

SUBIECTE EXAMEN ADMITERE LA MASTER
Domeniul Calculatoare și Tehnologia Informației
Septembrie 2012

Discipline de concurs :

Grupul de discipline 1-4	Grupul de discipline 5-11
1 - Limbaje de Programare	5 - Structura și arhitectura calculatoarelor
2 - Algoritmi	6 - Protocoale de comunicație în Rețele de calculatoare
3 - Calculatoare numerice	7 - Rețele locale de calculatoare
4 - Baze de date	8 - Sisteme de operare
	9 - Inteligența artificială
	10 - Grafică pe calculator
	11 - Managementul proiectelor

Candidații vor avea de rezolvat subiecte de la patru discipline.
Este obligatoriu ca minim 3 discipline să fie alese din Grupul de discipline 1-4.
Pe prima foaie de concurs se vor scrie numele disciplinelor tratate în lucrare.

GRUPUL DE DISCIPLINE 1-4

1 - Limbaje de Programare

2 puncte fiecare subiect

1) Fie următorul segment de cod:

```
public class Oclasa {
    private static final Oclasa INSTANTA = new Oclasa();
    private Oclasa () { ... }
    public static Oclasa getInstanta () { return INSTANTA; }
}
```

- Apelul metodei `getInstanta()` întoarce de fiecare dată aceeași referință
- Apelul metodei `getInstanta()` întoarce de fiecare dată altă referință
- Este incorectă declararea constructorului `private`

Alegeți varianta corectă și justificați răspunsul.

2) Să se completeze cele 3 puncte de suspensie astfel încât apelul metodei `getNo()` să returneze numărul de obiecte instanțiate din clasa `Clasa`. Nu rescrieți clasa `Test`!

```
class Clasa { ... }
class Test {
    public static void main(String[] args) {
        Clasa x = new Clasa();
        System.out.println(x.getNo()); // afișează 1
        Clasa[] y = new Clasa[10];
        for(int i = 0; i < 4; i++)
            y[i] = new Clasa();
        System.out.println(y[0].getNo()); // afișează 5
    }
}
```

3) Ce se va afisa in urma executiei metodei main din clasa TestMasina? De ce?

```

class Vehicul {
    public void conduce() {
        System.out.println("A");
    }
}

class Masina extends Vehicul {
    public void conduce() {
        System.out.println("B");
    }
}

class TestMasina {
    public static void main(String args[]){
        Vehicul v;
        Masina c;
        v = new Vehicul ();
        c = new Masina ();
        v.conduce ();
        c.conduce ();
        v = c;
        v.conduce ();
    }
}
    
```

4) Sa se defineasca o clasa Punct pentru o pereche de puncte (x,y) cu coordonate numere intregi. Clasa va contine definitia a doi constructori (unul fara parametri - initializare cu zero, iar celalalt cu 2 parametri - cate unul pentru fiecare coordonata) si metoda toString() - ce returneaza o reprezentare a unui obiect din clasa curenta. Constructorii se vor defini unul pe baza celui alt!

5) Fie urmatorul fragment de cod, unde clasa Punct se considera definita:

```

ArrayList<Punct> a = new ArrayList<Punct>();
ArrayList<Object> b;
b = a;
b.add(new Punct(10,20));
    
```

Apar erori de compilare sau rulare? De ce?

2 - Algoritmi

Problema 1. (4 puncte) Fie v un vector cu n numere întregi. Algoritmul **suma**, prezentat mai jos, calculează suma elementelor din vector.

```

suma(v, i, j) {
    if i == j
        return v[i]
    else {
        k = (i + j) / 2
        return suma(v, i, k) + suma(v, k+1, j)
    }
}
    
```

Apelul algoritmului este: **suma(v, 1, n)**. Complexitatea temporală a algoritmului **suma** este (alegeți una dintre variante):

a) $\Theta(n \lg(n))$ b) $\Theta(n)$ c) $\Theta(\lg(n))$

Justificați răspunsul calculând complexitatea temporală a algoritmului.

Problema 2. (2 puncte) Dați un exemplu de problemă care este în clasa de complexitate P, dar nu este în clasa de complexitate NP. Justificați răspunsul.

Problema 3. (4 puncte) Dați un exemplu de graf orientat (cu minim 4 și maxim 6 vârfuri) în care arcele pot avea atât costuri pozitive, cât și negative (dar fără a avea cicluri de cost negativ), pentru care algoritmul lui Dijkstra pentru determinarea drumurilor minime nu funcționează corect. Specificați vârful sursă, vârful pentru care algoritmul Dijkstra nu funcționează corect, distanța calculată de către Dijkstra, precum și distanța corectă până la vârful respectiv.

3 - Calculatoare numerice

1. Se consideră execuția a două programe (P1i este programul 1 iar P2i este programul 2). Pentru aceste programe se cere:

P11: lw \$1, 40(\$6)	P21: lw \$5, -16(\$5)
P12: add \$6, \$2, \$2	P22: sw \$5, -16(\$5)
P13: sw \$6, 50(\$t1)	P23: add \$5, \$5, \$5

- Specificați dacă există hazarduri de date în secvențele de cod prezentate
- În cazul în care există hazarduri de date, să se specifice tipul de hazard identificat și locul apariției acestor hazarduri în cadrul programelor

- c. În cazul existenței hazardurilor de date, să se specifice metoda utilizată pentru eliminarea lor precum și versiunea corectă a programelor.

2. Să se prezinte diagrama de funcționare (figurati și semnalele corespunzătoare) și codul Verilog / VHDL pentru un numărator modulo 3.

4 - Baze de date

Fie o tabela PROPRIETATI continand date despre persoane si proprietatile acestora (automobile si imobile):
PROPRIETATI(NumePersoana, MarcaAuto, AnAuto, TipImobil, AnConstructieImobil)

- In cazul unei proiectari corecte folosind modelul Entitate-Asociere numarul de entitati din diagrama este:
a. 1 b. 2 c. 3 d. 4 e. 6
- In cazul unei proiectari corecte folosind modelul Entitate-Asociere numarul de asocieri din diagrama este:
a. 1 b. 2 c. 3 d. 4 e. 6
- Asocierile de la punctul 2 sunt de tipul:
a. Unare unu-unu b. Unare multi-unu c. Unare unu-multi d. Binare multi-multi e. Binare multi-unu
- Numarul de dependente multivalorice netriviiale independente (nu se deduc din altele) care se pot identifica in tabela PROPRIETATI este:
a. 1 b. 2 c. 3 d. 4 e. Nici unul din raspunsurile anterioare nu este corect
- Numarul de dependente functionale netriviiale independente (nu se deduc din altele) care se pot identifica in tabela PROPRIETATI este:
a. 1 b. 2 c. 3 d. 4 e. Nici unul din raspunsurile anterioare nu este corect
- Care din cererile urmatoare este corecta:
a. TRUNCATE PROPRIETATI; b. TRUNCATE TABLE PROPRIETATI; c. TRUNCATE PROPRIETATI TABLE; d. TRUNCATE * FROM PROPRIETATI; e. TRUNCATE FROM PROPRIETATI;
- Care din cererile urmatoare este corecta:
a. DELETE PROPRIETATI; b. DELETE TABLE PROPRIETATI; c. DELETE PROPRIETATI TABLE; d. DELETE * FROM PROPRIETATI; e. DELETE FROM PROPRIETATI;
- UNION se foloseste pentru a lega:
a. Doar cereri INSERT b. Doar cereri UPDATE c. Doar cereri SELECT d. Doar cereri MERGE e. Cereri INSERT, UPDATE si DELETE
- Clauza WITH CHECK OPTION se foloseste:
a. Doar in cereri INSERT b. Doar in cereri UPDATE c. Doar in cereri DELETE d. Atat in cereri INSERT, UPDATE, DELETE cat si alte cereri SQL e. Variantele a, b, c si d sunt toate gresite
- Cererea COMMIT se foloseste in conjunctie:
a. Cu cereri INSERT, UPDATE si DELETE b. Doar cu cereri INSERT c. Doar cu cereri UPDATE d. Doar cu cereri DELETE e. Doar cu cereri SELECT

Un singur raspuns corect la fiecare intrebare. Rezolvarea constă în scrierea pe foaia de concurs a perechilor (numar întrebare, litera răspuns corect, de ex.). Exemplu: 1-a, 2-b, ..., 10-e.

1 punct fiecare raspuns corect.

