεκινήσει η αντιγραφή (συμπεριλαμβάνεται) και ως δεύτερο όρισμα τον δείκτη στον οποίο θα τελειώσει (δεν συμπεριλαμβάνεται). Για παράδειγμα η κλήση `s1.substring(0,2)` επιστρέφει το υπο-αλφαριθμητικό του `s1` που αποτελείται από τους χαρακτήρες στις θέσεις 0 και 1, ενώ η κλήση `s1.substring(3, 5)` θα επιστρέφει τους χαρακτήρες στις θέσεις 3 και 4 (στην ουσία 4^ο και 5^ο χαρακτήρα).
- Για τη μετατροπή ενός αλφαριθμητικού σε ακέραιο μπορείτε να χρησιμοποιήσετε την μέθοδο **Integer.parseInt(παράμετρος String)**
Για παράδειγμα, αν `s1` είναι μια μεταβλητή τύπου `string`:
`int num = Integer.parseInt(s1);`
- Για να ελέγξετε αν δύο αλφαριθμητικά έχουν την ίδια τιμή δεν θα χρησιμοποιήσετε τον τελεστή `==` αλλά τη μέθοδο, **equals()**. Η μέθοδος `equals()` επιστρέφει `boolean` τιμή - `true` ή `false` ανάλογα αν είναι ίσα ή όχι. Σύνταξη μεθόδου: `s1.equals(s2)`, όπου `s1`, `s2` τα συγκρινόμενα αλφαριθμητικά.
- Για την είσοδο δεδομένων από το πληκτρολόγιο θα χρησιμοποιηθεί ένα αντικείμενο της κλάσης `Scanner`:
`Scanner input = new Scanner(System.in);`
`String s = input.nextLine();`
`int x = input.nextInt();`
`double z = input.nextDouble();`
`input.nextLine();` //clear the buffer
`String s = input.nextLine();`
`boolean flag = input.nextBoolean();`
Προσοχή! Αν μετά από την κλήση μεθόδου για διάβασμα αριθμητικών ή λογικών δεδομένων ακολουθεί διάβασμα αλφαριθμητικών δεδομένων θα πρέπει να φροντίσετε να καθαρίσετε τον `buffer` απομακρύνοντας τον χαρακτήρα αλλαγής γραμμής που έχει μείνει σε αυτόν με μια εντολή: `input.nextLine();`

Κώδικας για το ερώτημα b)

Α' τρόπος: επαναληπτική δομή do{...}while =>

οι εντολές στο σώμα του βρόχου επαναλαμβάνονται όσο η συνθήκη που ελέγχεται στο τέλος κάθε επανάληψης * *while(answer.equals("y"))* * είναι αληθής

```
Scanner input = new Scanner(System.in);
String answer;

do{
    System.out.print("Give title? ");
    String title = input.nextLine();

    //Συμπληρώστε τον κώδικα για τα ερωτήματα i), ii), iii), iv)

    System.out.print("Continue? (y/n) ");
    answer = input.nextLine();
}while (answer.equals("y"));
```

Β' τρόπος: επαναληπτική δομή while ελεγχόμενη από λογική μεταβλητή =>

οι εντολές στο σώμα του βρόχου επαναλαμβάνονται όσο η συνθήκη που ελέγχεται στην αρχή κάθε επανάληψης * *while (more)* ή ισοδύναμα *while (more==true)* * είναι αληθής

```
Scanner input = new Scanner(System.in);

boolean more = true;

while (more){
    System.out.print("Give title? ");
    String title = input.nextLine();

    //Συμπληρώστε τον κώδικα για τα ερωτήματα i), ii), iii), iv)

    System.out.print("Continue? (y/n) ");
    answer = input.nextLine();
    if (answer.equals("n"))
        more = false;
}
```

Γ' τρόπος: επαναληπτική δομή while που τερματίζεται (break) από συνθήκη εντός του βρόχου

οι εντολές στο σώμα του βρόχου επαναλαμβάνονται όσο η συνθήκη που ελέγχεται στην εντολή if στο σώμα του βρόχου είναι ψευδής ή αλλιώς μέχρι η συνθήκη *answer.equals("n")* να γίνει αληθής και να εκτελεστεί η εντολή *break*

```
Scanner input = new Scanner(System.in);

while (true){
    System.out.print("Give title? ");
    String title = input.nextLine();

    //Συμπληρώστε τον κώδικα για τα ερωτήματα i), ii), iii), iv)

    System.out.print("Continue? (y/n) ");
    answer = input.nextLine();
    if (answer.equals("n"))
        break; //break the loop => έξοδος από τον βρόχο
}
```