



## Αντικειμενοστρεφής Προγραμματισμός

Εαρινό Εξάμηνο 2020-21 - Α' Εξεταστική

Εισηγητής: Αδαμίδης Παναγιώτης  
Ασδρέ Κατερίνα

Δευτέρα 14 Ιουνίου 2021 (11:30-13:30)

### ΟΔΗΓΙΕΣ (Κ2)

- Οι εξεταζόμενοι πρέπει να απαντήσουν σε όλα τα θέματα.
- Διάρκεια εξέτασης **1 ώρα και 30'**
- Μπορείτε να συμβουλευέστε μόνο έντυπο 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
- 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:
- **2ος κύκλος εξέτασης - 11:30-13:30**

Αίθουσα	Username	password
R1	33818	92439
R2	35982	66965
R3	36095	33697
R4	41274	97749
R5	44391	40071
R6	45009	19693

Αίθουσα	Username	password
R7	45140	88403
R8	47585	87018
E5	51637	12936
E6	52168	15647
E7	52391	83370
E8	52514	54419

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



## ΘΕΜΑΤΑ (Κ2)

### ΘΕΜΑ 1 (3 μονάδες)

1. (Βαθμός 1) Επιλέξτε ποια θεωρείτε σωστά (Σ: 0.5, Λ: -0.2)

I. Οι κλάσεις MountainBike, TandemBike και RoadBike επεκτείνουν την κλάση Bike. Ποια από τα παρακάτω είναι σωστά:

- a. Bike [] bikes = {new Bike(), new TandemBike(), new TandemBike()}; Σωστό
- b. MountainBike [] bike2 = new RoadBike[3]; Λάθος
- c. RoadBike[] bike1 = {new RoadBike(), new TandemBike(), new Bike()}; Λάθος
- d. Bike[] rb = new RoadBike[10]; Σωστό

II. Δίνονται τα interfaces **A** και **B**, η abstract κλάση **C** και οι κλάσεις **D** και **E**. Ποια από τα παρακάτω είναι σωστά για τον ορισμό του **F**:

- a. interface F implements B { } Λάθος
- b. class F implements A { } Σωστό
- c. interface F extends A, B { } Σωστό
- d. class F extends D implements B { } Σωστό
- e. class F extends B, D { } Λάθος
- f. class F implements E Λάθος

2. (Βαθμός 1) Αιτιολογείστε την έξοδο του παρακάτω προγράμματος;

```
class A {
    int i;
    A () { add(1); }
    void add(int v) { i+=v; }
    void print() { System.out.println(i); }
}

class B extends A {
    B() { add(5); }
    void add(int v) { i+=v*3; }
}

public class K2_Q1_3 {
    public static void main(String args[]) {
        bogo(new B());
    }
    static void bogo(A b) {
        b.add(2);
        b.print();
    }
}
```

**3. (Βαθμός 1) Εξηγείστε γιατί** συμφωνείτε ή διαφωνείτε με τις παρακάτω προτάσεις

- I. Ένα αρχείο Java περιλαμβάνει πολλές κλάσεις. Το αρχείο πρέπει να έχει το όνομα της κλάσης που περιέχει την main(). *Λάθος. Ένα όνομα αρχείου περιέχει μια κλάση και έχει για όνομα το όνομα της κλάσης.*
- II. Οι μεταβλητές static αποθηκεύονται στο stack. *Λάθος. Μόνο οι τοπικές μεταβλητές και οι παράμετροι των μεθόδων αποθηκεύονται στο stack.*
- III. Μεταβλητές και μέθοδοι μιας κλάσης που δηλώνονται protected μπορούν να προσπελαστούν από οποιαδήποτε κλάση.
- IV. Η κληρονομικότητα αναπαριστά μία σχέση "έχει-ένα" (has-a). *Ένα είδος κληρονομικότητας το οποίο ονομάζεται "σύνθεση" αναπαριστά τη σχέση "έχει-ένα".*

**ΘΕΜΑ 2** (3 μονάδες)

1. Ταξινομήσετε σε αύξουσα σειρά τον πίνακα: «20, 16, 7, 22, 9, 12, 18» 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, την αναδρομική μέθοδο dispSquares η οποία δέχεται ως παράμετρο έναν ακέραιο αριθμό  $n$  και εμφανίζει τα τετράγωνα των αριθμών από 1 έως  $n$ . Πχ. `dispSquares(5)`  $\longrightarrow$  1 4 9 16 25  
`dispSquares(8)`  $\longrightarrow$  1 4 9 16 25 36 49 64

**ΘΕΜΑ 3** (4 μονάδες)

Ορίστε τη δομή (class ή interface) με όνομα "**Bike**" η οποία αναπαριστά ένα ποδήλατο και διαθέτει το χαρακτηριστικό/ιδιότητα: βασική τιμή (basePrice, int).

Ορίστε τις κλάσεις "**MountainBike**" και "**TandemBike**" οι οποίες κληρονομούν/υλοποιούν την "**Bike**" και αναπαριστούν δύο τύπους ποδηλάτου, ποδήλατο βουνού και ποδήλατο με περισσότερες θέσεις, αντίστοιχα. Η κλάση "**MountainBike**" διαθέτει το χαρακτηριστικό: ύψος σέλας (seatHeight, double), ενώ η κλάση "**TandemBike**" διαθέτει το χαρακτηριστικό: αριθμός θέσεων (seatNo, int).

Εάν δοθεί αρνητική τιμή για την τιμή του ποδηλάτου, τότε δημιουργείται η εξαίρεση **NegativeBikePriceException** την οποία πρέπει να ορίσετε. τον χειρισμό της εξαίρεσης, η τιμή του ποδηλάτου να γίνεται μηδέν.

Ένα κατάστημα ποδηλάτων διαθέτει ποδήλατα και των δύο ειδών. Κάθε ποδήλατο προσφέρεται με ένα πακέτο με επιπλέον εξαρτήματα. Υπάρχουν πέντε(5) διαφορετικά πακέτα και κάθε πακέτο εξαρτημάτων αυξάνει την τιμή του ποδηλάτου κατά ένα ποσοστό από 10% έως 50%. Η μεταβλητή extraEquipment με ακέραια τιμή από 1-5 (στην κατάλληλη κλάση) είναι το είδος του πακέτου επιπλέον εξοπλισμού. Η τελική τιμή του ποδηλάτου υπολογίζεται από τον τύπο:

$$\text{Τελική τιμή} = \text{basePrice} + \text{extraEquipment}/10.0 * \text{basePrice}$$

Ορίστε στην κατάλληλη κλάση (ή κλάσεις) τη μέθοδο finalPrice() η οποία υπολογίζει την τελική τιμή. Τοποθετείστε την μεταβλητή extraEquipment και τη μέθοδο finalPrice() στις κατάλληλες κλάσεις.

Στην κλάση **BikeShop**:

- ορίστε τις μεθόδους:
  - ♦ `averageMountainBikeSeatHeight()`: Δέχεται ως παράμετρο ένα πίνακα με ποδήλατα και των δύο ειδών. Υπολογίζει και επιστρέφει το μέσο όρο του ύψους της σέλας των ποδηλάτων βουνού (**MountainBike**).
  - ♦ `averageBikePrice()`: Δέχεται ως παράμετρο ένα πίνακα με ποδήλατα και των δύο ειδών. Υπολογίζει και επιστρέφει το μέσο όρο των **βασικών** τιμών (`basePrice`) όλων των ποδηλάτων.
  - ♦ `maxTandemBikePrice()`: Δέχεται ως παράμετρο ένα πίνακα με ποδήλατα και των δύο ειδών. Υπολογίζει και επιστρέφει την μεγαλύτερη **τελική** τιμή ποδηλάτου τύπου **TandemBike** (με περισσότερες θέσεις).
- στη μέθοδο `main()`:
  - ♦ δημιουργείστε ένα πίνακα με τουλάχιστον 50 αντικείμενα/ποδήλατα. Σε κάθε θέση του πίνακα τοποθετείστε ένα αντικείμενο "**MountainBike**" ή "**TandemBike**" τυχαία με πιθανότητα 50%. Όλα τα αντικείμενα δημιουργούνται με τον πλήρη δομητή (αυτός 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 τα στοιχεία όλων των ποδηλάτων.
  - ♦ Εμφανίστε την μέγιστη τελική τιμή ποδηλάτου τύπου **TandemBike** (με περισσότερες θέσεις).
  - ♦ Εμφανίστε τον μέσο όρο των βασικών τιμών (`basePrice`) των ποδηλάτων του καταστήματος.

#### Υποδείξεις:

Ορίστε δομητές και μεθόδους `set/get`.

Υπερβείτε την μέθοδο `toString` της `Object`.

Προσοχή πρέπει να γίνεται έλεγχος για να μην δοθεί αρνητική τιμή στην βασική τιμή του ποδηλάτου. Σε περίπτωση λάθους (προσπάθεια εισαγωγής αρνητικής τιμής) να δημιουργείται/εγείρεται η αντίστοιχη εξαίρεση **NegativeBikePriceException** και στο χειρισμό της εξαίρεσης, η βασική τιμή του ποδηλάτου (`basePrice`) να γίνεται μηδέν και να εμφανίζεται το μήνυμα "Base price is set to zero".