

ΑΤΕΙ Πειραιά
Τμήμα Ηλεκτρονικών Υπολογιστικών Συστημάτων

Αντικειμενοστραφής Προγραμματισμός
Β' Εξεταστική εαρινού α.ε. 2009-2010

(διαθέσιμος χρόνος: 2 ώρες)

v1

Θέμα Α (5 μονάδες)

Όλες οι ερωτήσεις έχουν ίση βαθμολογία. Δεν υπάρχει αρνητική βαθμολογία.

Ερώτηση 1: Έστω ο πιο κάτω κώδικας:

```
public int m1(int x) {
    int count=1;
    try {
        count += x;
        count += m2(count++);
    } catch(Exception e) {
        count -= x;
    }
    return count;
}
```

Τι θα επιστρέψει η πιο πάνω μέθοδος αν κληθεί με όρισμα την τιμή 2 – δηλ. m1(2), δεδομένου ότι η κλήση της μεθόδου m2() δημιουργεί ένα Exception;

1. Θα επιστρέψει την τιμή 1
2. Θα επιστρέψει την τιμή 2
3. Θα επιστρέψει την τιμή 3
4. Τίποτε, το πρόγραμμα θα τερματίσει

Ερώτηση 2: Έστω οι πιο κάτω εκφράσεις:

- (α) String s = "Hello Java";
- (β) new String s = "Hello Java";
- (γ) String s = new String("Hello Java");

Ποια από τις πιο πάνω εκφράσεις αποδίδει επιτυχώς την τιμή "Hello Java" στη μεταβλητή s;

1. Η (α) και η (β)
2. Η (β) και η (γ)
3. Η (α) και η (γ)
4. Και οι τρεις

Ερώτηση 3: Έστω ο πιο κάτω κώδικας:

```
class Vehicle {}
class Car extends Vehicle{}

public static void main( String[] args ) {
    Vehicle v = new Car();

    if(v instanceof Vehicle)
        System.out.println( "Vehicle" );

    if(v instanceof Car )
        System.out.println( "Car" );
}
```

Τι θα έχει εκτυπώσει ο κώδικας στην οθόνη μετά το τέλος της εκτέλεσης του;

1. Vehicle
2. Car

3. Vehicle Car
4. Τίποτε από τα παραπάνω

Ερώτηση 4: Έστω ο πιο κάτω κώδικας:

```
String s[] = new String[5];

int i = 0;
while (i<=5) {
    String s[i++] = Integer.toString(i);
}
System.out.println("#"+s[4]+"#");
```

Τι θα έχει εκτυπώσει ο κώδικας στην οθόνη μετά το τέλος της εκτέλεσής του;

1. #3#
2. #4#
3. #5#
4. Τίποτε από τα παραπάνω

Ερώτηση 5: Έστω ο πιο κάτω κώδικας:

```
int j;
for (int i = 0; i < 14; i++) {
    if (i < 10) {
        j = i + 2;
    }
    System.out.println("j:"+j+"i:"+i);
}
```

Τι λάθος υπάρχει στον πιο πάνω κώδικα;

1. Η μεταβλητή j δεν έχει αρχικοποιηθεί
2. Η δήλωση της μεταβλητής i δεν είναι σωστή
3. Η σύνταξη του if δεν είναι σωστή
4. Δεν υπάρχει κάποιο λάθος, ο κώδικας τρέχει κανονικά

Ερώτηση 6: Έστω ο πιο κάτω κώδικας:

```
int values[] = {1,2,3,4,5,6,7,8};

for (int i = 0; i < X; ++i) {
    System.out.println(values[i]);
}
```

Ποια τιμή του X θα έχει ως αποτέλεσμα την εκτύπωση όλων των στοιχείων του πίνακα χωρίς την παρουσίαση κάποιου σφάλματος;

1. X = 7;
2. X = 8;
3. X = 9;
4. Η δομή του κώδικα είναι τέτοια που δεν υπάρχει κάποια τιμή του X που να επιτρέπει την εκτύπωση όλων των στοιχείων του πίνακα.

Ερώτηση 7: Έστω ο πιο κάτω κώδικας:

```
import java.io.*;
class Daddy {
    int methodOne( int a, long b ) throws IOException    { ... }

    float methodTwo( char a, int b ) { ... }
}
public class Boy extends Daddy { }
```

και έστω οι μέθοδοι:

- (α) public static void main(String args[]) { ... }
- (β) int methodOne(int c, long d) throws ArithmeticException { ... }
- (γ) float methodTwo(char c, int d) { ... }

Ποιες από τις πιο πάνω μεθόδους μπορούν να ανήκουν στην κλάση **Boy**;

1. Η (α) και η (β)
2. Η (β) και η (γ)
3. Η (α) και η (γ)
4. Και οι τρεις

v1

Θέμα Β (5 μονάδες)

Να κατασκευάσετε μια βιβλιοθήκη (package) η οποία θα εμπεριέχει τα απαραίτητα αρχεία κλάσεων για την υλοποίηση ενός προγραμματιστικού στοιχείου (control) ηλεκτρονικής ατζέντας (**Agenda**). Η ατζέντα θα οργανώνει συναντήσεις καταχωρώντας τις σε εβδομαδιαία χρονική βάση. Κάθε αντικείμενο ατζέντας τηρεί αντικείμενα της κλάσης **CalendarDay** η οποία μοντελοποιεί μια ημερολογιακή ημέρα συναντήσεων. Η κλάση **CalendarDay** θα έχει τις ιδιότητες: Ημέρα της εβδομάδας ως αλφαριθμητικό κι ένα πίνακα αλφαριθμητικών **dayMeetings** δέκα (10) θέσεων για την εισαγωγή συναντήσεων με διάφορα θέματα. Θα υπάρχει κατάλληλος κατασκευαστής και επίσης μια μέθοδος **CreateDayMeeting** η οποία θα εισάγει μια νέα συνάντηση στην ημερολογιακή ημέρα στην αμέσως επόμενη κενή θέση του πίνακα **dayMeetings**. Μια ιδιότητα της κλάσης **Agenda** θα είναι ένας πίνακας **daysMatrix** επτά (7) αντικειμένων **CalendarDay**. Κατά την αρχικοποίηση (κατασκευαστές) ενός αντικειμένου τύπου **Agenda** θα πρέπει να συμπληρώνεται και ο εν λόγω πίνακας με νέα αντικείμενα τύπου **CalendarDay**. Επιπλέον ιδιότητα της κλάσης **Agenda** θα είναι μια λογική (boolean) ένδειξη **readOnly** η οποία θα υποδηλώνει εάν τα περιεχόμενα της ατζέντας υπόκεινται σε επεξεργασία. Η κλάση **Agenda** θα έχει τρεις (3) κατασκευαστές. Έναν default, έναν με όρισμα έναν πίνακα από αντικείμενα **CalendarDay**, και έναν τρίτο με ορίσματα έναν πίνακα από αντικείμενα **CalendarDay** και μια λογική (boolean) μεταβλητή η οποία υποδηλώνει αν ο χρήστης της εφαρμογής είναι συνδεδεμένος (logged in) ή όχι. Η επεξεργασία των δεδομένων επιτρέπεται μόνο και μόνο αν ο χρήστης είναι συνδεδεμένος. Η ιδιότητα **readOnly** πρέπει να αρχικοποιείται μόνο μέσα από τον κώδικα των κατασκευαστών. Η κλάση **Agenda** θα περιέχει μια μέθοδο **CreateCalendarMeeting** η οποία θα εισάγει μια νέα συνάντηση σε κάποια ημερολογιακή ημέρα και θα παίρνει τα παρακάτω ορίσματα: θέση στον πίνακα **daysMatrix** όπου θα γίνει η εισαγωγή και θέμα συνάντησης. Εντός της παραπάνω μεθόδου θα γίνεται κλήση της αντίστοιχης μεθόδου εισαγωγής συνάντησης (**CreateDayMeeting**) του κατάλληλου αντικειμένου **CalendarDay** από τον πίνακα **daysMatrix**. Η μέθοδος **CreateCalendarMeeting** πρέπει να ελέγχει αν επιτρέπεται η επεξεργασία και επομένως η εισαγωγή συναντήσεων. Αν δεν επιτρέπεται τότε πρέπει να προκαλείται εξαίρεση τύπου **InvalidOperationException**. Η διαχείριση της εξαίρεσης δε θα γίνεται εντός της μεθόδου. Απλώς η υπογραφή της θα δηλώνει ότι η συγκεκριμένη μέθοδος προκαλεί τη συγκεκριμένη εξαίρεση.