

Universitatea Politehnica București  
Facultatea de Automatică și Calculatoare  
Admitere Master  
Varianta 1  
Toate întrebările au un singur răspuns corect.

1. Limbaje de programare

1. După apelul metodei de mai jos, care dintre cei 2 parametri vor avea un conținut modificat?

```
void f(int x, MyClass c) {  
    x = 5;  
    c = new MyClass();  
    c.member = false;  
}
```

- A. x, deoarece are un tip primitiv
- B. nici unul
- C. amandoi
- D. c, deoarece are un tip referință

2. Fie următoarele definiții de clase:

```
class Foo {  
    static int x;  
    Foo() {  
        doSth(); x++;  
    }  
    void doSth() {  
        x++;  
    }  
}  
class Bar extends Foo {  
    Bar() {  
        System.out.println(x);  
    }  
    void doSth() {  
        x += 2;  
    }  
}
```

și apelul: `new Bar()`. Care va fi mesajul afișat?

- A. 2
- B. 1
- C. 3
- D. 0

3. O clasă privată, care nu este internă, poate fi instantiată în următoarele condiții:

- A. nu poate fi instantiată
- B. de cod definit într-o clasă din același fișier sursă
- C. de cod definit într-o clasă din același pachet
- D. nu poate fi definită ca privată

4. Care dintre afirmațiile următoare este falsă:

- A. dacă B și C sunt două clase care extind clasa A, atunci dacă b este o instanță a clasei B, apelul `(C)((A)b).m()` va converti b prin `upcast` la A apoi prin `downcast` la C și va executa metoda `m()` a clasei C, presupunând că există o astfel de metodă
- B. C extinde B și B extinde A. Obiectul x are tipul C. Valoarea lui `"x instanceof A"` este `true`
- C. dacă `String s1 = "Habar"` și `String s2 = "Ha"+"bar"`, atunci `s1.equals(s2)` este `true`
- D. clasa A definește metoda m, clasa B extinde A și nu îl suprascrie pe m. Dacă b este un obiect de tip B atunci apelul `b.m()` este corect și va apela m din clasa A

5. Care tip dintre variantele propuse va produce o eroare de compilare dacă este folosit ca tip întors de metoda `method` din clasa `Test`:

```
class A {}  
class B extends A {}  
class Test {  
    "?" method() {  
        return new ArrayList<B>();  
    }  
}
```

- A. `Iterable<?>`
- B. `Collection<? extends A>`
- C. `ArrayList<A>`
- D. `List<B>`

6. Următoarele definiții/declarații fac parte dintr-un program Java:

```
interface I1 {}  
interface I2 {}  
class C1 implements I1 {}  
class C2 implements I2 {}  
class C3 extends C1 implements I2 {} ...  
C1 obj1 = ...;  
C2 obj2 = ...;  
C3 obj3 = ...;
```

Care dintre următoarele instrucțiuni este corectă:

- A. `obj2 = obj1;`
- B. `obj3 = obj2;`
- C. `I2 c = obj1;`
- D. `I1 b = obj3;`

7. Care dintre afirmațiile următoare este adevărată:

- A. dacă două clase A și B implementează aceeași interfață I, atunci A este o subclasă a lui B sau B este o subclasă a lui A
- B. dacă `int a = ceva` și `double b = altceva`, atunci `((double)a < b)` și `(a < (int)b)` sunt echivalente
- C. C extinde B și suprascrie metoda x din B. b are tipul B. Apelul `b.x(args)` apelează x din C
- D. dacă B este o subclasă a lui A și b este o instanță a lui B, atunci `((B)(A)b == b)` este `true`

8. Alegeți afirmația falsă:

- A. o clasă poate implementa mai multe interfețe
- B. dacă clasa B este o subclasă a clasei A, atunci ea poate avea o metodă m numai dacă clasa A are aceeași metodă
- C. o definiție generică de tipul `<? extends A>` poate fi instantiată cu orice tip care este un subtip al tipului A, dar nu cu un supertip
- D. o clasă neabstractă care implementează o interfață nu trebuie să includă cod pentru metodele interfeței care au fost implementate de o superclasă

## 2. Algoritmi

**1. Consideram problema Q: "Fie  $n$  un numar natural,  $n > 10^{100}$ . Este  $n$  divizibil cu 23?".**

**Problema Q este:**

- A. Semidecidabila
- B. Niciuna din celelalte variante
- C. Nedecidabila
- D. Decidabila

**2. Problema rucsacului, in cazul in care obiectele au valori numere reale iar greutatile lor au valori numere reale pozitive, poate fi rezolvata:**

- A. Atat programare dinamica cat si algoritmul lacom (greedy)
- B. Niciuna din celelalte variante
- C. Folosind un algoritmul lacom (greedy)
- D. Folosind un algoritmul de programare dinamica

**3. Intr-un graf orientat si aciclic cu costuri pozitive, algoritmul optim pentru drumurile minime de sursa unica este:**

- A. Algoritmul lui Dijkstra
- B. Algoritmul lui Johnson
- C. Niciuna din celelalte variante
- D. Algoritmul lui Bellman-Ford

**4. Considerati urmatorul algoritmul:**

```
Alg(v[1..n]){
  if (v[1] == 0)
    return 0
  else if (n == 1)
    return v[1]
  else
    return v[1] * Alg(v[2..n])
}
```

**Timpul de executie pentru Alg este:**

- A.  $\Theta(n * \lg n)$
- B.  $\omega(n * \lg n)$
- C.  $\Theta(n)$
- D.  $O(n)$

**5. Demonstratiile prin inductie structurala sunt bazate pe ideea multimilor care au asociata o relatie de ordine cu proprietatea:**

- A. Nu exista siruri (secvente) infinite descrescatoare cu elemente ordonate din multime
- B. Niciuna din celelalte variante
- C. Nu exista siruri (secvente) infinite crescatoare cu elemente ordonate din multime
- D. Nu exista siruri (secvente) infinite cu elemente din multime

**6. La o anumita problema data la care initial s-a folosit o schema „divide-et-impera” s-a decis ca mai eficient este sa se foloseasca schema programarii dinamice. Aceasta decizie a fost luata deoarece:**

- A. Aparea o blocare intr-un maxim local
- B. Se intra intr-un ciclu infinit
- C. Multe subsolutii se recalculau
- D. Nu exista un criteriu de maxim

**7. Care dintre variantele de mai jos reflecta cea mai buna complexitate care se poate obtine pentru problema identificarii daca un graf  $G(V, E)$  este aciclic:**

- A.  $\Theta(|V|^2)$
- B.  $\Theta(|V| + |E|)$
- C.  $\Theta(|V| * |E|)$
- D.  $\Theta(|V| * \log |V| + |E|)$

**8. Fie Q o problema astfel incat Q apartine clasei P. Ce afirmatie este adevarata:**

- A. Daca Q este NP-completa atunci  $P \neq NP$
- B. Q nu apartine clasei NP
- C. Niciuna din celelalte variante
- D. Q nu poate fi rezolvata folosind algoritmi deterministi polinomiali

### 3. Calculatoare numerice

#### 1. Următoarea secvență de cod reprezintă:

```
i1): mov 20(r3), 30(r4);  
i2): add r3, 1;
```

- A. Un hazard de tipul RAW
- B. Un hazard de tipul WAR
- C. Un hazard de tipul WAW
- D. Un program cu o execuție normală care nu generează nici un tip de hazard

#### 2. Tehnica de bypassing se utilizează la:

- A. Eliminarea unui hazard structural
- B. Eliminarea unui hazard de date la încărcare
- C. Această tehnică nu există
- D. Eliminarea unui hazard de control

#### 3. Următorul cod Verilog descrie:

```
module JK_flipflop(q, q_negat, j, k,  
                 set, reset, clock);  
input j, k, set, reset, clock;  
output q, q_negat;  
reg q;  
always @(posedge clock or posedge set  
        or posedge reset)  
    if (set) #1 q = 1;  
    else if (reset) #1 q = 0;  
    else if (clock)  
        case ()  
            2'b00: #2 q = q;  
            2'b01: #2 q = 0;  
            2'b10: #2 q = 1;  
            2'b11: #2 q = ~q;  
        endcase  
    assign #1 q_negat = ~q;  
endmodule
```

- A. Un bistabil JK cu intrarea SET sincronă cu ceasul și intrarea RESET asincronă cu ceasul, active pe 1 logic
- B. Un bistabil JK cu intrarea SET asincronă cu ceasul și intrarea RESET sincronă cu ceasul, active pe 1 logic
- C. Un bistabil JK cu intrări asincrone de SET și RESET, active pe 1 logic. Intrările sunt asincrone cu ceasul.
- D. Un bistabil JK cu intrări sincrone de SET și RESET, active pe 1 logic. Intrările sunt sincrone cu ceasul

#### 4. Dezavantajul lui RAID 4 constă în faptul că:

- A. Paritatea disk-ului trebuie recalculată la fiecare citire
- B. Paritatea disk-ului trebuie recalculată la fiecare scriere
- C. RAID 4 nu are nici un dezavantaj
- D. Timpul de scriere este mult prea mare

#### 5. O optimizare a metodei WRITE-THROUGH presupune:

- A. Nu există o astfel de tehnică
- B. Folosirea unui dirty bit pentru scrierea în memoria cache și folosirea unui buffer pentru scrierea într-un anumit nivel al ierarhiei de memorie
- C. Folosirea unui buffer de scriere
- D. Folosirea unui dirty bit

#### 6. Tehnica WRITEBACK este folosită pentru:

- A. Scrierea datelor în memoria cache dar și în nivelul următor al ierarhiei de memorie
- B. Scrierea datelor în oricare nivel al ierarhiei de memorie
- C. Scrierea datelor în memoria cache
- D. Nu există o astfel de tehnică

#### 7. Într-o memorie cache cu asociativitate parțială, creșterea gradului de asociativitate conduce la:

- A. Reducerea ratei de eșec dar și reducerea timpului de HIT
- B. Creșterea ratei de eșec și reducerea timpului de HIT
- C. Creșterea ratei de eșec și creșterea timpului de HIT
- D. Reducerea ratei de eșec dar creșterea timpului de HIT

#### 8. Principiul de bază folosit în cadrul ierarhiilor de memorie spune că:

- A. Programele prezintă doar localizare spațială
- B. Programele nu prezintă localizare temporală și localizare spațială
- C. Programele prezintă localizare temporală cât și localizare spațială
- D. Programele prezintă doar localizare temporală

#### 4. Baze de date

1. Fie o relație  $R = ABCDE$  și mulțimea de dependențe funcționale  $F = \{A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E, D \rightarrow A\}$ . Câte chei distincte are  $R$ ?

- A. 4
- B. 3
- C. 2
- D. 1

2. Fie o relație  $R = ABCDE$  și mulțimea de dependențe funcționale  $F = \{A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E, D \rightarrow A\}$ . Atunci  $R$  este în formele normale:

- A. Forma normală 3 (FN3) și Forma normală Boyce Codd (FNBC)
- B. FNBC dar nu și în FN3
- C. Nici în FN3 nici în FNBC
- D. FN3 dar nu și în FNBC

3. Fie tabela  $STUD(CODS, NUME, MEDIE, CODFACULT)$ .

Cererea `SELECT * FROM STUD S WHERE 3 > (SELECT COUNT(*) FROM STUD WHERE MEDIE > S.MEDIE)` returnează:

- A. 3 linii
- B. 2 linii
- C. 2 sau mai multe linii
- D. 3 sau mai multe linii

4. În domeniul gestiunii tranzacțiilor, I din ACID înseamnă:

- A. Izolare
- B. Inamovibilitate
- C. Intervenție
- D. Independență

5. Fie tabela  $STUD(CODSTUD, NUMESTUD, MEDIE, CODFACULT)$ . Care cerere SQL nu returnează eroare:

- A. `DELETE STUD WHERE MEDIE = MAX(MEDIE);`
- B. `UPDATE STUD SET MEDIE = MEDIE - MAX(MEDIE);`
- C. `INSERT INTO STUD VALUES(5, MONICA, 9.56, 2);`
- D. `SELECT * FROM STUD WHERE MEDIE = MEDIE;`

6. Fie o relație  $R = ABCDE$  și mulțimea de dependențe funcționale  $F = \{A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E, D \rightarrow A\}$ . Descompunerea  $(AE, BCD)$  are proprietățile:

- A. Păstrează dependențele și are proprietatea de join fără pierderi
- B. Nu păstrează dependențele și nici nu are proprietatea de join fără pierderi
- C. Nu păstrează dependențele dar are proprietatea de join fără pierderi
- D. Păstrează dependențele dar nu are proprietatea de join fără pierderi

7. Protocolul de blocare în 2 faze garantează serializabilitatea la execuția a:

- A. Două tranzacții
- B. Unei singure tranzacții
- C. Maxim 2 tranzacții
- D. Oricâtor tranzacții

8. Fie o relație  $R = ABCDE$  și mulțimea de dependențe funcționale  $F = \{A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E, D \rightarrow A\}$ . Care dependență nu se poate deduce din  $F$ ?

- A.  $C \rightarrow E$
- B.  $D \rightarrow B$
- C.  $A \rightarrow D$
- D.  $A \rightarrow BCD$

## 8. Sisteme de operare

### 1. Cum se definește noțiunea de livelock?

- A. un blocaj activ de procese realizat cu forme de busy-waiting
- B. un sistem de procese care folosesc pentru sincronizare mutex-uri
- C. o mulțime de procese blocate în care fiecare proces așteaptă producerea unui eveniment care ar putea fi realizat numai prin execuția unui alt proces din această mulțime
- D. un sistem de procese care accesează cel puțin o resursă în regim de excludere mutuală

### 2. Cu ce apel de sistem se asociază noțiunea "copy-on-write"?

- A. socket
- B. open
- C. signal
- D. fork

### 3. Câte procese vor rezulta în urma rulării secvenței de mai jos? Se pornește de la un singur proces. Considerați și procesul inițial. Se presupune că toate apelurile reușesc.

```
fork();  
fork();  
fork();
```

- A. 3
- B. 8
- C. 6
- D. 4

### 4. Fie prototipul de structură de mai jos. Ce primitivă de sincronizare reprezintă?

```
Struct ... {  
    /* list of waiting threads */  
    queue_t waiting_threads_queue;  
    /* TRUE or FALSE */  
    bool is_locked;  
};
```

- A. monitor
- B. variabilă condiție
- C. semafor
- D. mutex

### 5. Care dintre următoarele apeluri de bibliotecă generează apeluri de sistem?

- A. atoi
- B. memcpy
- C. strcat
- D. fopen

### 6. Pagina unui sistem este de 4KB (4096 de octeți). Care este numărul MAXIM de pagini pe care îl va ocupa buffer-ul de mai jos și care va fi mesajul aferent afișat?

```
/* buffer of 13KB */  
byte buffer[13*1024];  
  
/* Get index of page storing the first byte.*/  
first_page = address_to_page(&buffer[0]);  
  
/* Get index of page storing the last byte.*/  
last_page = address_to_page(&buffer[13*1024-1]);  
  
printf("buffer occupies %d pages.\n",  
       last_page - first_page + 1);
```

- A. buffer occupies 3 pages.
- B. buffer occupies 1 pages.
- C. buffer occupies 4 pages.
- D. buffer occupies 5 pages.

### 7. Ce reprezintă o operație de I/E programată (programmed I/O), denumită și polling?

- A. după declanșarea operației de I/E procesul își poate continua activitatea dacă aceasta nu depinde de rezultatul operației de I/E
- B. o operație de I/E desfășurată printr-un modul DMA care controlează schimbul de date între memoria principală și dispozitivele de I/E
- C. o operație care folosește procesoare specializate de I/E care pot fi programate să execute un set restrâns de instrucțiuni
- D. după declanșarea operației de I/E procesul executa busy wait pentru a aștepta încheierea operației

### 8. Ce este o tabelă inversată de pagini?

- A. o memorie cache care reține asocierea între un subset de pagini logice și pagini fizice (cheie=pagina logică, valoare= pagina fizică)
- B. o tabelă de pagini care e asociată per proces nu per sistem
- C. o tabelă de pagini dedicată spațiului nucleu (kernel space)
- D. o tabelă care are o intrare pentru fiecare pagină fizică, conținutul intrării corespunzând paginii logice

