



Αντικειμενοστρεφής Προγραμματισμός

Εαρινό Εξάμηνο 2020-21 - Β' Εξεταστική

Εισηγητής: Αδαμίδης Παναγιώτης
Ασδρέ Κατερίνα

Πέμπτη 9 Σεπτεμβρίου 2021 (16:30-18:30)

ΟΔΗΓΙΕΣ

- Οι εξεταζόμενοι πρέπει να απαντήσουν σε όλα τα θέματα.
- Διάρκεια εξέτασης **1 ώρα και 30'**
- Μπορείτε να συμβουλευέστε το διδακτικό υλικό του μαθήματος κατά την ώρα της εξέτασης.
- Θα πρέπει να γράψετε τις απαντήσεις των θεμάτων χειρόγραφα σε χαρτί και στη συνέχεια να ψηφιοποιήσετε το γραπτό σας. Κατά την ψηφιοποίηση, στην πρώτη σελίδα των απαντήσεών σας θα πρέπει να τοποθετήσετε και την ακαδημαϊκή σας ταυτότητα. Φροντίστε να μην καλύπτονται οι απαντήσεις σας.
- Κάθε σελίδα του γραπτού σας θα πρέπει να έχει γραμμένο τουλάχιστον ονοματεπώνυμο και αριθμό μητρώου του εξεταζόμενου.
- Το όνομα του αρχείου που θα στείλετε θα πρέπει να είναι της μορφής:
"ΕΠΙΘΕΤΟ_ΟΝΟΜΑ_ΑΜ" στα ελληνικά
- Η παράδοση του αρχείου των απαντήσεων πρέπει να γίνει μέσω του συνδέσμου <http://submit.iee.ihu.gr/> με τα παρακάτω username και password ανά αίθουσα:

Αίθουσα	Username	password
R1	18789	27676
R2	22949	36035
R3	26679	21180
R4	29923	50940
R5	31402	75222

Αίθουσα	Username	password
R6	36095	33697
R7	45009	19693
E5	51637	12936
E6	52514	54419
E7	55912	65308

- **ΠΡΟΣΟΧΗ: Να καταθέσετε το γραπτό στην σωστή αίθουσα**
- Μετά την κατάθεση του αρχείου θα πρέπει να δώσετε τον αριθμό παράδοσης στον επιτηρητή (το γράφετε στο chat).
- Αποχωρείτε από την αίθουσα, αφού σας βεβαιώσει ο επιτηρητής ότι έχει καταγράψει τον αριθμό παράδοσης που του δώσατε.



Αντικειμενοστρεφής Προγραμματισμός

Εαρινό Εξάμηνο 2020-21 - Β' Εξεταστική

Εισηγητής: Αδαμίδης Παναγιώτης
Ασδρέ Κατερίνα

Πέμπτη 9 Σεπτεμβρίου 2021 (16:30-18:30)

ΘΕΜΑΤΑ (Μονός Αρ. Μητρώου)

ΘΕΜΑ 1 (3 μονάδες)

1. (Βαθμός 1) Εξηγείστε γιατί συμφωνείτε ή διαφωνείτε με τις παρακάτω προτάσεις:
 - I. Ένα τμήμα κώδικα finally εκτελείται όταν δεν προκληθεί κάποια εξαίρεση στο αντίστοιχο τμήμα try. *Λάθος. ο κώδικας του finally εκτελείται πάντα.*
 - II. Μπορούμε να διαγράψουμε ένα αντικείμενο με τη λέξη κλειδί delete. *Λάθος. Δεν υπάρχει αυτή η λέξη κλειδί ???.*
 - III. Μια υποκλάση κληρονομεί όλα τα πεδία της υπερκλάσης της ανεξάρτητα από τον προσδιοριστή πρόσβασης που έχουν τα πεδία αυτά. *Λάθος. Δεν κληρονομεί τα private.*
 - IV. Για να υπερφορτώσουμε (overload) μία μέθοδο, η νέα μέθοδος θα πρέπει να έχει το ίδιο όνομα και την ίδια υπογραφή. *Λάθος. Μπορούμε να προσθέσουμε νέες παραμέτρους.*
2. (Βαθμός 1) Επιλέξτε ποια θεωρείτε σωστά (Σωστό: +0.5, Λάθος: -0.3)
 - I. Οι κλάσεις Action, Comedy και Drama επεκτείνουν την κλάση Movie. Ποια από τα παρακάτω είναι σωστά:
 - a. Action[] acMovies = {new Movie(), new Action(), new Drama()}; *Λάθος*
 - b. Movie[] mov1 = new Drama[3];
 - c. Movie[] mov2 = {new Drama(), new Comedy(), new Movie()};
 - d. Comedy[] com = {new Movie(), new Comedy(), new Comedy()};
 - e. Drama[] = {new Drama(), new Drama(), new Movie()};
 - II. Δίνονται οι κλάσεις **A** και **B**, η abstract κλάση **C** και τα interfaces **D** και **E**. Ποια από τα παρακάτω είναι σωστά για τον ορισμό του **F**:

a. class F implements C, D { }	b. interface F implements A { }
c. class F extends E { }	d. class F extends E implements A { }
e. interface F extends E, D { }	f. class F extends C implements D { }
3. (Βαθμός 1) Δίνονται οι παρακάτω κλάσεις:

```
class Movie {  
class Action extends Movie {  
class Comedy extends Movie {
```

Εξηγείστε τι κάνουν οι παρακάτω εντολές με τη σειρά που δίνονται. Αν κάποια/ες είναι λάθος εξηγείστε:

- γιατί είναι λάθος,
- αν είναι λάθος κατά την μεταγλώττιση ή την εκτέλεση και
- αν και πως είναι δυνατό να διορθωθεί.

Κάντε σχόλιο τις εντολές που δεν διορθώνονται (αφού τις κάνετε σχόλιο δεν τις παίρνετε υπόψη σας):

```
1 Movie m = new Movie();  
2 Action act1 = new Action();  
3 Action act2 = new Action();  
4 Comedy com1 = new Comedy();  
5 Comedy com2 = new Action();  
6 act1 = m;  
7 m = act1;  
8 com1 = act1;  
9 act2 = (Action) com1;  
10 act2 = m  
11 com1 = (Comedy) m;  
12 Comedy com2 = com1;
```

ΘΕΜΑ 2 (3 μονάδες)

1. Ταξινομήσετε σε αύξουσα σειρά τον πίνακα: «24, 14, 8, 4, 19, 15, 10» με τις μεθόδους: bubble sort, ταξινόμηση με εισαγωγή (insertion sort), ταξινόμηση με επιλογή (selection sort), ταξινόμηση με συγχώνευση (merge sort) εμφανίζοντας τους αριθμούς μετά από κάθε “πέρασμα”. **ΟΧΙ κώδικας.**
2. Γράψτε στην γλώσσα προγραμματισμού Java, μία αναδρομική μέθοδο για τον υπολογισμό της συνάρτησης Ackerman, η οποία ορίζεται ως εξής:

$$A(m, n) = \begin{cases} n+1 & \text{άν } m=0 \\ A(m-1, 1) & \text{άν } m \neq 0 \text{ και } n=0 \\ A(m-1, A(m, n-1)) & \text{άν } m \neq 0 \text{ και } n \neq 0 \end{cases}$$

ΘΕΜΑ 3 (4 μονάδες)

Ορίστε τη δομή (class ή interface) με όνομα “**Vessel**” η οποία αναπαριστά ένα σκάφος και διαθέτει τα χαρακτηριστικά/ιδιότητες (attributes): όνομα σκάφους (vNname, String), μήκος σκάφους από 10-200m (length, double), και έτος κατασκευής (year, int).

Εάν δοθεί τιμή εκτός ορίων στο μήκος του σκάφους, τότε δημιουργείται η εξαίρεση **WrongVesselLengthException** την οποία πρέπει να ορίσετε και να χειριστείτε. Στον χειρισμό της εξαίρεσης, το μήκος του συγκεκριμένου σκάφους να γίνεται μηδέν.

Οι κλάσεις **“Ferry”** και **“Cargo”** εξειδικεύουν την κλάση **“Vessel”**. Η κλάση **“Ferry”** αντιπροσωπεύει ένα επιβατικό-οχηματαγωγό πλοίο, και κληρονομεί την **“Vessel”** προσθέτοντας τις επιπλέον ιδιότητες/χαρακτηριστικά (attributes): αριθμό επιβατών και αριθμό οχημάτων (vehicleNo). Η κλάση **“Cargo”** αντιπροσωπεύει ένα φορτηγό πλοίο και κληρονομεί την **“Vessel”** προσθέτοντας την επιπλέον ιδιότητα του ωφέλιμου φορτίου (load).

Δημιουργήστε την κλάση **“Port”** η οποία αντιπροσωπεύει ένα λιμένα. Η κλάση **“Port”** διαθέτει τα εξής χαρακτηριστικά: Όνομα λιμένα, και πίνακα σκαφών. Τα σκάφη μπορεί να είναι είτε τύπου **“Ferry”**, είτε **“Cargo”**.

Στην κατάλληλη κλάση (ή τις κατάλληλες κλάσεις) υλοποιήστε την διασύνδεση **“Comparable”** χρησιμοποιώντας το μήκος του σκάφους ως μέτρο σύγκρισης.

Υλοποιήστε τις παρακάτω μεθόδους στις κατάλληλες κλάσεις:

- **compPassengerNo()**: υπολογισμός συνολικού πλήθους επιβατών των ελλιμενισμένων επιβατικών-οχηματαγωγών σκαφών σε ένα λιμένα,
- **compTotalCargo()**: υπολογισμός συνολικού ωφέλιμου φορτίου των ελλιμενισμένων φορτηγών σκαφών.
- **departVessel(<όνομα_σκάφους>)**: αναχώρηση σκάφους από τον λιμένα (Port). Δέχεται ως παράμετρο το όνομα του πλοίου και αν το βρει, τότε διαγράφει το αντικείμενο από την συγκεκριμένη θέση του πίνακα,
- **maxVessel()**: υπολογίζει και επιστρέφει τη θέση (int) του σκάφους του λιμένα (Port) το οποίο έχει το μεγαλύτερο μήκος **χρησιμοποιώντας** την υλοποίηση της διασύνδεσης **“Comparable”**.

Υποδείξεις:

Ορίστε επιπλέον χαρακτηριστικά/μεταβλητές, δομητές, μεθόδους πρόσβασης/μεταβολής και ότι άλλες μεθόδους (πχ toString) κρίνετε απαραίτητες.

- στη μέθοδο main():
 - δημιουργήστε ένα λιμένα (Port) με τουλάχιστον 10 σκάφη (Vessel). Σε κάθε θέση του πίνακα τοποθετείστε ένα αντικείμενο **“Ferry”** ή ένα αντικείμενο **“Cargo”** με πιθανότητα 50%. Όλα τα αντικείμενα δημιουργούνται με τον δομητή ο οποίος αρχικοποιεί όλες τις μεταβλητές και όχι με τον εξ’ορισμού (default) δομητή. Αυτό σημαίνει ότι πριν την δημιουργία κάποιου αντικειμένου θα πρέπει να δίνονται όλα τα στοιχεία του. Οι τιμές των στοιχείων μπορεί να είναι τυχαίες ή να δίνονται από το πληκτρολόγιο (δική σας επιλογή).
 - Εμφανίστε τον συνολικό πλήθος επιβατών των ελλιμενισμένων επιβατικών-οχηματαγωγών σκαφών (Ferry) με την χρήση της μεθόδου **compPassengerNo()**.
 - Εμφανίστε το συνολικό ωφέλιμο φορτίο των ελλιμενισμένων φορτηγών σκαφών (Cargo) με την χρήση της μεθόδου **compTotalCargo()**.
 - Εμφανίστε τα στοιχεία του μεγαλύτερου σκάφους..

Υποδείξεις:

Ορίστε δομητές και μεθόδους set/get. Υπερβείτε την μέθοδο toString της Object.

Προσοχή πρέπει να γίνεται έλεγχος για να μην δοθεί τιμή εκτός ορίων στο μήκος του σκάφους. Σε περίπτωση λάθους (προσπάθεια εισαγωγής τιμής εκτός ορίων) να δημιουργείται/εγείρεται η αντίστοιχη εξαίρεση **WrongVesselLengthException** και στο χειρισμό της εξαίρεσης, το μήκος του σκάφους μηδέν και να εμφανίζεται το μήνυμα "Vessel length is set to zero".