

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΕΡΙΟΔΟΣ ΙΟΥΝΙΟΥ 2015

ΟΝΤΟΚΕΝΤΡΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΔΙΔΑΣΚΟΝΤΕΣ: Μακρής Χρήστος, Ρήγκου Μαρία, Χατζηλυγερούδης Ιωάννης

ΜΕΡΟΣ Α-ΓΛΩΣΣΑ JAVA

ΘΕΜΑ 1 (20)

Ποιοι από τους παρακάτω ισχυρισμούς είναι σωστοί και ποιοι λάθος (δώστε σύντομη εξήγηση για τους λανθασμένους);

- (1) Η μέθοδος `init()` ενός applet εκτελείται κάθε φορά που εκτελείται και η μέθοδος `paint()` στη διάρκεια ζωής ενός applet.
- (2) Ένα stream εισόδου ή εξόδου είναι ένα αρχείο.
- (3) Η μέθοδος `run()` είναι μέθοδος της Runnable, αλλά όχι της Thread.
- (4) Ένα νήμα είναι ένα αυτόνομο πρόγραμμα μέσα σ' ένα άλλο πρόγραμμα.
- (5) Για είσοδο από πληκτρολόγιο δεν χρειάζεται να δημιουργήσουμε stream εισόδου στο πρόγραμμα.
- (6) Ένα νήμα από 'εκτελούμενο' (running) γίνεται 'αναμένον' (waiting) όταν δεν είναι διαθέσιμος ένας πόρος I/O.
- (7) Δύο ταυτόχρονες διεργασίες αντιστοιχούν σε δύο διαφορετικά νήματα.
- (8) Η μέθοδος `main()` ενός applet είναι αυτή που καθορίζει το περιεχόμενό του.
- (9) Η μέθοδος `run()` είναι αυτή που ενεργοποιεί ένα νήμα.
- (10) Η μέθοδος `close()` χρησιμοποιείται για το κλείσιμο ενός (αντικειμένου) αρχείου.

ΘΕΜΑ 2 (34)

Δίνεται ο παρακάτω κώδικας Java.

(α) Καταγράψτε τους δημιουργούς των κλάσεων (εμφανείς και αφανείς). [4]

(β) Εξηγείστε συνοπτικά, αλλά πλήρως, ποιο είναι το αποτέλεσμα (εσωτερικό και εξωτερικό, είτε εκτυπώνεται είτε όχι) της κάθε γραμμής κώδικα 1-14 της μεθόδου `main` της κλάσης 'RunWindow'. [22]

(γ) Αν η 'protected int size = 2;' στην XWindow γινόταν 'private int size = 2;' σε ποιες από τις προτάσεις (1)-(14) θα είχαμε πρόβλημα και γιατί; [4]

(δ) Σχεδιάστε την ιεραρχία (κλάσεων και στιγμιοτύπων) που θα δημιουργήσει η Java με την εκτέλεση της `main`. [4]

<pre> class Window { public int size = 1; public static int count = 0; public Window () {this.size = 3; setCount ();} public static void setCount () {count++;} public void setSize(int x) {size = x;}} class XWindow extends Window { protected int size = 2; public XWindow (int x) {this.size = --x; } public void setSize1(int x) {size -= x; } public void setSize2(int x) {super.setSize(x++); System.out.println (x);} public void printSize1() {System.out.println (size); } </pre>	<pre> public void printSize2() {System.out.println (super.size);} } public class RunWindow { public static void main (){ (1) XWindow w1 = new XWindow (6); (2) XWindow w2 = new XWindow (4); (3) Window w = new Window (); (4) System.out.println (w1.size); (5) System.out.println (Window.count); (6) Window.setCount(); (7) System.out.println (w.count); (8) w1.printSize1 (); (9) w1.printSize2 (); (10) w1.setSize1 (10); (11) w1.setSize2 (7); (12) w1.printSize2 (); (13) w2.setSize2 (5); (14) w2.printSize1 (); }} </pre>
--	---

ΘΕΜΑ 3 (26)

Δίνεται ο παραπάνω κώδικας Java.

(α) Να τον επεκτείνετε ορίζοντας μια νέα κλάση MXWindow, ως υποκλάση της XWindow, η οποία περιέχει δύο νέες μεταβλητές width και height που αναπαριστούν το πλάτος και το ύψος του παραθύρου καθώς και τον απαιτούμενο δημιουργό. [6]

(β) Να εισάγετε στη νέα κλάση δύο μεθόδους area και circumference οι οποίες υπολογίζουν το εμβαδόν και το μήκος της περιμέτρου ενός παραθύρου αντίστοιχα. [6]

(γ) Να δημιουργήσετε και να εισάγετε όπου πρέπει μια μέθοδο κλάσης biggerwindow που να συγκρίνει δύο παράθυρα ως προς το εμβαδόν τους και να επιστρέφει αυτό που έχει το μεγαλύτερο εμβαδό. [6]

(δ) Να γράψετε τις απαιτούμενες γραμμές κώδικα στην RunWindow που να δημιουργούν δύο στιγμιότυπα της MXWindow, mw1 και mw2, με διαστάσεις (πλάτοςxύψος) 2.8x4.0 και 3.2x3.4, να υπολογίζουν και εκτυπώνουν τα εμβαδά και τα μήκη περιμέτρων των δύο παραθύρων και να συγκρίνουν τα δύο παράθυρα ως προς το εμβαδόν τους (επιστρέφοντας αυτό με το μεγαλύτερο) και να αναθέτουν τα αποτελέσματα σε μεταβλητές. [6]

(ε) Να σχεδιάσετε την ιεραρχία (κλάσεων και στιγμιοτύπων) που θα δημιουργήσει τώρα η Java με την εκτέλεση της main. [2]

ΜΕΡΟΣ Β-ΓΛΩΣΣΑ C++

ΘΕΜΑ 4 (35)

Εξηγήστε τι θα εκτυπωθεί στην έξοδο.

```
#include <stdio.h>
#include <iostream>
class Class1
{
private:
    int a;
public:
    Class1 ():a(9){};
    Class1(int f)
        {a=(int)f};
    void operator ()(int x, int y)
    {
        if (x<y) a=x+y;
        else a=y/x;
    };
    void operator ()(int x, int y, int z)
    {
        if (x>y) a=(x+y)/z; else a=(y+x)*x;
    };
    void print ()
    {
        std::cout<<a<<"\n";
    };
};

int main (void)
{
    Class1 c11;
    c11.print();
    c11(0,1);
    c11.print();
    c11(1,2,4);
    c11.print();
    c11=Class1(3);
    c11.print();
    return 0;
}
```

ΘΕΜΑ 5 (35)

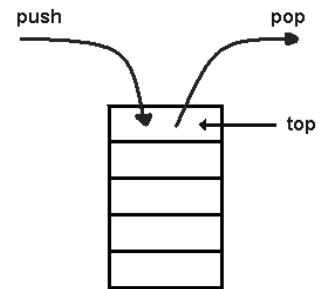
Υλοποιήστε μια αφηρημένη κλάση που αναπαριστά Πανεπιστημιακό Μάθημα και δύο παράγωγες κλάσεις της που αναπαριστούν αντίστοιχα Εργαστηριακό Μάθημα και Θεωρητικό Μάθημα. Το Πανεπιστημιακό μάθημα έχει πεδία περιγραφή, διδάσκοντα και φοιτητές που το παρακολουθούν (έστω N). Το Εργαστηριακό Μάθημα έχει επιπλέον σαν πεδίο ένα πίνακα δύο διαστάσεων NX4 πραγματικών τιμών που είναι οι βαθμοί των N φοιτητών σε 4 εργαστηριακές ασκήσεις, ενώ το Θεωρητικό μάθημα έχει ένα πίνακα μίας διαστάσεως N κελιών με τους βαθμούς των φοιτητών στη γραπτή εξέταση του μαθήματος.

- Φτιάξτε δημιουργούς με τα απαραίτητα ορίσματα σε κάθε μία.
- Δημιουργήστε πολυμορφική μέθοδο "mesos_oros" που επιστρέφει το μέσο όρο βαθμολογίας
- Υπερφορτώστε τον τελεστή << ώστε να εμφανίζει τα στοιχεία ενός Πανεπιστημιακού Μαθήματος (περιγραφή, διδάσκοντα και μέσο όρο βαθμολογίας)
- Υλοποιήστε main συνάρτηση στην οποία να αξιοποιείτε την πολυμορφική μέθοδο mesos_oros και να χρησιμοποιεί τον τελεστή <<.

ΘΕΜΑ 6 (30)

Επιθυμούμε να δημιουργήσουμε μια τάξη Stack (Στοιίβα), αντικειμένων γενικού τύπου T (συνεπώς πρέπει να χρησιμοποιήσετε templates). Μια στοίβα υποστηρίζει τις πράξεις push (ένθεση στην κορυφή της στοίβας) και pop (αφαίρεση από την κορυφή της στοίβας). Υλοποιήστε μία Stack χρησιμοποιώντας **μία διπλά διασυνδεδεμένη λίστα**. Ζητείται να γράψετε τους ορισμούς όλων των μεθόδων της Stack (κατασκευαστής, καταστροφέας, push, pop). Όταν η στοίβα είναι άδεια, η pop να επιστρέφει 0. Ο κατασκευαστής να δημιουργεί μια κενή στοίβα.

Δεν επιτρέπεται να χρησιμοποιήσετε καμία βιβλιοθήκη της C++. Χρησιμοποιήστε δυναμική δέσμευση μνήμης για τη δημιουργία των απαιτούμενων νέων κόμβων. Φροντίστε να μην υπάρχουν διαρροές μνήμης.



ΚΑΛΗ ΕΠΙΤΥΧΙΑ!