

GRILE BD

1. SGBD-ul oferă următoarele servicii de bază:
 - a. Arhivarea și dezarhivarea datelor
 - b. Gestiunea accesului concurențial la date
 - c. Limbaj de programare visual pentru accesul la date
 - d. Scanarea datelor
 - e. Optimizarea datelor
2. Precizați care din serviciile de mai jos este pus la dispoziție de un DBMS relațional:
 - a. Difuzarea datelor
 - b. Analiza datelor
 - c. Salvarea și refacerea bazelor de date
 - d. Operativitatea datelor
 - e. Optimizarea și scanarea datelor
3. Obiectele dintr-o bază de date pot fi:
 - a. Funcții pentru concatenarea datelor
 - b. Operatori conjunctivi
 - c. Proceduri stocate
 - d. Cursori
 - e. Tabele și indecși
4. Modelul relațional a fost dezvoltat de:
 - a. Pascal
 - b. Fibonacci
 - c. Bill Gates
 - d. Codd
 - e. Ajax
5. Care din elemente pot fi componente ale bazelor de date relaționale:
 - a. Uniuni
 - b. Reguli
 - c. Restricții NOT NULL
 - d. Interogări
 - e. Operatori logici
6. Restricția CHECK:
 - a. Validează valoarea unei coloane
 - b. Reface datele în caz de eroare
 - c. Asigură ordonarea datelor
 - d. Împiedică redundanța datelor
 - e. Asigură backup-ul bazei de date
7. Restricția NOT NULL:
 - a. Permite indexarea tabelor
 - b. Împiedică folosirea valorilor nule
 - c. Asigură integritatea datelor
 - d. Împiedică redundanța datelor
 - e. Asigură coerența datelor

8. Un tabel poate avea cel mult:
 - a. O singură cheie de index
 - b. O singură cheie externă
 - c. O singură cheie secundară
 - d. O singură cheie primară
 - e. O singură cheie unică

9. Relațiile dintr-o bază de date pot fi:
 - a. Mixte
 - b. Ciclice
 - c. De tipul mai mulți la unu
 - d. Unice
 - e. Singulare

10. Cardinalitatea maximă poate fi:
 - a. Unu
 - b. Unu la unu
 - c. Unu la mai mulți
 - d. N-1
 - e. Infinit

11. Cardinalitatea minimă poate fi:
 - a. Mai mulți la unu
 - b. Unu la unu
 - c. Unu la mai mulți
 - d. -1
 - e. Unu

12. Cheia primară poate fi:
 - a. Formată din valori care se repetă
 - b. Compusă din mai multe coloane
 - c. De tipul 1 la 1
 - d. Folosită la salvarea datelor
 - e. De tipul 1 la N

13. Coloanele care participă la o restricție de unicitate:
 - a. Pot avea valori nule
 - b. Nu pot avea valori nule
 - c. Trebuie să fie de tip numeric
 - d. Trebuie să fie de tip autonumber
 - e. Trebuie să fie de tip CHAR

14. O valoare nulă:
 - a. Este un spațiu liber
 - b. Este un șir vid
 - c. Este o valoare specială care nu e egală cu nimic altceva
 - d. Trebuie să fie de tip autonumber
 - e. Este valoarea zero

15. Vizualizarea:
 - a. Este o tabelă fizică
 - b. Este o tabelă cu cel mult o cheie unică

- c. Este un obiect de tip container
 - d. Nu stochează date
 - e. Nu se comportă ca un tabel
16. Vizualizarea poate:
- a. Împiedica redundanța datelor
 - b. Optimiza structura tabelelor
 - c. Optimiza indecșii
 - d. Stoca cheile de index
 - e. Masca coloanele pe care utilizatorul nu trebuie să le vadă
17. Într-o bază de date putem avea:
- a. Reguli redundante
 - b. Restricții referențiale
 - c. Restricții de tip integer
 - d. Restricții de tip CHAR
 - e. Reguli de concatenare a datelor
18. La nivel conceptual:
- a. Datele sunt destinate utilizatorilor finali
 - b. Structura bazei de date apare sub forma unor view-uri
 - c. Datele sunt privite d.p.d.v. semantic
 - d. Baza de date apare sub forma fișierelor stocate pe calculator
 - e. Se simplifică interacțiunea dintre baza de date și utilizatori
19. Nivelurile de abstractizare a datelor pot fi:
- a. Intern, Extern
 - b. Intern, Referențial
 - c. Semantic, Extern
 - d. Conceptual, Virtual
 - e. Fizic, Interior
20. Proiectarea bazei de date presupune și:
- a. Crearea view-urilor
 - b. Optimizarea structurii bazei de date
 - c. Scăderea timpului de răspuns al server-ului
 - d. Optimizarea spațiului ocupat de baza de date
 - e. Optimizarea relațiilor
21. Analiza cerințelor informaționale presupune
- a. Stabilirea tipurilor de indecși
 - b. Stabilirea relațiilor dintre tabele
 - c. Alegerea SGBD-ului
 - d. Stabilirea fișierelor de tip tabelă
 - e. Stabilirea listelor și a datelor de ieșire ce trebuie obținute
22. Analiza semnificației semantice a datelor:
- a. Identifică documentele primare
 - b. Identifică datele redundante
 - c. Este specifică bazelor de date de mari dimensiuni
 - d. Se folosește în cazul sistemelor proiectate pentru firmele noi
 - e. Stabilirea relațiilor și restricțiile dintre tabele

23. Optimizarea structurii bazei de date urmărește:
- Identificarea indicatorilor și a datelor de ieșire
 - Identificarea datelor redundante
 - Eliminarea anomaliilor de actualizare
 - Eliminarea restricțiilor CHECK
 - Eliminarea valorilor de tip NULL
24. În structura unui tabel nu se pot regăsi:
- Câmpuri de stare
 - Câmpuri de analiză multidimensionale
 - Câmpuri de legătură cu alte tabele
 - Câmpuri de identificare
 - Câmpuri cheie primară
25. Procesul de normalizare determină:
- Un număr mai mare de relații decât într-un model fără normalizare
 - Numărul optim de câmpuri compuse dintr-un tabel
 - Un număr mai mare de restricții decât într-un model fără normalizare
 - Un număr mai mic de tabele decât într-un model fără normalizare
 - Un număr de minim 3 tabele în diagrama ERD
26. Alegeți propoziția adevărată:
- Proiectarea bazelor de date începe cu optimizarea structurii
 - Normalizarea elimină redundanța datelor
 - Cheile candidate se mai numesc și indecși
 - O tabelă poate avea cel puțin o cheie primară
 - Relațiile într-o bază de date se referă la restricții
27. Prima formă normală:
- Asigură eliminarea dependențelor parțiale
 - Asigură eliminarea dependențelor tranzitive
 - Folosește indecși unici în relațiile m la m
 - Presupune descompunerea bazei de date în minim 3 tabele
 - Asigură eliminarea datelor multivaloare
28. A doua formă normală:
- Asigură eliminarea dependențelor tranzitive
 - Asigură eliminarea restricțiilor referențiale
 - Presupune descompunerea bazei de date în minim 3 tabele
 - Asigură eliminarea dependențelor parțiale
 - Asigură coerența datelor multivaloare
29. A treia formă normală:
- Asigură eliminarea restricției NOT NULL
 - Asigură eliminarea dependențelor tranzitive
 - Presupune descompunerea bazei de date în minim 3 tabele
 - Asigură optimizarea spațiului ocupat pe disc de baza de date
 - Asigură partajarea datelor multivaloare
30. A doua formă normală se aplică:
- Numai relațiilor care au identificatoare unice formate din attribute multiple

- b. Numai în cazul relațiilor 1 la m
 - c. Numai dacă baza de date are minim 3 tabele
 - d. Numai pentru relațiile care folosesc indecși unici
 - e. Numai pentru coloanele multivaloare
31. O relație este în a doua formă normală dacă cel puțin:
- a. Relațiile care au identificatoare unice formate din atribute multiple sunt limitate la 1
 - b. Toate relațiile sunt de tipul 1 la m
 - c. Toate atributele non-cheie sunt dependente funcțional de identificatorul unic, luat ca întreg.
 - d. Folosim indecși unici
 - e. Coloanele multivaloare sunt indexate cu chei externe
32. O relație este în a treia formă normală dacă cel puțin:
- a. Relațiile care au identificatoare unice formate din atribute multiple sunt limitate la 2
 - b. Toate relațiile sunt de tipul m la 1
 - c. Toate relațiile de tipul m la n au fost descompuse în relații de tipul 1 la 1
 - d. Toate atributele non-cheie depind numai de identificatorul unic
 - e. Coloanele multivaloare sunt indexate cu chei externe
33. Reducerea redundanței datelor poate asigura:
- a. Eliminarea anomaliilor de ștergere
 - b. Eliminarea anomaliilor de inserare
 - c. Eliminarea restricțiilor CHECK
 - d. Eliminarea atributelor non-cheie
 - e. Eliminarea câmpurilor obținute prin calcul pe baza câmpurilor atomice
34. Limbajul SQL:
- a. Este un limbaj potrivit pentru programarea generală a aplicațiilor
 - b. Este un limbaj declarativ
 - c. Este un limbaj procedural
 - d. Este un limbaj pentru dezvoltarea aplicațiilor web
 - e. Este un limbaj structurat
35. Clienții SQL pot fi:
- a. În cod mașină
 - b. Bazați pe limbaj
 - c. Grafici
 - d. Procedurali
 - e. Bazați pe reguli
36. Care convenție de sintaxă este corectă pentru SQL:
- a. Instrucțiunea SQL se termină prin delimitatorul punct
 - b. Instrucțiunea SQL se termină prin delimitatorul /
 - c. Articolele într-o listă sunt separate prin simbolul /
 - d. Articolele într-o listă sunt separate prin punct și virgulă
 - e. Articolele într-o listă sunt separate prin virgulă
37. Limbajul de manipulare a datelor include instrucțiunile:
- a. DELETE
 - b. SELECT
 - c. DROP
 - d. GRANT

- e. ALTER
38. Limbajul de definiere a datelor include instrucțiunile:
- a. DELETE
 - b. SELECT
 - c. DROP
 - d. GRANT
 - e. COMMIT
39. Limbajul pentru controlul datelor include instrucțiunile:
- a. INSERT
 - b. REVOKE
 - c. DROP
 - d. ROLLBACK
 - e. COMMIT
40. Limbajul de interogare a datelor include instrucțiunile:
- a. GRANT
 - b. REVOKE
 - c. DELETE
 - d. SELECT
 - e. UPDATE
41. Comanda DROP:
- a. Face parte din limbajul de definiere a datelor și șterge un obiect existent în baza de date
 - b. Face parte din limbajul de definiere a datelor și șterge o înregistrare existentă în baza de date
 - c. Face parte din limbajul de manipulare a datelor și șterge un obiect existent în baza de date
 - d. Face parte din limbajul de definiere a datelor și șterge rândurile corespondente din tabela copil
 - e. Face parte din limbajul de definiere a datelor și șterge rândurile corespondente din tabela părinte
42. Comanda CREATE DATABASE:
- a. Face parte din limbajul de manipulare a datelor și crează un obiect de tip bază de date
 - b. Se poate executa după o comandă de tipul CREATE TABLE
 - c. Face parte din limbajul de manipulare a datelor și crează obiectele de tip tabelă
 - d. Face parte din limbajul de definiere a datelor și actualizează bazele de date
 - e. Trebuie executată înainte de comanda CREATE TABLE
43. Restricția CHECK la nivel de coloană impune o regulă:
- a. Prin care coloana trebuie să fie de tip cheie primară
 - b. De unicitate a valorilor din acea coloană
 - c. Care nu poate referi nici o altă coloană din tabel
 - d. Care poate referi o altă coloană din tabel
 - e. Care poate referi o altă coloană din alt tabel
44. Restricția referențială (FOREIGN KEY) la nivel de tabel impune o regulă:
- a. Prin care coloana trebuie să fie de tip cheie externă
 - b. De unicitate a valorilor din acea coloană
 - c. Care nu poate referi nici o altă coloană din tabel
 - d. De repetare a valorilor unei coloane
 - e. Care poate referi coloane multiple

45. Clauza DEFAULT impune o regulă:
- Prin care coloana trebuie să fie de tip numeric sau șir de caractere
 - La nivel de coloană care la inserarea unui nou rând stabilește o valoare explicită pe acea coloană
 - Care nu poate referi nici o altă coloană din tabel
 - De unicitate a valorilor unei coloane
 - Care poate referi simultan coloane multiple
46. Principalul avantaj al restricțiilor la nivel de tabel este că:
- Pot referi mai multe coloane
 - Pot referi mai multe rânduri
 - Scade timpul de răspuns al serverului
 - Simplifică structura de date
 - Asigură integritatea și refacerea bazei de date
47. Prin instrucțiunea CREATE VIEW se obține o:
- Nouă tabelă fizică al cărei conținut poate fi identic sau nu cu cel al tabelii din care a fost create
 - Interogare SQL stocată al cărei conținut trebuie să fie identic cu cel al tabelii din care a fost create
 - Tabelă SQL care poate fi referită de instrucțiunile DML și DQL
 - Interogare SQL stocată care poate fi referită de instrucțiunile DML și DQL
 - Tabelă SQL al cărei conținut nu poate fi modificat
48. Instrucțiunea ALTER TABLE permite:
- Modificarea conținutului unui rând
 - Adăugarea unui rând
 - Adăugarea unei restricții
 - Modificarea interogării datelor din tabelă
 - Crearea unei tabele părinte
49. Instrucțiunea ALTER TABLE permite:
- Actualizarea conținutului unei tabelei
 - Ștergerea cheii primare a unui tabel
 - Ștergerea datelor redundante
 - Modificarea datelor din tabelă
 - Implementarea funcției SUM
50. Alegeți propoziția corectă:
- Instrucțiunea DELETE permite ștergerea tabelilor dintr-o bază de date
 - Instrucțiunea ALTER TABLE nu permite modificarea definiției unei coloane
 - Instrucțiunea ALTER TABLE nu permite redenumirea unei coloane
 - Instrucțiunea DROP permite ștergerea relațiilor de tipul 1 la m sau m la 1 sau m la n
 - Dacă un tabel este referit de o restricție referențială, sistemul DBMS nu va permite să ștergeți tabelul
51. Instrucțiunea SELECT:
- Face parte din limbajul de interogare a datelor alături de instrucțiunile DELETE, INSERT
 - Face parte din limbajul de manipulare a datelor
 - Face parte din limbajul de control a datelor
 - Permite stabilirea pseudonimelor pentru coloanele afișate în interogare
 - Permite stabilirea pseudonimelor pentru rândurile afișate în interogare
52. Utilizarea clauzei WHERE în instrucțiunea SELECT:
- Permite filtrarea rezultatelor

- b. Asigură sortarea datelor crescător sau descrescător
 - c. Elimină rândurile duplicate din rezultatul interogării
 - d. Permite stabilirea pseudonimelor pentru coloanele afișate în interogare
 - e. Nu poate fi folosită în standardul SQL
53. Utilizarea clauzelor DISTINCT și ORDER BY în instrucțiunea SELECT:
- a. Asigură sortarea datelor crescător sau descrescător
 - b. Elimină rândurile duplicate și stabilește criteriile de sortare a datelor afișate în rezultatul interogării
 - c. Permite stabilirea pseudonimelor și ordonarea datelor afișate în interogare
 - d. Permite specificarea unor condiții de selecție a rezultatelor
 - e. Nu pot fi folosite simultan în standardul SQL
54. Folosirea operatorilor de comparare în clauza WHERE:
- a. Asigură sortarea datelor crescător sau descrescător
 - b. Presupune folosirea cuvântului cheie LIKE
 - c. Presupune folosirea cuvântului cheie IN
 - d. Se referă la cuvintele cheie AND, OR
 - e. Are ca rezultat o valoare logică de „adevărat” sau „fals” în funcție de care se afișează rezultatele
55. Operatorii logici din clauza WHERE pot fi:
- a. AND, OR
 - b. !=
 - c. NOT BETWEEN
 - d. IS TRUE
 - e. FALSE
56. Instrucțiunea SELECT MOVIE_TITLE FROM MOVIE WHERE MOVIE_TITLE LIKE '%on%';
- a. Afișează numai titlurile de filme care conțin șirul de caractere „ON”
 - b. Afișează numai titlurile de filme care conțin șirul de caractere „on”
 - c. Afișează titlurile de filme care se termină cu șirul de caractere „on”
 - d. Afișează titlurile de filme care conțin șirul de caractere „on” sau „ON” sau „On” sau „oN”
 - e. Afișează titlurile de filme care încep cu șirul de caractere „on”
57. Instrucțiunea SELECT PERSON_FAMILY_NAME AS NUME FROM PERSON WHERE SUBSTRING(PERSON_FAMILY_NAME,1,1)='B'
- a. Afișează numele de familie folosind pseudonimul PERSON pentru persoanele al căror nume începe cu „B” sau „b”
 - b. Afișează numele de familie folosind pseudonimul NUME pentru persoanele al căror nume începe cu „B” sau „b”
 - c. Afișează numele de familie folosind pseudonimul NUME pentru persoanele al căror nume începe numai cu „B”
 - d. Afișează numele de familie folosind pseudonimul NUME pentru persoanele al căror nume începe numai cu „b”
 - e. Afișează numele de familie folosind pseudonimul NUME pentru persoanele al căror nume conține litera „B” sau „b”
58. Instrucțiunea SELECT 'Matei!' LIKE 'Matei_*';
- a. Returnează valoarea 0
 - b. Returnează valoarea 1
 - c. Returnează valoarea Matei
 - d. Nu returnează nici o valoare

- e. Nu poate fi executată deoarece conține o eroare de sintaxă
59. Rolul unei funcții de agregare este de a:
- Combina mai multe rânduri de date într-un singur rând care afișează rezultatul funcției
 - Combina mai multe coloane într-o singură coloană
 - Combina mai multe rânduri de date cu coloanele pe care se aplică funcția
 - Returna o valoare logică de „adevărat” sau „fals”
 - Compara diferite coloane din structura unei table
60. Alegeți instrucțiunea corectă care afișează fiecare cod de gen, împreună cu numărul de filme asociate fiecărui cod:
- SELECT Movie_Genre_Code, SUM(Pret_Dvd) AS Nr_Filme_Cod FROM Movie GROUP BY Movie_Genre_Code;
 - SELECT Movie_Genre_Code, COUNT(*) AS Nr_Filme_Cod FROM Movie ORDER BY Movie_Genre_Code;
 - SELECT Movie_Genre_Code, COUNT(*) AS Nr_Filme_Cod FROM Movie GROUP BY Movie_Genre_Code;
 - SELECT COUNT(*) AS Nr_Filme_Cod FROM Movie ORDER BY Movie_Genre_Code;
 - SELECT Movie_Genre_Code, Sum(Id_Film) AS Nr_Filme_Cod FROM Movie ORDER BY Movie_Genre_Code;
61. Operatorul UNION:
- Combină mai multe rânduri de date într-un singur rând care afișează rezultatul funcției folosită pentru prelucrarea datelor
 - Combină mai multe coloane într-o singură coloană și aplică o funcție de prelucrare
 - Returnează o valoare logică de „adevărat” sau „fals”
 - Adaugă rândurile din setul de înregistrări al unei interogări la cel al unei alte interogări fără a elimina rândurile duplicate
 - Adaugă rândurile din setul de înregistrări al unei interogări la cel al unei alte interogări și, în același timp, elimină rândurile duplicate
62. Printr-o uniune (JOIN):
- Obținem o funcție de agregare a datelor
 - Se realizează gruparea datelor
 - Se returnează o valoare logică de „adevărat” sau „fals” în funcție de rezultatul funcției
 - Se combină coloane din două sau mai multe table în rezultatul unei singure interogări
 - Se adaugă rândurile din setul de înregistrări al unei interogări la cel al unei alte interogări și, în același timp, se elimină rândurile duplicate
63. Realizarea uniunilor (JOIN) se poate face:
- Prin folosirea funcțiilor de agregare a datelor
 - Prin folosirea clauzei WHERE
 - Folosind operatorul LIKE
 - Folosind clauza GROUP BY
 - Folosind clauza EQUIJOIN
64. Uniunea standard sau internă:
- Folosește clauza STANDARD
 - Se aplică dacă tablele interogate au structura identică
 - Leagă una sau mai multe coloane dintr-un tabel cu coloane similare dintr-un alt tabel dacă valorile datelor sunt egale pe acele coloane

- d. Folosește clauza `INNER JOIN`
 - e. Poate fi aplicată numai tabelelor legate printr-o relație 1 la 1
65. Uniunea prin `JOIN` folosind cuvântul cheie `USING` e posibilă când:
- a. Coloanele pe care se face legătura au nume identice în ambele tabele
 - b. Coloanele pe care se face legătura au nume diferite în ambele tabele
 - c. Tabelele legate au un număr identic de rânduri
 - d. Tabelele legate au structura de date diferită
 - e. Tabelele sunt legate printr-o relație 1 la 1
66. Uniunea externă:
- a. Elimină rândurile duplicate
 - b. Include în setul de rezultate și rândurile pentru care nu există legături din cel puțin unul dintre tabele
 - c. Se aplică tabelor legate care au un număr identic de rânduri
 - d. Tabelele legate au structura de date diferită
 - e. Folosește clauza `EXCEPT` în sintaxa instrucțiunii
67. O auto-uniune:
- a. Nu poate fi realizată în sistemul MySQL
 - b. Include și coloanele care au aceleași pseudonime
 - c. Se aplică în cazul restricțiilor recursive
 - d. Se aplică în cazul relațiilor recursive
 - e. Folosește clauza `EXCEPT` în sintaxa instrucțiunii
68. Subinterogarea:
- a. se referă la o instrucțiune `SELECT` care conține o instrucțiune `SELECT` subordonată
 - b. Poate fi realizată cu operatorul `INTERSECT`
 - c. Se aplică în cazul relațiilor de tipul 1 la 1
 - d. Nu este implementată în sistemul MySQL
 - e. Folosește clauza `EXCEPT` în sintaxa instrucțiunii
69. Subinterogările corelate:
- a. Sunt subinterogări în care interogarea internă nu face nici o referire la interogarea externă care o conține
 - b. Pot fi realizate cu operatorul `INTERSECT`
 - c. Sunt mult mai puțin eficiente decât subinterogările necorelate
 - d. Sunt mult mai eficiente decât subinterogările necorelate
 - e. Folosesc clauza `EXCEPT` în sintaxa instrucțiunii
70. Alegeți regula corectă pentru formarea unei instrucțiuni DML:
- a. Ștergerile în tabele trebuie să fie făcute ierarhic: mai întâi rândurile părinte, apoi rândurile copil
 - b. Nu puteți șterge un rând părinte dacă există rânduri subordonate care referă valoarea din rândul părinte, decât dacă restricția a fost definită cu opțiunea `ON DELETE CASCADE`
 - c. O instrucțiune `INSERT` poate stoca într-o coloană o valoare care încalcă o restricție `CHECK` definită pentru coloana respectivă
 - d. O instrucțiune `UPDATE` poate stoca într-o coloană o valoare care încalcă o restricție `CHECK` definită pentru coloana respectivă
 - e. Inserările în tabele trebuie să fie făcute ierarhic: mai întâi rândurile copii, apoi rândurile părinte

Aplicații:

71. Pornind de la relația nenormalizată referitoare la facturile emise către clienți, este redată soluția normalizată indicând relațiile corespunzătoare celei de-a doua forme normale – FN2.

Relația nenormalizată este FACTURI (nr_factura, data_factura, nume_client, adresa_client, banca_client, nr_cont_client, delegat, cod_produș, denumire_produș, unitate_de_masura, cantitate, pret_unitar, valoare, valoare_tva, total_valoare_factura, total_valoare_tva).

Alegeți schema corectă pentru FN2.

- a. FACTURI (**nr_factura** - **pk**, data_factura, delegat, total_valoare_factura, total_valoare_tva, nume_client - *fk*)
PRODUSE (**nr_factura, cod_produș** – **pk**, denumire_produș, unitate_de_masura, cantitate, pret_unitar, valoare, valoare_tva) și
CLIENTI (**nume_client** - **pk**, judet_client, localitate_client, stradă_client, nr_stradă_client, bloc_client, apartament_client, banca_client, nr_cont_client)
- b. FACTURI (**nr_factura** - **pk**, data_factura, delegat, total_valoare_factura, total_valoare_tva, nume_client - *fk*)
ARTICOLE_FACTURI (**nr_factura, cod_produș** – **pk**, cantitate, pret_unitar, valoare, valoare_tva)
PRODUSE (**cod_produș** - **pk**, denumire_produș, unitate_de_masura) și
CLIENTI (**nume_client** - **pk**, judet_client, localitate_client, stradă_client, nr_stradă_client, bloc_client, apartament_client, banca_client, nr_cont_client)
- c. FACTURI (**nr_factura** - **pk**, data_factura, nume_client, judet_client, localitate_client, stradă_client, nr_stradă_client, bloc_client, apartament_client, banca_client, nr_cont_client, delegat, total_valoare_factura, total_valoare_tva)
și
PRODUSE (**cod_produș** - **pk**, denumire_produș, unitate_de_masura, cantitate, pret_unitar, valoare, valoare_tva)
- d. FACTURI (**nr_factura** - **pk**, data_factura, nume_client, judet_client, localitate_client, stradă_client, nr_stradă_client, bloc_client, apartament_client, banca_client, nr_cont_client, delegat, total_valoare_factura, total_valoare_tva)
ARTICOLE_FACTURI (**nr_factura, cod_produș** – **pk**, cantitate, pret_unitar, valoare, valoare_tva) și
PRODUSE (**cod_produș** - **pk**, denumire_produș, unitate_de_masura)
- e. FACTURI (**nr_factura** - **pk**, data_factura, nume_client, judet_client, localitate_client, stradă_client, nr_stradă_client, bloc_client, apartament_client, banca_client, nr_cont_client, delegat, total_valoare_factura, total_valoare_tva, cod_produș, cantitate, pret_unitar, valoare, valoare_tva)
și
PRODUSE (**cod_produș** - **pk**, denumire_produș, unitate_de_masura)

72. Se proiectează o bază de date folosind tehnica normalizării pentru un furnizor de servicii telefonice, având în vedere următoarele constrângeri:

- furnizorul are mai mulți abonați
- fiecare abonat poate deține unul sau mai multe numere de telefon
- de pe fiecare număr de telefon se pot efectua mai multe apeluri telefonice
- apelul telefonic se taxează în funcție ora la care a fost efectuat.

Structura bazei de date nenormalizată este redată de relația de mai jos:

ABONATI (idAbonat, numeAbonat, prenumeAbonat, adresaAbonat, idTelefon, nrTelefon, idApel, destinatieApel, durataApel, intervalOrar, tarifApel).

Alegeți schema corectă pentru FN2.

- a. ABONATI (**idAbonat** - **pk**, numeAbonat, prenumeAbonat, judetAbonat, localitateAbonat, stradaAbonat)

TELEFOANE (**idTelefon** - **pk**, nrTelefon)

ABONATI_TELEFOANE (**idAbonat, idTelefon** - **pk**)

APELURI (**idApel** - **pk**, destinatieApel, durataApel, intervalOrar, tarifApel)

TELEFOANE_APELURI (**idTelefon, idApel** - **pk**)

b. ABONATI (**idAbonat** - **pk**, numeAbonat, prenumeAbonat, judetAbonat, localitateAbonat, stradaAbonat)

TELEFOANE (**idAbonat, idTelefon** - **pk**, nrTelefon, idApel, destinatieApel, durataApel, intervalOrar, tarifApel)

c. TELEFOANE (**idAbonat, idTelefon** - **pk**, nrTelefon)

APELURI (**idTelefon, idApel** - **pk**, destinatieApel, durataApel, intervalOrar, tarifApel)

ABONATI (**idAbonat** - **pk**, numeAbonat, prenumeAbonat, judetAbonat, localitateAbonat, stradaAbonat)

d. ABONATI (**idAbonat** - **pk**, numeAbonat, prenumeAbonat, judetAbonat, localitateAbonat, stradaAbonat)

TELEFOANE (**idTelefon** - **pk**, nrTelefon)

ABONATI_TELEFOANE (**idAbonat, idTelefon** - **pk**)

TELEFOANE_APELURI (**idTelefon, idApel** - **pk**, destinatieApel, durataApel, intervalOrar, tarifApel)

e. ABONATI (**idAbonat** - **pk**, numeAbonat, prenumeAbonat, judetAbonat, localitateAbonat, stradaAbonat)

ABONATI_TELEFOANE (**idAbonat, idTelefon** - **pk**, nrTelefon)

APELURI (**idApel** - **pk**, destinatieApel, durataApel, intervalOrar, tarifApel)

TELEFOANE_APELURI (**idTelefon, idApel** - **pk**)

73. Se proiectează o bază de date folosind tehnica normalizării pentru a gestiona comenzile de produse dintr-o firmă, având în vedere următoarele constrângeri:

- Un client poate lansa n comenzi
- Fiecare comandă poate avea mai multe repere
- Fiecare comandă poate fi livrată în mai multe localități.

Structura bazei de date nenormalizată este redată de relația de mai jos:

COMENZI

IdComanda	DataComanda	Localitate	IdReper	NumeReper	Cantitate	UM	CodClient	NumeClient
C01	05.12.2011	Craiova, Slatina, Bals	R1	Ulei	10	tone	M1	Metro
C01	05.12.2011	Craiova, Slatina, Bals	R2	Zahar	10	kg	M1	Metro
C02	07.12.2011	Craiova, Bucuresti	R1	Ulei	2	tone	K1	Kaufland
C03	02.12.2011	Slatina, Bals	R3	Faina	108	kg	A1	S.C. Alfa S.R.L

Alegeți schema corectă pentru FN1.

a. LOCALITATI (**IdLocalitate** - **pk**, Localitate)

COMENZI (**IdComanda** - **pk**, DataComanda, IdClient, NumeClient)

REPERE (**IdReper** - **pk**, NumeReper, UM)

REPERE_COMANDA (*IdComanda* - *fk*, *IdReper* - *fk*, Cantitate)

LOCALITATI_COMANDA (*IdComanda* - *fk*, *IdLocalitate* - *fk*)

b. LOCALITATI (**IdLocalitate** - **pk**, Localitate)

COMENZI (**IdComanda** - **pk**, DataComanda, IdClient, NumeClient)

REPERE (**IdReper** - **pk**, NumeReper, UM)

c. LOCALITATI (**IdLocalitate** - **pk**, Localitate)

- COMENZI (**IdComanda - pk**, DataComanda, IdClient, NumeClient)
 REPERE_COMANDA (*IdComanda - fk, IdReper - fk, Cantitate*)
 LOCALITATI_COMANDA (*IdComanda - fk, IdLocalitate - fk*)
- d. LOCALITATI (**IdLocalitate - pk**, Localitate)
 COMENZI (**IdComanda - pk**, DataComanda, IdClient, NumeClient)
 REPERE (**IdReper - pk**, NumeReper, UM)
 REPERE_COMANDA (*IdComanda - fk, IdReper - fk, Cantitate*)
- e. COMENZI (**IdComanda, IdReper, IdLocalitate - pk**, DataComanda, NumeReper, Cantitate, UM, CodClient, NumeClient)
 LOCALITATI (**IdLocalitate - pk**, Localitate)

74. Se proiectează o bază de date folosind tehnica normalizării pentru a gestiona comenzile de produse dintr-o firmă, având în vedere următoarele constrângeri:

- Un client poate lansa n comenzi
- Fiecare comandă poate avea mai multe reperi
- Fiecare comandă poate fi livrată în mai multe localități.

Structura bazei de date nenormalizată este redată de relația de mai jos:

COMENZI

IdComanda	DataComanda	Localitate	IdReper	NumeReper	Cantitate	UM	CodClient	NumeClient
C01	05.12.2011	Craiova, Slatina, Bals	R1	Ulei	10	tone	M1	Metro
C01	05.12.2011	Craiova, Slatina, Bals	R2	Zahar	10	kg	M1	Metro
C02	07.12.2011	Craiova, Bucuresti	R1	Ulei	2	tone	K1	Kaufland
C03	02.12.2011	Slatina, Bals	R3	Faina	108	kg	A1	S.C. Alfa S.R.L

Alegeți schema corectă pentru FN3.

- a. LOCALITATI (**IdLocalitate - pk**, Localitate)
 COMENZI (**IdComanda - pk**, DataComanda, IdClient, NumeClient)
 REPERE (**IdReper - pk**, NumeReper, UM)
 REPERE_COMANDA (*IdComanda - fk, IdReper - fk, Cantitate, IdLocalitate - fk*)
- b. LOCALITATI (**IdLocalitate - pk**, Localitate)
 COMENZI (**IdComanda - pk**, DataComanda, IdClient, NumeClient)
 REPERE (**IdReper - pk**, NumeReper, UM, Cantitate)
- c. LOCALITATI (**IdLocalitate - pk**, Localitate)
 COMENZI (**IdComanda - pk**, DataComanda, IdClient, NumeClient)
 REPERE_COMANDA (*IdComanda - fk, IdReper - fk, Cantitate*)
 LOCALITATI_COMANDA (*IdComanda - fk, IdLocalitate - fk, IdClient - fk*)
- d. LOCALITATI (**IdLocalitate - pk**, Localitate)
 COMENZI (**IdComanda - pk**, DataComanda, IdClient, NumeClient)
 REPERE (**IdReper - pk**, NumeReper, UM)
 REPERE_COMANDA (*IdComanda - fk, IdReper - fk, Cantitate*)
- e. LOCALITATI (**IdLocalitate - pk**, Localitate)
 COMENZI (**IdComanda - pk**, DataComanda, *IdClient - fk*)
 CLIENTI (**IdClient - pk**, NumeClient)
 REPERE (**IdReper - pk**, NumeReper, UM)
 REPERE_COMANDA (*IdComanda - fk, IdReper - fk, Cantitate*)
 LOCALITATI_COMANDA (*IdComanda - fk, IdLocalitate - fk*)

75. Se proiectează o bază de date folosind tehnica normalizării pentru o bibliotecă care poate oferi cititorilor cărți, având în vedere următoarele constrângeri:
- Fiecare carte poate fi livrată în format: electronic sau hârtie
 - Fiecare carte face parte dintr-o categorie de cărți
 - Fiecare carte poate avea mai mulți autori
 - Fiecare carte poate fi tradusă în mai multe limbi.

Structura bazei de date nenormalizată este redată de relația de mai jos:

BIBLIOTECA

IdCarte	ISBN	IdCategorie	DenCategorie	Limba	TitluCarte	AnPublicare	Format	Editura	Autori
1	234456	D	Drama	En, Fr, Es	Poezii de seară	2010	Hârtie, Electronic	All	Anghel I.
2	123456	A	Actiune	En, Ro	Aventuri	2011	Hârtie, Electronic	Univers	Lorn M., Travolta J.

Alegeți schema corectă pentru FN2.

- LIMBA (*IdCarte - fk*, Limba)
AUTORI (*IdCarte - fk*, NumeAutor, PrenumeAutor)
CARTI (**IdCarte - pk**, ISBN, IdCategorie, DenCategorie, TitluCarte, AnPublicare, FormatHartie, FormatElectronic, Editura)
 - AUTORI (*IdCarte - fk*, NumeAutor, PrenumeAutor)
CARTI (**IdCarte - pk**, ISBN, IdCategorie, DenCategorie, TitluCarte, AnPublicare, FormatHartie, FormatElectronic, Editura, Limba)
 - Deoarece orice relație în prima formă normală care are cheia primară formată dintr-un singur atribut este *automat* în a doua formă normală, în acest caz FN1=FN2
 - AUTORI (*IdCarte - fk*, NumeAutor, PrenumeAutor)
CARTI (**IdCarte - pk**, ISBN, *IdCategorie - fk*, TitluCarte, AnPublicare, FormatHartie, FormatElectronic, Editura)
CATEGORII (**IdCategorie - pk**, DenCategorie)
 - CARTI (**IdCarte - pk**, ISBN, *IdCategorie - fk*, TitluCarte, AnPublicare, FormatHartie, FormatElectronic, Editura, NumeAutor, PrenumeAutor, Limba)
CATEGORII (**IdCategorie - pk**, DenCategorie)
76. Folosind Baza de date COMENZI care conține tabelele: PERSOANE (**IdPersoana - pk**, Nume, Prenume, Oras) și COMENZI (**IdComanda - pk**, Valoare, *IdPersoana - fk*) precizați codul SQL pentru o interogare care să afișeze Numele, Prenumele, Id Comanda și Valoarea comenzii iar înregistrările să fie ordonate după Oras.
- SELECT a.IdPersoana, a.Nume, a.Prenume, a.Oras, b.IdComanda, b.Valoare
FROM Persoane a LEFT JOIN Comenzi b
ON a.IdPersoana = b.IdPersoana
ORDER BY a.Nume DESC;
 - SELECT Persoane.Nume, Persoane.Prenume, Comenzi.IdComanda, Comenzi.Valoare
FROM Persoane INNER JOIN Comenzi
ON Persoane.IdPersoana = Comenzi.IdPersoana
ORDER BY Persoane.Oras;
 - SELECT a.IdPersoana, a.Nume, a.Prenume, a.Oras, b.IdComanda, b.Valoare
FROM Persoane a LEFT JOIN Comenzi b
ON a.IdPersoana = b.IdPersoana
GROUP BY a.Nume;
 - SELECT a.IdPersoana, a.Nume, a.Prenume, a.Oras, b.IdComanda, b.Valoare
FROM Persoane a RIGHT JOIN Comenzi b

ON a.IdPersoana = b.IdPersoana

ORDER BY a.Nume DESC;

- e. SELECT a.IdPersoana, a.Nume, a.Prenume, a.Oras, b.IdComanda, b.Valoare
FROM Persoane a FULL JOIN Comenzi b
ON a.IdPersoana = b.IdPersoana
ORDER BY a.Oras DESC;

77. Folosind Baza de date COMENZI care conține tabelele: PERSOANE (**IdPersoana - pk**, Nume, Prenume, Oras) și COMENZI (**IdComanda - pk**, Valoare, *IdPersoana - fk*) precizați codul SQL care generează o interogare ce afișează Id Persoana și Valoarea totală a comenzilor efectuate de fiecare persoană.

- a. SELECT Comenzi.IdPersoana, AVG(Comenzi.Valoare)
FROM Comenzi GROUP BY Comenzi.IdPersoana;
b. SELECT Comenzi.IdPersoana, AVG(Comenzi.Valoare)
FROM Comenzi ORDER BY Comenzi.IdPersoana;
c. SELECT Comenzi.IdPersoana, SUM(Comenzi.Valoare)
FROM Comenzi ORDER BY Comenzi.IdPersoana;
d. SELECT Comenzi.IdPersoana, SUM(Comenzi.Valoare)
FROM Comenzi GROUP BY Comenzi.IdPersoana;
e. SELECT Comenzi.IdPersoana, SUM(Comenzi.Valoare)
FROM Comenzi;

78. Folosind Baza de date COMENZI care conține tabelele: PERSOANE (**IdPersoana - pk**, Nume, Prenume, Oras) și COMENZI (**IdComanda - pk**, Valoare, *IdPersoana - fk*) precizați codul SQL pentru o interogare care afișează Numele, Prenumele, Id Comanda și Valoarea comenzii pentru acele comenzi care nu fac parte din intervalul 1500-2000 lei.

- a. SELECT Persoane.Nume, Persoane.Prenume, Comenzi.IdComanda, Comenzi.Valoare
FROM Persoane INNER JOIN Comenzi
ON Persoane.IdPersoana = Comenzi.IdPersoana
WHERE Comenzi.Valoare NOT BETWEEN 1500 AND 2000;
b. SELECT Persoane.Nume, Persoane.Prenume, Comenzi.IdComanda, Comenzi.Valoare
FROM Persoane INNER JOIN Comenzi
WHERE Comenzi.Valoare NOT BETWEEN 1500 AND 2000;
c. SELECT Persoane.Nume, Persoane.Prenume, Comenzi.IdComanda, Comenzi.Valoare
FROM Persoane INNER JOIN Comenzi
ON Persoane.IdPersoana = Comenzi.IdPersoana
WHERE Comenzi.Valoare BETWEEN 1500 AND 2000;
d. SELECT Persoane.Nume, Persoane.Prenume, Comenzi.IdComanda, Comenzi.Valoare
FROM Persoane FULL JOIN Comenzi
ON Persoane.IdPersoana = Comenzi.IdPersoana
WHERE Comenzi.Valoare BETWEEN 1500 AND 2000;
e. SELECT Persoane.Nume, Persoane.Prenume, Comenzi.IdComanda, Comenzi.Valoare
FROM Persoane RIGHT JOIN Comenzi
ON Persoane.IdPersoana = Comenzi.IdPersoana
WHERE Comenzi.Valoare NOT BETWEEN 1500 AND 2000;

79. Folosind Baza de date MOVIE cu tabelele relaționate:

Movie Copy

MOVIE_ID (pk)	COPY_NUMBER (pk)	DATA_ACQUIRED	DATE_SOLD	MEDIA_FORMAT
---------------	---------------------	---------------	-----------	--------------

Movie Language

MOVIE_ID (fk)	LANGUAGE_CODE
---------------	---------------

Movie

MOVIE_ID (pk)	MOVIE_GENRE_CODE	MPAA_RATING_CODE	MOVIE_TITLE	RETAIL_PRICE_VHS	RETAIL_PRICE_DVD	YEAR_PRODUCED
------------------	------------------	------------------	-------------	------------------	------------------	---------------

Precizați secvența corectă de instrucțiuni SQL pentru a șterge filmul cu MOVIE_ID = 210

- DELETE FROM MOVIE_COPY AND FROM MOVIE_LANGUAGE AND FROM MOVIE WHERE MOVIE_ID = 21;
- DELETE FROM MOVIE WHERE MOVIE_ID = 21;
DELETE FROM MOVIE_COPY WHERE MOVIE_ID = 21;
DELETE FROM MOVIE_LANGUAGE WHERE MOVIE_ID = 21;
- DELETE FROM MOVIE_LANGUAGE WHERE MOVIE_ID = 21;
DELETE FROM MOVIE WHERE MOVIE_ID = 21;
DELETE FROM MOVIE_COPY WHERE MOVIE_ID = 21;
- DELETE FROM MOVIE_COPY AND FROM MOVIE_LANGUAGE WHERE MOVIE_ID = 21;
DELETE FROM MOVIE WHERE MOVIE_ID = 21;
- DELETE FROM MOVIE_COPY WHERE MOVIE_ID = 21;
DELETE FROM MOVIE_LANGUAGE WHERE MOVIE_ID = 21;
DELETE FROM MOVIE WHERE MOVIE_ID = 21;

80. Folosind Baza de date MOVIE cu tabelele relaționate:

Movie_Language

MOVIE_ID (fk)	LANGUAGE_CODE
---------------	---------------

Movie

MOVIE_ID (pk)	MOVIE_GENRE_CODE	MPAA_RATING_CODE	MOVIE_TITLE	RETAIL_PRICE_VHS	RETAIL_PRICE_DVD	YEAR_PRODUCED
------------------	------------------	------------------	-------------	------------------	------------------	---------------

Presupunem că toate filmele din tabela MOVIE sunt acum disponibile și în limba germană. Tabelul MOVIE_LANGUAGE conține rânduri pentru limba germană pentru o parte din filme, așa că acum vrem să le adăugăm numai pe cele care lipsesc.

Alegeți codul SQL corect care rezolvă cerința de mai sus.

- INSERT INTO MOVIE_LANGUAGE (MOVIE_ID, LANGUAGE_CODE)
SELECT MOVIE_ID, 'ge' FROM MOVIE WHERE MOVIE_ID NOT IN;
- INSERT INTO MOVIE_LANGUAGE (MOVIE_ID, LANGUAGE_CODE)
SELECT MOVIE_ID, 'ge' FROM MOVIE WHERE MOVIE_ID NOT IN
(SELECT MOVIE_ID FROM MOVIE_LANGUAGE WHERE LANGUAGE_CODE = 'ge');
- INSERT INTO MOVIE_LANGUAGE (MOVIE_ID, LANGUAGE_CODE)
SELECT MOVIE_ID, 'ge' FROM MOVIE WHERE MOVIE_ID IN
(SELECT MOVIE_ID FROM MOVIE_LANGUAGE WHERE LANGUAGE_CODE = 'ge');
- INSERT INTO MOVIE_LANGUAGE (MOVIE_ID, LANGUAGE_CODE)
SELECT MOVIE_ID, 'ge' FROM MOVIE;
- INSERT INTO MOVIE_LANGUAGE (MOVIE_ID, LANGUAGE_CODE)
SELECT MOVIE_ID, 'ge' FROM MOVIE WHERE MOVIE_ID IN
(SELECT MOVIE_ID FROM MOVIE_LANGUAGE WHERE LANGUAGE_CODE < >
'ge');

RĂSPUNSURI CORECTE BD

Nr. Întrebare	Răspuns corect
1	B
2	C
3	E
4	D
5	C
6	A
7	B
8	D
9	C
10	A
11	E
12	B
13	A
14	C
15	D
16	E
17	B
18	C
19	A
20	B
21	E
22	D
23	C
24	B
25	A
26	C
27	E
28	D
29	B
30	A
31	C
32	D
33	E
34	B
35	C
36	E
37	A
38	C
39	B
40	D
41	A
42	E
43	C
44	E
45	B
46	A
47	D

Nr. Întrebare	Răspuns corect
48	C
49	B
50	E
51	D
52	A
53	B
54	E
55	C
56	D
57	B
58	A
59	A
60	C
61	E
62	D
63	B
64	C
65	A
66	B
67	D
68	A
69	C
70	B
Aplicații	
71	D
72	A
73	E
74	E
75	C
76	B
77	D
78	A
79	E
80	B

GRILE POO

1. *În procesul de abstractizare atenția este îndreptată exclusiv spre:*

- a. caracteristicile esențiale ale unui obiect, care îl aseamănă cu alte obiecte;
- b. comportarea unui obiect ce se caracterizează printr-un anumit serviciu sau resursă pe care o pune la dispoziția altor obiecte;
- c. aspectul exterior al obiectului, adică spre comportarea lui, ignorând implementarea acestei comportări;
- d. toate caracteristicile esențiale ale unui obiect, oferind o definiție precisă a elementelor comune cu alte obiectele.
- e. executarea de acțiuni, unde își poate modifica și comunica starea și poate comunica cu alte obiecte din sistem dezvăluind cum au fost implementate acele facilități.

2. *Modelul Client-Server are mai multe caracteristici între care nu se regăsește:*

- a. totalitatea serviciilor oferite de un obiect server constituie un contract sau o responsabilitate a obiectului față de alte obiecte;
- b. comportarea unui obiect se caracterizează printr-o sumă de servicii sau resurse pe care el le pune la dispoziția altor obiecte;
- c. responsabilitățile sunt îndeplinite prin intermediul unor operații, fiecare operație a unui obiect caracterizându-se printr-o semnătură unică;
- d. un obiect, numit server, oferă servicii altor obiecte, numite clienți;
- e. comportarea unui obiect se caracterizează printr-un anumit serviciu sau resursă pe care le pune la dispoziția altui obiect.

3. *Încapsularea asigură:*

- a. faptul că obiectele nu pot schimba starea internă a altor obiecte în mod direct (ci doar prin metode puse la dispoziție de obiectul respectiv);
- b. identificarea protocolului său;
- c. caracteristicile esențiale ale unui obiect, care fac ca acesta să se distingă de alte obiecte;
- d. o definiție precisă a granițelor conceptuale ale obiectului;
- e. procesul în care are loc expunerea implementării față de majoritatea obiectelor-client.

4. *Beneficiile încapsulării se concretizează în:*

- a. permite modificarea programelor cu un efort considerabil;
- b. permite modificarea reprezentării afectând clienții săi;
- c. conferă dependență între programe, operații și date;
- d. permite modificarea programelor într-o manieră eficientă, cu un efort limitat și bine localizat;
- e. nu aplică principiul abstracției.

5. *Una din regulile de mai jos nu corespund regulilor generale de modularizare:*

- a. structura fiecărui modul trebuie să fie suficient de simplă pentru a putea fi complet înțeleasă;
- b. implementarea unui modul trebuie să depindă doar de interfețele altor module, respectiv trebuie să fie posibilă modificarea implementării unui modul fără a avea cunoștințe despre implementarea altor module și fără a afecta comportarea celorlalte module;
- c. detaliile sistemului care se presupune că se vor modifica independent vor fi plasate în același modul;
- d. singurele legături între module vor fi acelea a căror modificare este improbabilă;
- e. orice structură de date este încapsulată într-un modul, ea putând fi accesată direct din interiorul modulului dar nu poate fi accesată din afara modulului decât prin intermediul obiectelor și claselor conținute în acel modul.

6. Conceptul de moștenire nu permite:

- a. organizarea și facilitarea polimorfismului și încapsulării permițând definirea și crearea unor clase specializate plecând de la clase (generale) care sunt deja definite;
- b. definirea unei relații între clase în care o clasă împărtășește structura și comportarea definită în una sau mai multe clase;
- c. o ierarhie de tip generalizare/specializare, în care clasa derivată specializează structura și comportamentul mai general al clasei din care a fost derivată;
- d. construirea unor clase noi, care păstrează caracteristicile și comportarea;
- e. relația între două obiecte în care unul dintre obiecte aparține celuiilalt obiect.

7. Programarea orientată-obiect presupune:

- a. ordonarea obiectelor și a acțiunilor din lumea reală sub formă de clase care pot fi create, manevrate și distruse;
- b. o parte structurată de date (definită de tipuri și variabile), o parte operațională (definită prin proceduri) și o parte de program principal care apelează procedurile;
- c. ca o modificare într-o structură de date să se repercuteze în toate programele care le manipulează;
- d. regruparea procedurilor legate semantic în module care importă sau exportă proceduri de la/către alte module;
- e. multe date globale.

8. Un program orientat obiect:

- a. este un ansamblu de obiecte care printr-un schimb de mesaje declanșează anumite operații sau metode, facilitându-le stările lor interne și returnându-le parametrii;
- b. presupune comportarea într-o manieră strictă și imposibil de prevăzut;
- c. nu conduce la o corectă structurare și modularizare a programului;
- d. nu se poate utiliza același nume de funcție pentru efectuarea de acțiuni similare la obiecte diferite;
- e. prin crearea în mod vizual a unei singure linii de cod, nu se poate obține un program funcțional.

9. Între avantajele utilizării POO nu se poate înscrie:

- a. ușurința proiectării și reutilizării codului;
- b. siguranța datelor;
- c. conduce la formarea rapidă a unor concepte globale de funcționare a metodei elementului finit;
- d. interfața este separată de implementare;
- e. separarea aduce un minus de flexibilitate pentru implementatori.

10. Visual C++ prezintă și alte avantaje față de celelalte limbaje obiectuale, printre care nu se regăsește:

- a. posibilitatea de interfațare cu numeroase programe importante;
- b. C++ produce un cod cu un timp de execuție foarte eficient;
- c. legăturile cu sistemul de operare Unix (datorate limbajului C);
- d. existența multor compilatoare, inclusiv a celor gratuite;
- e. produce implementări, pe același tip de platformă.

11. Elementele caracteristice ale conceptului de clasă se pot concretiza în:

- a. într-o clasă nu sunt precizate denumirile variabilelor;
- b. în interiorul unei clase sunt precizate numai denumirile tuturor variabilelor;
- c. o clasă este schema, planul după care este creat un obiect;
- d. o clasă nu trebuie să asigure o anumită interfață, utilizată pentru manipularea obiectelor existente în acea clasă;
- e. detaliile de implementare sunt la dispoziția utilizatorului.

12. O clasă nu are următoarele componente:

- a. declarația clasei și implementarea clasei;
- b. declarațiile atributelor și funcțiilor membru clasei, codul sau implementarea;
- c. interfața clasei și informații privind datele membre ale respectivei clase;
- d. toate funcțiile membru care au fost declarate ca parte a clasei;
- e. informații privind datele membre ale tuturor claselor.

13. Există următoarele modalități de organizare a claselor:

- a. organizarea claselor după modelul unui arbore genealogic, cu clase care moștenesc datele și funcțiile de la predecesorii lor;
- b. organizarea claselor care moștenesc doar datele nu și funcțiile de la predecesorii lor;
- c. un set de clase se folosesc împreună și prezintă un fel de independență între ele;
- d. clasele derivate nu au proprietățile generale ale clasei de bază;
- e. clasele între care nu există legături strânse sunt adesea grupate în categorii.

14. Membrii claselor care sunt declarați în secțiunea publică au următoarele caracteristici:

- a. constituie implementarea serviciilor pe care o clasă nu le oferă clienților săi;
- b. sunt utilizați în primul rând pentru informațiile stării unui obiect;
- c. sunt accesibili tuturor obiectelor și funcțiilor;
- d. conține declarații ale datelor ce reprezintă atributele unui obiect;
- e. conține declarațiile funcțiilor membru care nu sunt oferite spre utilizare clienților clasei.

15. Fișierul header nu trebuie să conțină următoarele informații:

- a. descriere a ceea ce face clasa;
- b. structura clasei, cea care este implementată în porțiunea privată;
- c. numele fișierului nu trebuie neapărat să reprezinte clasa;
- d. funcțiile membru împărțite pe categorii;
- e. comentarii tactice și strategice ale funcțiilor membru și grupurile de funcții membru, chiar și de atribute.

16. Utilitatea constructorilor este evidentă din cel puțin următoarele aspecte:

- a. este apelat automat la eliminarea unui obiect, la încheierea timpului sau de viață, sau poate fi solicitat prin program;
- b. constructorul asigură inițializarea corectă a tuturor variabilelor membru ale unui obiect;
- c. este declarat ca neavând tip de valoare returnată și nu este declarat niciodată cu o listă de parametri;
- d. constructorul nu oferă o garanție că inițializarea unui obiect se va realiza exact o dată;
- e. constructorul nu asigură inițializarea tuturor variabilelor membru ale unui obiect.

17. Dintre caracteristicile specifice constructorilor și destructorilor nu se regăsesc:

- a. constructorii sunt lansați în ordinea declarării obiectelor iar destructorii în ordine inversă;
- b. se pot utiliza pointeri către constructori sau destructori;
- c. numele funcțiilor constructor sau destructor coincide cu numele clasei căreia îi aparțin;
- d. în declarație și definiție nu se specifică nici un tip de rezultat, nici măcar void;
- e. dacă o clasă nu dispune de constructori și destructori definiți, compilatorul va genera automat un constructor implicit, respectiv un destructor, funcții publice.

18. Procedul de supraîncărcare constă în:

- a. definirea unei funcții cu numele operator;
- b. efectuarea mai multor operații matematice;

- c. furnizarea unei notații neconvențională și mai convenabilă pentru a manipula obiectele unei clase;
- d. furnizarea unei notații mai puțin convențională și neconvenabilă pentru a manipula obiectele unei clase;
- e. operarea asupra mai multor tipuri de bază.

19. O funcție membru virtuală:

- a. nu trebuie să posede un corp;
- b. compilatorul construiește un tabel special, denumit tabel de funcții, care se folosește pentru a păstra evidența funcțiilor care trebuie apelate pentru fiecare obiect din clasa respectivă;
- c. o funcție nu trebuie precizată ca fiind virtuală, deoarece acest caracter este prestabilit;
- d. când este apelată o funcție virtuală, tabelul de funcții este folosit pentru accesarea directă a funcției corecte;
- e. utilizarea unei astfel de funcții nu permite implementarea polimorfismului.

20. Tabelul de funcții virtuale se folosește pentru:

- a. pentru a păstra evidența funcțiilor care trebuie apelate pentru fiecare obiect din clasa respectivă;
- b. manipularea obiectelor care sunt reprezentări concrete ale unor concepte abstracte;
- c. furnizează o facilitate de reprezentare a obiectelor neprimitive împreună cu un set de operații care pot fi efectuate;
- d. definirea unei funcții cu numele operator simbol;
- e. accesarea directă a funcției corecte.

21. Caracteristicile claselor de date abstracte:

- a. clasele de bază abstracte ca orice clase pot avea instanțe;
- b. pot ține locul oricăreia din clasele derivate;
- c. este o clasă care nu are nici o funcție membru virtuală pură;
- d. nu este posibilă derivarea de clase din cele abstracte;
- e. când se proiectează un produs software utilizând tehnicile orientate obiect nu se va utiliza nici o clasă abstractă.

22. Unul din obiectivele moștenirii este:

- a. este acela de a partaja un cod între mai multe clase;
- b. scăderea factorului de reutilizare al codului;
- c. dacă o operație poate fi executată utilizând o clasă de bază, nu este obligatoriu ca toate subclasele să fie capabile să execute acea operație;
- d. clasele derivate, nu conțin datele și funcții membru care sunt incluse în clasele de bază;
- e. conduce la scăderea productivității programării orientate obiect.

23. Printre regulile de bază pentru obținerea ierarhiilor de clase nu se regăsește:

- a. alegerea unei clase abstracte de bază pentru a servi drept clasă de bază tuturor celorlalte clase (abstracte sau nu);
- b. este posibilă crearea unei clase de bază abstracte prin derivarea dintr-o altă clasă de bază abstractă;
- c. clasele concrete sunt proiectate pentru a fi instantiate;
- d. niciodată nu se va deriva o clasă abstractă dintr-una concretă;
- e. **întotdeauna se va deriva o clasă abstractă dintr-una concretă.**

24. În ceea ce privește domeniul unei variabile nu este valabilă informația:

- a. se referă la vizibilitatea acesteia într-un program;
- b. dacă o variabilă se află în domeniu într-un anumit punct din program, atunci este vizibilă și poate fi folosită în majoritatea cazurilor;

- c. dacă variabila se află în afara domeniului atunci este vizibilă și programul o va putea folosi;
- d. un nume este vizibil în întregul său domeniu de definiție dacă nu este redefinit într-un bloc inclus în domeniul respectiv;
- e. domeniul unui identificador intră în acțiune la fiecare declarare a unui identificador.

25. Biblioteca de clase MFC poate fi împărțită pe trei nivele:

- a. la al treilea nivel se află clasele care descriu comportamentul obiectelor grafice și încapsulează funcțiile API (Application Program Interface);
- b. la al doilea nivel se află clasele care nu depind de sistemul de operare Windows și care implementează structuri fundamentale de date;
- c. la al doilea nivel se află clasele care descriu comportamentul obiectelor grafice și încapsulează funcțiile API (Application Program Interface);
- d. la primul nivel se află clasele care nu depind de sistemul de operare Windows și care implementează structuri fundamentale de date;
- e. la primul nivel este o colecție de componente soft (clase, macro-uri, funcții globale) care pot fi utilizate în realizarea unei aplicații generice

26. CObject nu asigură următoarele tipuri de servicii:

- a. diagnosticarea gestiunii memoriei furnizează mesaje de diagnostic la detectarea unor pierderi de memorie (memory leak);
- b. suportul creării dinamice folosește clasa CRuntimeClass pentru a permite crearea obiectelor la rulare programului;
- c. suportul de serializare permite stocarea și încărcarea unui obiect într-o modalitate orientată-obiect;
- d. informația de clasă runtime (în timpul execuției);
- e. suportul de serializare nu permite stocarea și încărcarea unui obiect într-o modalitate orientată-obiect.

27. Clasa de bază CWnd, este derivată din CObject și oferă:

- a. un grad mare de funcționalitate, propriu tuturor ferestrelor dintr-un program MFC;
- b. nu este utilizată ca un pointer generic de fereastră;
- c. nu conține funcții suplimentare ce ar putea fi aplicate majorității ferestrelor;
- d. o implementare a obiectului șiruri tip matrice și a metodelor aferente acestuia;
- e. creează obiecte capabile de a stoca alte obiecte.

28. Clasa CString:

- a. este utilizată în mod normal ca un pointer generic de fereastră;
- b. este folosită în mod normal drept clasă de bază atunci când programatorul își creează propriile clase;
- c. este folosită în mod normal drept clasă de bază atunci când programatorul își creează propriile clase;
- d. este adeseori transferată drept parametru de funcție și este folosită ca pointer generic pentru orice tip de fereastră într-un program MFC;
- e. creează șiruri inteligente care pot fi cu ușurință adunate, copiate sau manevrate în alt mod.

29. Clasele tip colecție CList:

- a. este utilizată pentru a crea colecții care sunt mai utile decât matricele încorporate în C++;
- b. este folosită pentru crearea colecțiilor care se indexează folosind o cheie unică pentru fiecare articol stocat în colecție;
- c. este ideală pentru stocarea articolelor care dețin numere de identitate sau de serie unice;
- d. este folosită atunci când o colecție de articole este utilizată ca un lanț de articole înrudite;
- e. permite dezvoltarea dinamică a matricei la adăugarea de elemente noi.

30. *Clasa tip colecție CMap*

- a. este utilizată pentru a crea colecții care sunt mai utile decât matricele încorporate în C++;
- b. permite adăugarea de articole la începutul și la sfârșitul listei;
- c. permite dezvoltarea dinamică a matricei la adăugarea de elemente noi, aspect imposibil în cadrul matricelor încorporate;
- d. este folosită pentru crearea colecțiilor care se indexează folosind o cheie unică pentru fiecare articol stocat în colecție;
- e. își stochează articolele într-o listă tip lanț, ceea ce facilitează în mare măsură adăugarea sau eliminarea articolelor, indiferent de amplasarea acestora.

31. *Bazele de date obiectuale:*

- a. au la bază tipuri de date simple, recunoscute (caractere, șiruri, întregi etc.) și nu permit crearea unor noi tipuri de date;
- b. operează cu tipurile de date la un nivel mai înalt, permițând crearea prin definiție a unor noi tipuri de date;
- c. un produs software care stochează datele într-o structură bine organizată și oferă mijloace eficiente de accesare și actualizare a acestor date;
- d. operează cu tipurile de date la un nivel mai înalt, și nu permit crearea unor noi tipuri de date;
- e. operează cu tipurile de date simple, recunoscute (caractere, șiruri, întregi etc.), permițând crearea prin definiție a unor noi tipuri de date

32. *Cel mai important element privind obiectele COM îl reprezintă:*

- a. interfețele acestora;
- b. legarea și încapsularea obiectelor;
- c. controale ActiveX;
- d. obiectele propriu-zise;
- e. comportamentul obiectelor.

33. *Legarea și încapsularea obiectelor (OLE):*

- a. descrie capacitatea de a insera obiecte de același tip în documente având un tip propriu;
- b. obiectele inserate nu pot fi încapsulate, ceea ce înseamnă că nu pot fi salvate odată cu documentul, sau legate;
- c. descrie capacitatea de a insera obiecte de diferite tipuri în documente având un tip propriu;
- d. un program nu poate să apeleze metodele unui alt program care rulează ascuns, fără ca utilizatorul să fie conștient de această interacțiune;
- e. un program poate să apeleze metodele unui alt program care rulează ascuns, cu acceptarea utilizatorului pentru această interacțiune.

34. *Automatizarea OLE presupune:*

- a. un program poate să apeleze metodele unui alt program care rulează ascuns, fără ca utilizatorul să fie conștient de această interacțiune;
- b. obiectele inserate nu pot fi salvate odată cu documentul;
- c. capacitatea de a insera obiecte de același tip în documente;
- d. un program care apelează o funcție declarată într-o interfață va găsi o implementare ce efectuează operația definită de acea funcție în cadrul interfeței respective;
- e. imposibilitatea punerii la dispoziție a unei interfețe dispencer.

35. *Caracteristicile unui server OLE se pot concretiza în:*

- a. nu permit obiectelor încapsulate să controleze zona client a containerului și să manipuleze direct fereastra, meniurile și barele cu instrumente ale acestuia;
- b. este o aplicație care poate fi lansată în cadrul unei ferestre a unei aplicații container;

- c. nu pot manipula direct fereastra, meniurile și barele cu instrumente;
- d. aplicațiile nu pot fi concomitent containere OLE și servere OLE;
- e. nu se poate implementa suport pentru operații complexe de editare locală la rularea într-un cadru în fereastra unei aplicații container.

36. Forma generală de declarație a unei clase este următoarea:

- a. *class* nume_clasa
 date și funcții membre private
 specificatori_de_acces
 date și funcții membre
 specificatori_de_acces
 date și funcții membre

 specificatori_de_acces
 date și funcții membre
 } lista_obiecte;
- b. *class* nume_clasa{
 date și funcții membre private
 specificatori_de_acces
 date și funcții membre
 specificatori_de_acces
 date și funcții membre

 specificatori_de_acces
 date și funcții membre
 } lista_obiecte;
- c. *class* nume_clasa{
 date și funcții membre
 specificatori_de_acces
 date și funcții membre

 specificatori_de_acces
 date și funcții membre
 } lista_obiecte;
- d. *class* nume_clasa{
 date și funcții membre private
 specificatori_de_acces
 lista_obiecte;
- e. *class* nume_clasa{
 date și funcții membre private
 specificatori_de_acces
 date și funcții membre
 specificatori_de_acces
 date și funcții membre

 specificatori_de_acces
 date și funcții membre
 lista_obiecte;

37. Pentru accesarea membrilor unei clase se poate utiliza următoarea secvență de instrucțiuni:

```

a. int main()
{
    stiva Stiva1; // apel stiva::stiva() pentru a crea obiectul Stiva1
    Stiva1.adauga( 5.0 );
    stiva Stiva2; // apel stiva::stiva() pentru a crea obiectul Stiva2
    Stiva2.adauga( 3.3 );
    Stiva1.adauga( 9.9 );
    cout << Stiva1.sterge() << endl;
    return 0;
}

b. int main()
{
    stiva Stiva1; // apel stiva::stiva() pentru a crea obiectul Stiva1
    Stiva1.adauga( 5.0 );
    stiva Stiva2; // apel stiva::stiva() pentru a crea obiectul Stiva2
    Stiva2.adauga( 3.3 );
    Stiva1.adauga( 9.9 );
    return 0;
}

c. int main()
    stiva Stiva1; // apel stiva::stiva() pentru a crea obiectul Stiva1
    Stiva1.adauga( 5.0 );
    stiva Stiva2; // apel stiva::stiva() pentru a crea obiectul Stiva2
    Stiva2.adauga( 3.3 );
    Stiva1.adauga( 9.9 );
    cout << Stiva1.sterge() << endl;
    return 0;

d. int main()
{
    cout << Stiva1.sterge() << endl;
    return 0;
}

e. int main()
{
    stiva Stiva1; // apel stiva::stiva() pentru a crea obiectul Stiva1
    Stiva1.adauga( 5.0 )
    stiva Stiva2; // apel stiva::stiva() pentru a crea obiectul Stiva2
    Stiva2.adauga( 3.3 )
    Stiva1.adauga( 9.9 )
    cout << Stiva1.sterge() << endl
    return 0;
}

```

38. Utilizând AppWizard Subclasarea claselor (crearea unei noi clase prin moștenirea atributelor și funcțiilor membru ale unei clase deja existente) presupune parcurgerea următorilor pași:

- se creează o nouă clasă utilizând ClassWizard, se va adăuga directiva #include menționată în ClassWizard, se utilizează noua clasă pentru a implementa operațiile dorite;
- se creează o nouă clasă utilizând ClassWizard, se utilizează noua clasă pentru a implementa operațiile dorite.
- se creează o nouă clasă prin adăugarea directivei #include, si se utilizează noua clasă pentru a implementa operațiile dorite.

- d. se va adăuga directiva **#include** menționată în ClassWizard, se utilizează noua clasă pentru a implementa operațiile dorite.
- e. se creează o nouă clasă și se va adăuga directiva **#include corespunzătoare**.

39. Crearea unei noi clase, utilizând ClassWizard, presupune parcurgerea următorilor pași:

- a.
 - se alege pagina Message Maps din ClassWizard;
 - se va introduce numele clasei noi create în caseta Name, se selectează tipul clasei de bază din caseta combinată Base Class, după care se efectuează clic pe OK;
 - se selectează clasa specifică casetei de dialog deschisă din caseta combinată Class Name;
 - din cadrul casetei cu listă Object IDs, se selectează IDC specific clasei nou create;
 - se dă clic pe butonul Member Variables, fiind afișată caseta de dialog Add Member Variable;
 - se introduce `m_` nume variabilă în cadrul câmpului Member Variable Name;
 - se selectează Control, din caseta combinată Category;
 - se alege clasa nou creată în caseta combinată Variable Type, în partea inferioară a casetei de dialog va trebui să apară afișat mesajul Map to nume clasa nou creată Member (Used-Defined-Class);
 - se alege OK pentru a adăuga noua variabilă membru, afișându-se un mesaj prin care se informează utilizatorul că este necesară adăugarea unei directive **#include**, după care se dă OK;
 - se finalizează prin selectarea lui OK pentru a se închide caseta de dialog ClassWizard.
- b.
 - se alege pagina Message Maps din ClassWizard;
 - se selectează numele clasei unde dorim să-i adăugăm o altă clasă din caseta combinată ClassName, și se va efectua un clic pe butonul Add Class, apoi se va selecta New din meniul nou apărut, apărând caseta de dialog New Class;
 - din cadrul casetei cu listă Object IDs, se selectează IDC specific clasei nou create;
 - se dă clic pe butonul Member Variables, fiind afișată caseta de dialog Add Member Variable;
 - se introduce `m_` nume variabilă în cadrul câmpului Member Variable Name;
 - se selectează Control, din caseta combinată Category;
 - se alege clasa nou creată în caseta combinată Variable Type, în partea inferioară a casetei de dialog va trebui să apară afișat mesajul Map to nume clasa nou creată Member (Used-Defined-Class);
 - se alege OK pentru a adăuga noua variabilă membru, afișându-se un mesaj prin care se informează utilizatorul că este necesară adăugarea unei directive **#include**, după care se dă OK;
 - se finalizează prin selectarea lui OK pentru a se închide caseta de dialog ClassWizard.
- c.
 - se alege pagina Message Maps din ClassWizard;
 - se selectează numele clasei unde dorim să-i adăugăm o altă clasă din caseta combinată ClassName, și se va efectua un clic pe butonul Add Class, apoi se va selecta New din meniul nou apărut, apărând caseta de dialog New Class;
 - se va introduce numele clasei noi create în caseta Name, se selectează tipul clasei de bază din caseta combinată Base Class, după care se efectuează clic pe OK;
 - se selectează clasa specifică casetei de dialog deschisă din caseta combinată Class Name;
 - din cadrul casetei cu listă Object IDs, se selectează IDC specific clasei nou create;
 - se dă clic pe butonul Member Variables, fiind afișată caseta de dialog Add Member Variable;
 - se introduce `m_` nume variabilă în cadrul câmpului Member Variable Name;
 - se selectează Control, din caseta combinată Category;
 - se alege clasa nou creată în caseta combinată Variable Type, în partea inferioară a casetei de dialog va trebui să apară afișat mesajul Map to nume clasa nou creată Member (Used-Defined-Class);
 - se alege OK pentru a adăuga noua variabilă membru, afișându-se un mesaj prin care se informează utilizatorul că este necesară adăugarea unei directive **#include**, după care se dă OK;

- se finalizează prin selectarea lui OK pentru a se închide caseta de dialog ClassWizard.
- d.
- se alege pagina Message Maps din ClassWizard;
 - se selectează numele clasei unde dorim sa-i adăugăm o altă clasă din caseta combinată ClassName, și se va efectua un clic pe butonul Add Class, apoi se va selecta New din meniul nou apărut, apărând caseta de dialog New Class;
 - se va introduce numele clasei noi create în caseta Name, se selectează tipul clasei de bază din caseta combinată Base Class, după care se efectuează clic pe OK;
 - se selectează clasa specifică casetei de dialog deschisă din caseta combinată Class Name;
 - se introduce m_nume variabilă în cadrul câmpului Member Variable Name;
 - se selectează Control, din caseta combinată Category;
 - se alege clasa nou creată în caseta combinată Variable Type, în partea inferioară a casetei de dialog va trebui să apară afișat mesajul Map to nume clasa nou creată Member (Used-Defined-Class);
 - se alege OK pentru a adăuga noua variabilă membru, afișându-se un mesaj prin care se informează utilizatorul că este necesară adăugarea unei directive #include, după care se dă OK;
 - se finalizează prin selectarea lui OK pentru a se închide caseta de dialog ClassWizard.
- e.
- se alege pagina Message Maps din ClassWizard;
 - se selectează numele clasei unde dorim sa-i adăugăm o altă clasă din caseta combinată ClassName, și se va efectua un clic pe butonul Add Class, apoi se va selecta New din meniul nou apărut, apărând caseta de dialog New Class;
 - se va introduce numele clasei noi create în caseta Name, se selectează tipul clasei de bază din caseta combinată Base Class, după care se efectuează clic pe OK;
 - se selectează clasa specifică casetei de dialog deschisă din caseta combinată Class Name;
 - din cadrul casetei cu listă Object IDs, se selectează IDC specific clasei nou create;
 - se finalizează prin selectarea lui OK pentru a se închide caseta de dialog ClassWizard.

40. Crearea și configurarea unei surse de date ODBC se desfășoară în următorii pași:

- a.
- în cadrul meniului Start se deschide submeniul Settings și se selectează apoi Control Panel.
 - se efectuează dublu clic pe pictograma etichetată 32bit ODBC; va fi afișată caseta de dialog ODBC DataSource Administrator;
 - se va selecta pagina User DSN și se dă clic pe Add, fiind deschisă caseta de dialog Create New data Source, putându-se modifica opțiunile asociate unei surse de date existente selectând numele său din lista User data Sources și efectuând un clic pe butonul Configure;
 - se va selecta numele driverului ODBC pentru care se dorește a se crea o sursă de date (acesta trebuie să corespundă tipului bazei de date). Presupunem ca avem o bază de date creată într-un SGBD cu numele Catalog Studentilor.
 - se introduce un nume pentru sursa de date în cadrul casetei de editare data Source Name, (Catalogul studenților), putându-se specifica orice nume, dar trebuie evitată existența a două surse de date având același nume;
 - opțional se introduce în caseta de editare Description o descriere pentru sursa de date;
 - se efectuează un clic pe Select, fiind afișată caseta de dialog Select Database;
 - se introduce numele fișierului bază de date în caseta de editare Database Name sau se selectează fișierul cu ajutorul listei de dosare Directories, și se dă OK; locația și numele bazei de date sunt afișate în caseta de dialog Setup;
 - se efectuează clic pe OK pentru a închide caseta de dialog ODBC; în lista User Data Sources ar trebui să apară acum catalogul studenților;
 - pentru a se închide caseta de dialog ODBC Data Source Administrator se alege OK și se închide apoi și fereastra panoului de control.

b.

- în cadrul meniului Start se deschide submeniul Settings și se selectează apoi Control Panel.
- se efectuează dublu clic pe pictograma etichetată 32bit ODBC; va fi afișată caseta de dialog ODBC DataSource Administrator;
- se introduce în caseta de editare Description o descriere pentru sursa de date;
- se efectuează un clic pe Select, fiind afișată caseta de dialog Select Database;
- se introduce numele fișierului bază de date în caseta de editare Database Name sau se selectează fișierul cu ajutorul listei de dosare Directories, și se dă OK; locația și numele bazei de date sunt afișate în caseta de dialog Setup;
- se efectuează clic pe OK pentru a închide caseta de dialog ODBC;
- pentru a se închide caseta de dialog ODBC Data Source Administrator se alege OK și se închide apoi și fereastra panoului de control.

c.

- în cadrul meniului Start se deschide submeniul Settings și se selectează apoi Control Panel.
- se efectuează dublu clic pe pictograma etichetată 32bit ODBC; va fi afișată caseta de dialog ODBC DataSource Administrator;
- se introduce un nume pentru sursa de date în cadrul casetei de editare data Source Name, (Catalogul studenților), putându-se specifica orice nume, dar trebuie evitată existența a două surse de date având același nume;
- opțional se introduce în caseta de editare Description o descriere pentru sursa de date;
- se efectuează un clic pe Select, fiind afișată caseta de dialog Select Database;
- se efectuează clic pe OK pentru a închide caseta de dialog ODBC; în lista User Data Sources ar trebui să apară acum catalogul studenților;
- pentru a se închide caseta de dialog ODBC Data Source Administrator se alege OK și se închide apoi și fereastra panoului de control.

d.

- în cadrul meniului Start se deschide submeniul Settings și se selectează apoi Control Panel.
- se efectuează dublu clic pe pictograma etichetată 32bit ODBC; va fi afișată caseta de dialog ODBC DataSource Administrator;
- se va selecta pagina User DSN și se dă clic pe Add, fiind deschisă caseta de dialog Create New data Source, putându-se modifica opțiunile asociate unei surse de date existente selectând numele său din lista User data Sources și efectuând un clic pe butonul Configure;
- se va selecta numele driverului ODBC pentru care se dorește a se crea o sursă de date (acesta trebuie să corespundă tipului bazei de date). Presupunem ca avem o bază de date creată într-un SGBD cu numele Catalogul Studentilor;
- se introduce un nume pentru sursa de date în cadrul casetei de editare data Source Name, (Catalogul studenților), putându-se specifica orice nume, dar trebuie evitată existența a două surse de date având același nume;
- opțional se introduce în caseta de editare Description o descriere pentru sursa de date;
- se efectuează un clic pe Select, fiind afișată caseta de dialog Select Database;
- se introduce numele fișierului bază de date în caseta de editare Database Name sau se selectează fișierul cu ajutorul listei de dosare Directories, și se dă OK; locația și numele bazei de date sunt afișate în caseta de dialog Setup;
- se efectuează clic pe OK pentru a închide caseta de dialog ODBC; în lista User Data Sources ar trebui să apară acum catalogul studenților;
- pentru a se închide caseta de dialog ODBC Data Source Administrator se alege OK și se închide apoi și fereastra panoului de control.

e.

- se va selecta pagina User DSN și se dă clic pe Add, fiind deschisă caseta de dialog Create New data Source, putându-se modifica opțiunile asociate unei surse de date existente selectând numele său din lista User data Sources și efectuând un clic pe butonul Configure;

- se va selecta numele driverului ODBC pentru care se dorește a se crea o sursă de date (acesta trebuie să corespundă tipului bazei de date). Presupunem ca avem o bază de date creată într-un SGBD cu numele Catalogul Studentilor;
- se introduce un nume pentru sursa de date în cadrul casetei de editare data Source Name, (Catalogul studenților), putându-se specifica orice nume, dar trebuie evitată existența a două surse de date având același nume;
- se efectuează clic pe OK pentru a închide caseta de dialog ODBC; în lista User Data Sources ar trebui să apară acum catalogul studenților;
- pentru a se închide caseta de dialog ODBC Data Source Administrator se alege OK și se închide apoi și fereastra panoului de control.

GRILE STRUCTURI DE DATE

41). Tipul unei constante rezultă din:

- a) forma sintactică (modul de scriere)
- b) specificarea în program a unui enunț special – declarație
- c) tipurile operanzilor și modul lor de compunere cu ajutorul operatorilor.
- d) declarația funcției (în C - prototip sau semnătură)
- e) tipul de limbaj de programare folosit

42). Tipul unei expresii rezultă din:

- a) forma sintactică (modul de scriere)
- b) specificarea sa în program cu un enunț special – declarație
- c) tipurile operanzilor și modul lor de compunere cu ajutorul operatorilor.
- d) declarația funcției (în C - prototip sau semnătură)
- e) tipul de limbaj de programare folosit

43). Tipul unei variabile rezultă din:

- a) forma sintactică (modul de scriere)
- b) specificarea sa în program cu un enunț special – declarație
- c) tipurile operanzilor și modul lor de compunere cu ajutorul operatorilor.
- d) declarația funcției (în C - prototip sau semnătură)
- e) tipul de limbaj de programare folosit

44). Tipul unei funcții rezultă din:

- a) forma sintactică (modul de scriere)
- b) specificarea în program a numelui funcției
- c) tipurile operanzilor și modul lor de compunere cu ajutorul operatorilor.
- d) declarația sa (în C - prototip sau semnătură)
- e) tipul de limbaj de programare folosit

45). Un tip de date este ordinal dacă:

- a) pe mulțimea elementelor sale se poate defini o relație de ordonare liniară
- b) pe mulțimea elementelor sale nu se poate defini o relație de ordonare liniară
- c) pe mulțimea valorilor sale se poate defini o relație de ordine
- d) pe mulțimea valorilor sale nu se poate defini o relație de ordine
- e) modelează o mulțime infinită de elemente

46). O structură de date reprezintă:

- a) numărul de valori distincte aparținând tipului de date
- b) mulțimea de obiecte care alcătuiesc un anumit tip, proprietățile obiectelor, relațiile dintre obiecte și operațiile asupra obiectelor
- c) o mulțime de elemente numite valori sau obiecte, care pot clar distinge între ele
- d) mulțimea obiectelor definite într-un program
- e) mulțimea tuturor datelor permise de un limbaj de programare

47). Cardinalitatea unui tip de date reprezintă:

- a) mulțimea de obiecte care alcătuiesc un anumit tip, proprietățile obiectelor, relațiile dintre obiecte și operațiile asupra obiectelor
- b) o mulțime de elemente numite valori sau obiecte, care pot clar distinge între ele
- c) mulțimea obiectelor definite într-un program
- d) mulțimea tuturor datelor permise de un limbaj de programare
- e) numărul de valori distincte aparținând tipului de date

48). Constructorii se utilizează atunci când:

- a) se generează noi obiecte aparținând structurii respective
- b) se eliberează memoria alocată obiectelor
- c) se accesează elementele care compun un obiect compus
- d) se actualizează elementele care compun un obiect compus
- e) se compară obiectele aparținând structurii respective

49). Modificatorii se utilizează atunci când:

- a) se generează noi obiecte aparținând structurii respective
- b) se eliberează memoria alocată obiectelor
- c) se accesează elementele care compun un obiect compus
- d) se actualizează elementele care compun un obiect compus
- e) se compară obiectele aparținând structurii respective

50). Care dintre tipurile de date compuse au cardinalitatea egală cu produsul cardinalităților tipurilor componente:

- a) tipul agregat, tipul tablou și tipul cu variante
- b) tipul mulțime
- c) tipul cu variante
- d) tipul agregat și tipul tablou
- e) tipul caracter

51). În cazul alocării dinamice a memoriei:

- a) obiectele sunt referite în program prin identificatorul introdus la declararea lor
- b) obiectele sunt referite prin declararea în cadrul unei clase
- c) obiectele sunt referite prin utilizarea tipului referință
- d) obiectele sunt referite prin definirea unor funcții
- e) nicio variantă nu este corectă

52). În cazul alocării statice a memoriei:

- a) obiectele sunt referite în program prin identificatorul introdus la declararea lor
- b) obiectele sunt referite prin declararea în cadrul unei clase
- c) obiectele sunt referite prin utilizarea tipului referință
- d) obiectele sunt referite prin definirea unor funcții
- e) nicio variantă nu este corectă

53). Care din afirmațiile următoare nu este adevărată:

- a) în cazul tipului tablou, componentele unui obiect sunt de același tip
- b) în cazul tipului tablou, componentele unui obiect sunt de tipuri diferite
- c) în cazul tipului agregat, componentele unui obiect sunt de tipuri diferite
- d) în cazul tipului mulțime, nu contează ordinea elementelor
- e) tipul mulțime reprezintă toate submulțimile unui tip de bază

54). Care din afirmațiile următoare nu este adevărată:

- a) în cazul tipului tablou, componentele unui obiect sunt de același tip
- b) în cazul tipului agregat, componentele unui obiect sunt de tipuri diferite
- c) în cazul tipului agregat, componentele unui obiect sunt de același tip
- d) în cazul tipului mulțime, nu contează ordinea elementelor
- e) tipul mulțime reprezintă toate submulțimile unui tip de bază

55). Care din afirmațiile următoare nu este adevărată:

- a) în cazul tipului tablou, componentele unui obiect sunt de același tip
- b) în cazul tipului agregat, componentele unui obiect sunt de tipuri diferite
- c) în cazul tipului mulțime, nu contează ordinea elementelor
- d) în cazul tipului mulțime, este importantă ordinea elementelor
- e) tipul mulțime reprezintă toate submulțimile unui tip de bază

56). Care din afirmațiile următoare nu este adevărată:

- a) în cazul unei cozi, elementul care se citește este primul intrat în coadă
- b) adăugarea unui element nou într-o stivă se face la vârf
- c) în cazul unei cozi, elementul care se citește este ultimul intrat în coadă
- d) adăugarea unui element nou într-o coadă se face la spate
- e) în cazul unei stive, elementul care se citește este ultimul intrat în stivă

57). Gradul unui arbore este:

- a) numărul de subarbori ai unui arbore
- b) maximul gradelor subarborilor corespunzători fiecărui nod al unui arbore
- c) nivelul maxim al nodurilor arborelui
- d) numărul total de noduri ale arborelui
- e) numărul total de noduri terminale (frunze)

58). Aritatea unui arbore este:

- a) numărul de subarbori ai unui arbore
- b) nivelul maxim al nodurilor arborelui
- c) numărul total de noduri ale arborelui
- d) numărul total de noduri terminale (frunze)
- e) maximul gradelor subarborilor corespunzători fiecărui nod al unui arbore

59). Se consideră următoarea codificare Dewey a nodurilor unui arbore:

$\alpha(A)=1, \alpha(A)=1.1, \alpha(C)=1.1.1, \alpha(D)=1.1.2, \alpha(E)=1.1.3, \alpha(F)=1.2$

Parcurserea în adâncime a acestui arbore este:

- a) A B F C D E
- b) A B C D E F
- c) A F B E D C
- d) A B E D C F
- e) F E D C B A

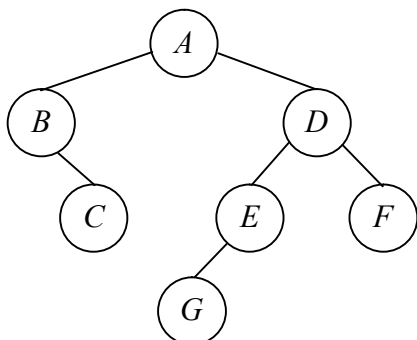
60). Se consideră următoarea codificare Dewey a nodurilor unui arbore:

$\alpha(A)=1, \alpha(A)=1.1, \alpha(C)=1.1.1, \alpha(D)=1.1.2, \alpha(E)=1.1.3, \alpha(F)=1.2$

Parcurserea în lărgime a acestui arbore este:

- a) A B F C D E
- b) A B C D E F
- c) A F B E D C
- d) A F B C D E
- e) F E D C B A

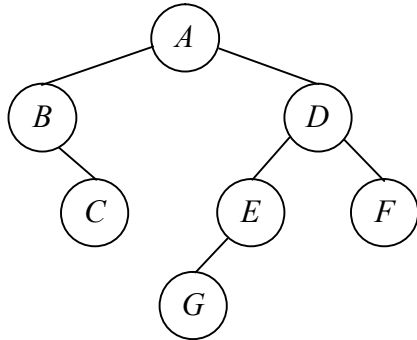
61). Se consideră arborele binar următor:



Parcurgerea în inordine este:

- a) B C A G E D F
- b) C B G E F D A
- c) A B C D E G F
- d) F D G E A C B
- e) F E D C B A

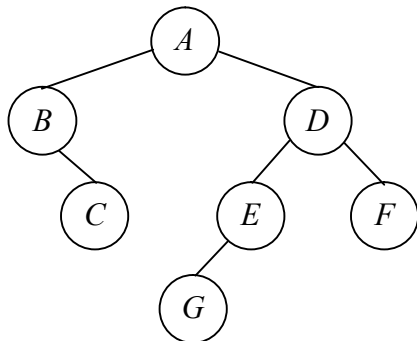
62). Se consideră arborele binar următor:



Parcurgerea în postordine este:

- a) B C A G E D F
- b) C B G E F D A
- c) A B C D E G F
- d) F D G E A C B
- e) G F E D C B A

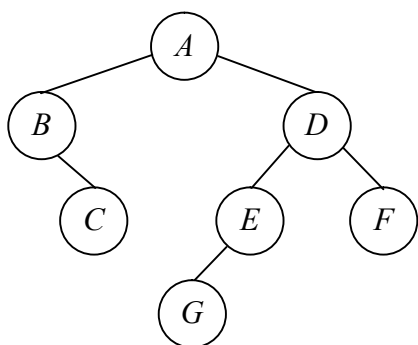
63). Se consideră arborele binar următor:



Parcurgerea în preordine este:

- a) B C A G E D F
- b) C B G E F D A
- c) A B C D E G F
- d) F D G E A C B
- e) G F E D C B A

64). Se consideră arborele binar următor:



Parcurgerea dreapta-rădăcină-stânga este:

- a) B C A G E D F
- b) C B G E F D A
- c) A B C D E G F
- d) F D G E A C B
- e) G F E D C B A

65). Se consideră o stivă s de numere întregi. Să presupunem că se execută următoarea secvență de operații:

inițializează s
 depune 1 în s
 depune 2 în s
 extrage din s
 depune 3 în s
 depune 4 în s

Atunci, conținutul final al stivei s (vârful stivei este cel din stânga) este:

- a) 2 3 4
- b) 1 3 4
- c) 4 3 1
- d) 4 3 2
- e) 1 2 3

66). Se consideră o stivă s de caractere. Să presupunem că se execută următoarea secvență de operații:

inițializează s
 depune x în s
 depune y în s
 extrage din s
 depune z în s
 depune w în s

Atunci, conținutul final al stivei s (vârful stivei este cel din stânga) este:

- a) $y z w$
- b) $x z w$
- c) $w z x$
- d) $w z y$
- e) $x y z$

67). Se consideră o stivă s de numere întregi. Să presupunem că se execută următoarea secvență de operații:

inițializează s
 depune 1 în s
 depune 2 în s
 extrage din s

depune 3 în s

depune 4 în s

Atunci, conținutul final al stivei s (vârful stivei este cel din dreapta) este:

- a) 2 3 4
- b) 4 3 1
- c) 4 3 2
- d) 1 2 3
- e) 1 3 4

68). Se consideră o stivă s de caractere. Să presupunem că se execută următoarea secvență de operații:

inițializează s

depune x în s

depune y în s

extrage din s

depune z în s

depune w în s

Atunci, conținutul final al stivei s (vârful stivei este cel din dreapta) este:

- a) $y z w$
- b) $x z w$
- c) $w z x$
- d) $w z y$
- e) $x y z$

69). Se consideră o coadă c de numere întregi. Să presupunem că se execută următoarea secvență de operații:

inițializează c

depune 1 în c

depune 2 în c

extrage din c

depune 3 în c

depune 4 în c

Atunci, conținutul final al cozii c (fața cozii este în stânga) este:

- a) 2 3 4
- b) 4 3 2
- c) 4 3 1
- d) 1 3 4
- e) 1 2 3

70). Se consideră o coadă c de caractere. Să presupunem că se execută următoarea secvență de operații:

inițializează c

depune x în c

depune y în c

extrage din c

depune z în c

depune w în c

Atunci, conținutul final al cozii c (fața cozii este în stânga) este:

- a) $y z w$
- b) $w z y$
- c) $w z x$
- d) $x z w$

e) x y z

71). Se consideră o coadă c de numere întregi. Să presupunem că se execută următoarea secvență de operații:

**inițializează c
depune 1 în c
depune 2 în c
extrage din c
depune 3 în c
depune 4 în c**

Atunci, conținutul final al cozii c (fața cozii este în dreapta) este:

- a) 2 3 4
- b) 4 3 2
- c) 4 3 1
- d) 1 3 4
- e) 1 2 3

72). Se consideră o coadă c de caractere. Să presupunem că se execută următoarea secvență de operații:

**inițializează c
depune x în c
depune y în c
extrage din c
depune z în c
depune w în c**

Atunci, conținutul final al cozii c (fața cozii este în dreapta) este:

- a) y z w
- b) w z y
- c) w z x
- d) x z w
- e) x y z

73). Un nod dintr-un arbore binar de căutare are:

- a) exact doi copii
- b) doi sau mai multi copii
- c) niciun copil
- d) oricâți copii, depinzând de gradul său
- e) nu mai mult de doi copii

74). Care este numărul maxim de noduri dintr-un arbore binar de înaltime 5?

- a) 15
- b) 31
- c) 63
- d) 64
- e) 32

75). Care este numărul maxim de noduri dintr-un arbore binar de înaltime 6?

- a) 15
- b) 31
- c) 63
- d) 64
- e) 32

76). Se consideră declarațiile:

tip *nod* \equiv (int *info*, ref *nod leg*)

var ref *nod p, q*

și următoarea secvență de instrucțiuni:

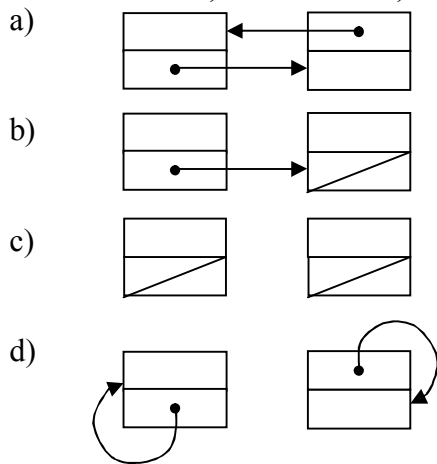
$p \leftarrow \text{nou}(\text{nod})$

$q \leftarrow \text{nou}(\text{nod})$

$\text{deref}(p).\text{leg} \leftarrow p$

$\text{deref}(q).\text{leg} \leftarrow q$

În urma execuției sale, se obține structura:



e) niciuna din cele prezentate

77). În urma execuției secvenței de program următoare:

var *arb a*

$a \leftarrow \text{AdPrimFiu}(\text{Nod2Arb}(\text{NodInfo}('B')),$

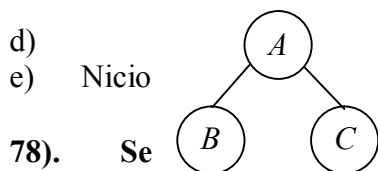
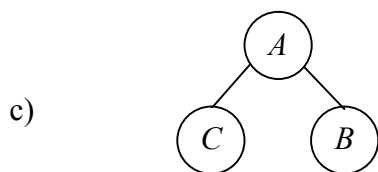
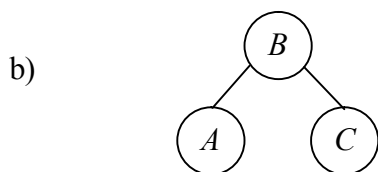
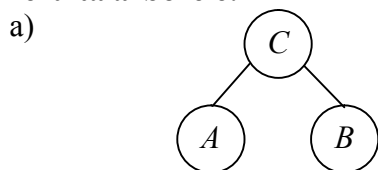
$\text{AdPrimFiu}(\text{Nod2Arb}(\text{NodInfo}('C')),$

$\text{Nod2Arb}(\text{NodInfo}('A'))$

$)$

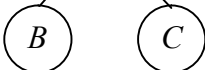
$)$

rezultă arborele:



e) Nicio variantă nu este corectă

78). Se consideră declarațiile:



tip nod \equiv (int *info*, ref *nod succ*, ref *nod pred*)

var ref *nod p, q*

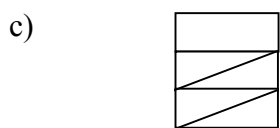
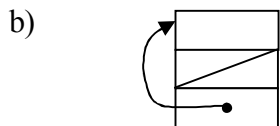
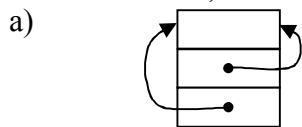
și următoarea secvență de instrucțiuni:

$p \leftarrow \text{nou}(\text{nod})$

$\text{deref}(p).\text{prec} \leftarrow p$

$\text{deref}(p).\text{succ} \leftarrow p$

În urma execuției sale, se obține structura:



e) Nicio variantă nu este corectă

79. Numărul maxim de noduri al unui arbore de aritate 3 și înălțime 4 este:

- a) 13
- b) 21
- c) 26
- d) 40
- e) 12

80. Câți arbori binari de căutare se pot construi cu cheile 10, 20 și 30:

- a) 1
- b) 3
- c) 4
- d) 2
- e) 5

RASPUNSURI CORECTE

Nr. intrebare	Raspuns corect	Punctaj
PROGRAMARE ORIENTATA OBIECT		
1	c	1
2	a	1
3	a	1
4	d	1
5	c	1
6	e	1
7	a	1
8	a	1
9	e	1
10	e	1
11	c	1
12	e	1
13	a	1
14	c	1
15	c	1
16	b	1
17	b	1
18	a	1
19	b	1
20	a	1
21	b	1
22	a	1
23	e	1
24	c	1
25	b	1
26	e	1
27	a	1
28	e	1
29	d	1
30	d	1
31	b	1
32	a	1
33	c	1
34	a	1
35	b	1
36	b	3
37	a	3
38	a	3
39	c	3
40	a	3
STRUCTURI DE DATE		
41	a	1
42	c	1
43	b	1
44	d	1
45	a	1

46	b	1
47	e	1
48	a	1
49	d	1
50	d	1
51	c	1
52	a	1
53	b	1
54	c	1
55	d	1
56	c	1
57	a	1
58	e	1
59	b	1
60	a	1
61	a	1
62	b	1
63	c	1
64	d	1
65	c	1
66	c	1
67	e	1
68	b	1
69	a	1
70	a	1
71	b	1
72	b	1
73	e	1
74	b	1
75	c	1
76	d	3
77	d	3
78	a	3
79	d	3
80	e	3

TESTE GRILĂ SO

1. Un sistem de operare este:
 - a). un program utilizator ce folosește resursele sistemului de calcul;
 - b). o resursă a sistemului de calcul folosită eficient de programele utilizatorilor;
 - c). un mediu utilizator care asigură folosirea eficientă a hardware-ului;
 - d). un set de programe specializate ce asigură legătura funcțională între elementele componente ale unui sistem de calcul, controlând și coordonând utilizarea resurselor acestuia între diferitele programe de aplicații ale utilizatorilor;
 - e). un ansamblu de programe utilizator eficiente.
2. Programele de serviciu:
 - a). coordonează activitatea celorlalte componente ale sistemului de operare;
 - b). sunt destinate optimizării accesului la resursele sistemului de calcul, fiind utilizate de programator pentru dezvoltarea programelor de aplicații;
 - c). planifică lucrările, alocă resursele și țin evidența lucrărilor executate, a utilizatorilor și a resurselor consumate;
 - d). vizează optimizarea utilizării resurselor sistemului de calcul prin coordonarea și controlul tuturor funcțiilor acestuia;
 - e). depistează și tratează evenimentele la execuție (evidența erorilor hardware, evidența întreruperilor etc.).
3. Apelurile supervisor sunt disponibile:
 - a). ca valori ale bitului de mod;
 - b). sub forma de instrucțiuni scrise în limbaj de asamblare;
 - c). ca registre ale unității centrale;
 - d). ca întreruperi;
 - e). ca instrucțiuni privilegiate.
4. Apelurile supervisor au rolul:
 - a). de a genera o cerere de execuție a unui program;
 - b). de a modifica programele utilizatorilor;
 - c). de a realiza transferul din modul supervisor în modul utilizator;
 - d). de a realiza transferul din modul utilizator în modul supervisor;
 - e). de a identifica modul curent de operare.
5. În cadrul modului de operare utilizator:
 - a). pot fi executate toate instrucțiunile;
 - b). pot fi adresate direct dispozitivele de I/E;
 - c). nu pot fi utilizate instrucțiunile care manipulează starea memoriei;
 - d). pot fi utilizate instrucțiuni privilegiate;
 - e). nu pot fi executate instrucțiunile utilizatorilor.
6. Un apel supervisor este tratat de către hardware:
 - a). ca o întrerupere mascabilă;
 - b). ca o întrerupere hardware;
 - c). ca o întrerupere software;
 - d). ca o excepție;
 - e). în niciunul din modurile anterioare.
7. Termenul de excepție este folosit pentru a desemna:
 - a). o întrerupere externă, generată de obicei ca urmare a apariției unei erori de program;

- b). apariția unui apel supervizor;
- c). apariția unui apel utilizator;
- d). o întrerupere hardware internă, generată de obicei ca urmare a apariției unei erori de program;
- e). apariția unei cereri de execuție a unui program.

8. Între cauzele ce pot determina apariția unei excepții se numără:

- a). greșelile de adresare a memoriei;
- b). întreruperea tensiunii de alimentare;
- c). modificarea rezultatului unei împărțiri;
- d). scrierea într-o locație de tip read-write;
- e). citirea dintr-o locație de tip read-write.

9. În cadrul tratării unei întreruperi:

- a). se execută în mod automat un vidaj de memorie și de registre, care poate fi utilizat pentru depanarea programului abandonat;
- b). se salvează starea sistemului, adică se salvează conținutul diverselor regiștri (numărătorul de program, etc.), în așa fel încât să se poată relua execuția programului întrerupt din starea în care se găsea la momentul întreruperii;
- c). dacă întreruperea este nemascabilă, ea va fi ignorată;
- d). se șterge numărătorul de program;
- e). se verifică anumite condiții de integritate.

10. Procesul secvențial este reprezentat de:

- a). un sistem de calcul secvențial împreună cu resursele software ale acestuia;
- b). o dată structurată;
- c). un apel supervizor;
- d). o întrerupere nemascabilă;
- e). un program aflat în execuție împreună cu datele sale și cu celelalte informații necesare execuției, instrucțiunile fiind parcurse una câte una la momente de timp diferite.

11. O stare a unui proces este definită ca fiind:

- a). o stare activă;
- b). o listă;
- c). o rezervă de job-uri;
- d). o cerere de resurse;
- e). o activitate curentă.

12. În cazul unui sistem monoprocessor, precizați care din listele proceselor sunt ordonate:

- a). lista proceselor blocate;
- b). lista proceselor gata de execuție;
- c). lista proceselor suspendate;
- d). toate;
- e). niciuna.

13. Lista proceselor blocate:

- a). este ținută în ordinea priorităților proceselor;
- b). este neordonată;
- c). este ordonată crescător;
- d). este ordonată descrescător;
- e). nu există o astfel de listă.

14. Care este semnificația stării blocate a unui proces?
- procesul urmează să folosească unitatea centrală;
 - procesul folosește în acel moment unitatea centrală;
 - procesul se află în așteptarea realizării unui eveniment;
 - procesul este scos din sistem;
 - procesul este lansat în execuție din punctul în care a fost suspendat.
15. Procesele blocate se vor debloca:
- în ordinea priorităților lor;
 - în ordinea în care se vor finaliza evenimentele ce au determinat blocarea;
 - pe măsura eliberării resurselor ce le-au fost alocate;
 - în ordinea din lista proceselor blocate;
 - în ordinea inversă blocării.
16. Când un proces este creat el este trecut:
- la începutul listei proceselor gata de execuție;
 - la sfârșitul listei proceselor blocate;
 - la începutul listei proceselor ce se execută;
 - la sfârșitul listei proceselor gata de execuție;
 - la sfârșitul listei proceselor ce se execută.
17. Care este semnificația stării gata de execuție a unui proces?
- procesul folosește în acel moment unitatea centrală;
 - procesul se află în așteptarea realizării unui eveniment;
 - procesul are alocate toate resursele de care are nevoie, cu excepția UC-ului.;
 - procesul este scos din sistem;
 - procesul este lansat în execuție din punctul în care a fost suspendat.
18. Starea de tranziție aferentă operației de expediere se indică astfel:
- dispatch (nume_proces): ready → running;
 - timerrunout (nume_proces): running → ready;
 - block (nume_proces): running → ready;
 - dispatch (nume_proces): running → ready;
 - wakeup (nume_proces): blocked → ready.
19. Implementarea unui șir de așteptare (șir ready) al proceselor se realizează printr-o:
- variabilă partajată;
 - variabilă nepartajată;
 - tabelă de mapare;
 - listă înlănțuită;
 - listă neînlănțuită.
20. Când UC devine inactivă, planificatorul selectează pentru execuție unul dintre procesele aflate în:
- șirul proceselor aflate în execuție;
 - șirul ready;
 - șirul proceselor blocate;
 - șirul proceselor suspendate;
 - șirul de dispozitiv I/E.
21. Rolul planificatorului pe termen scurt este acela :
- de a permite efectuarea saltului la locația corespunzătoare din programului utilizator pentru începerea/reluarea execuției acestuia;

- b). de a selecta unul dintre procesele gata de execuție aflate deja în memoria internă și a-i aloca UC;
- c). de a modifica gradul de multiprogramare atunci când este necesar, evacuând din memorie anumite procese, care altfel ar concura pentru dobândirea UC;
- d). de a stabili care sunt procesele ce vor fi încărcate în memoria internă a sistemului pentru a fi executate atunci când există mai multe cereri decât posibilitățile imediate de execuție;
- e). de a comuta un proces în stare gata de execuție.

22. Prin termenul de swapping se înțelege:

- a). comutarea de context;
- b). evacuarea și introducerea din/în memorie a procesului;
- c). ștergerea unui proces;
- d). crearea unui proces;
- e). suspendarea unui proces.

23. Care este durata medie de așteptare a următoarelor 5 procese, sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	10
2	18
3	9
4	18
5	20

în cazul utilizării unui algoritm de planificare de tip FCFS?

- a). 14;
- b). 28;
- c). 26;
- d). 34;
- e). 15.

24. Care este durata medie de așteptare a următoarelor 5 procese, sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	10
2	22
3	15
4	4
5	9

în cazul utilizării unui algoritm de planificare de tip FCFS?

- a). 22,8;
- b). 28;
- c). 25;
- d). 34,5;
- e). 19.

25. Care este durata medie de așteptare a următoarelor 5 procese, sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	20

2	18
3	15
4	10
5	5

în cazul utilizării unui algoritm de planificare de tip FCFS?

- a). 25,4;
- b). 30;
- c). 34,8;
- d). 33,4;
- e). 21.

26. Care este durata medie de așteptare a următoarelor 5 procese, sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	10
2	18
3	9
4	18
5	20

în cazul utilizării unui algoritm de planificare de tip SJF?

- a). 20;
- b). 26;
- c). 34;
- d). 14;
- e). 24.

27. Care este durata medie de așteptare a următoarelor 5 procese, sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	5
2	10
3	14
4	7
5	20

în cazul utilizării unui algoritm de planificare de tip SJF?

- a). 14;
- b). 10,3
- c). 21,5;
- d). 20;
- e). 15.

28. Care este durata medie de așteptare a următoarelor 5 procese, sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	12
2	15
3	9
4	28
5	10

în cazul utilizării unui algoritm de planificare de tip SJF?

- a). 30,4;
- b). 33;
- c). 20,7;
- d). 24,3;
- e). 21.

29. Algoritmul SJF alocă UC, atunci când ea devine disponibilă:

- a). primului proces din șirul ready, conform regulii FIFO;
- b). primelor două procese din șirul ready, conform regulii FCFS;
- c). procesului cu cel mai scurt ciclu "rafală" UC următor, existent în șirul ready;
- d). procesului cu cel mai scurt ciclu "rafală" I/E următor, existent în șirul ready;
- e). procesului cu cel mai lung ciclu "rafală" UC următor, existent în șirul ready;

30. Algoritmul SJF este optimal datorită faptului că:

- a). permite reducerea numărului de procese executate într-un interval de timp precizat;
- b). asigură o durată medie de așteptare minimă oricare ar fi setul de procese luat în considerare;
- c). asigură un timp mediu de răspuns minim;
- d). permite cunoașterea din timp a duratei ciclului "rafală" UC următor;
- e). reduce încărcarea sistemului.

31. În cadrul algoritmului SJF, dacă există două procese cu aceeași durată a ciclului "rafală" UC următor, între ele se aplică regula:

- a). FIFO;
- b). LIFO;
- c). FILO;
- d). FCFS;
- e). Round-Robin.

32. În cadrul algoritmilor de planificare a UC bazați pe priorități, UC va fi alocată în momentul în care devine disponibilă:

- a). procesului cu prioritatea cea mai mare din șirul proceselor blocate;
- b). procesului cu prioritatea cea mai mare din șirul ready;
- c). procesului a cărui prioritate a fost modifică cel mai recent;
- d). procesului cu prioritatea cea mai mică din șirul ready;
- e). procesului cu prioritatea cea mai mică din șirul proceselor blocate.

33. "Îmbătrânirea" proceselor este o tehnică prin care:

- a). se micșorează treptat prioritatea proceselor care se constată că rămân în sistem un timp mai îndelungat;
- b). se mărește treptat prioritatea proceselor care se constată că rămân în sistem un timp mai îndelungat;
- c). se mărește treptat prioritatea proceselor din șirul proceselor blocate;
- d). nu se modifică prioritatea proceselor, ci se micșorează cuanta de timp cât le este alocată UC;
- e). nu se rezolvă problema "înfometării" proceselor..

34. Care este durata medie de așteptare a următoarelor 5 procese sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	10
2	16

3	28
4	15
5	30

în cazul utilizării unui algoritm de planificare de tip Round-Robin? Se consideră valoarea cuantei de timp ca fiind de 10 unități.

- a). 43,2;
- b). 30,1;
- c). 19,4;
- d). 35,6;
- e). 45,4.

35. Care este durata medie de așteptare a următoarelor 5 procese sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	12
2	14
3	20
4	10
5	15

în cazul utilizării unui algoritm de planificare de tip Round-Robin? Se consideră valoarea cuantei de timp ca fiind de 10 unități.

- a). 42,5;
- b). 38,5;
- c). 35,3;
- d). 34,8;
- e). 44,6.

36. Care este durata medie de așteptare a următoarelor 5 procese sosite în șirul ready la același moment de timp, în ordinea numerotării,

Proces	Durata ciclului rafală
1	10
2	20
3	15
4	10
5	8

în cazul utilizării unui algoritm de planificare de tip Round-Robin? Se consideră valoarea cuantei de timp ca fiind de 10 unități.

- a). 30,3;
- b). 28,4;
- c). 31,2;
- d). 29,2;
- e). 33,5.

37. În cazul algoritmului SJF preemptiv, întreruperea execuției procesului curent se realizează dacă:

- a). în șirul ready sosește un proces al cărui ciclul "rafală" UC următor este mai mare decât ceea ce a mai rămas de executat din ciclul "rafală" UC al procesului curent;
- b). în șirul ready sosește un proces al cărui ciclul "rafală" UC următor egal cu ciclul "rafală" UC al procesului curent;
- c). în șirul proceselor blocate sosește un proces al cărui ciclul "rafală" UC următor este mai scurt decât ceea ce a mai rămas de executat din ciclul "rafală" UC al procesului curent;

- d). în șirul ready sosește un proces al cărui ciclu “rafală” UC următor este mai scurt decât ceea ce a mai rămas de executat din ciclul “rafală” UC al procesului curent;
- e). apare un proces limitat I/E.
38. Ce se întâmplă, în cadrul algoritmului Round-Robin, atunci când durata ciclului “rafală” UC a procesului curent este mai mică decât durata cuantei temporale?:
- ceasul va genera o întrerupere și procesul va fi inclus la sfârșitul șirului ready;
 - contextul procesului va fi salvat în blocul de control asociat;
 - procesul va fi limitat I/E;
 - se va modifica durata cuantei temporale;
 - procesul eliberează UC prin emiterea unei cereri de I/E sau prin comunicarea încheierii execuției.
39. Când se utilizează un algoritm de planificare UC pentru șiruri multinivel?:
- când procesele din sistem pot fi clasificate în grupe diferite, în funcție de anumite caracteristici;
 - când se dorește evitarea fenomenului de convoi;
 - când numărul proceselor limitate I/E este mai mic decât cel al proceselor limitate UC;
 - când se dorește limitarea încărcării UC;
 - când nu există un criteriu anume care să stea la baza utilizării unui anumit tip de algoritm de planificare UC.
40. Procesul de compactare a memoriei este cel prin care:
- două sau mai multe zone de memorie ocupate neadiacente se cumulează într-o zonă de dimensiune mai mare;
 - două sau mai multe zone de memorie libere (goluri) neadiacente se cumulează într-un gol de dimensiune mai mare;
 - două sau mai multe zone de memorie libere (goluri) adiacente se cumulează într-un gol de dimensiune mai mare;
 - două sau mai multe zone libere de pe disc se cumulează într-o zonă de dimensiune mai mare;
 - două sau mai multe zone de memorie libere (goluri) și zone libere de pe disc se cumulează într-un gol de dimensiune mai mare.
41. În cadrul cărei strategii de plasare în memorie, job-ului care vine să se execute i se va aloca prima zonă de memorie suficient de mare în care acest poate să încapă?:
- Best Fit;
 - Worst Fit;
 - tuturor strategiilor;
 - First Fit și Best Fit;
 - First Fit.
42. În cazul strategiei de plasare în memorie, First Fit, lista zonelor de memorie liberă este:
- neordonată;
 - ținută în ordinea adreselor;
 - ordonată crescător după dimensiunile golurilor;
 - ordonată crescător după dimensiunile golurilor și în cadrul acesteia descrescător după adrese;
 - ordonată descrescător după dimensiunile golurilor.
43. În cazul strategiei de plasare în memorie, Best Fit, lista zonelor de memorie liberă este:
- ordonată descrescător în ordinea adreselor;
 - ordonată descrescător după dimensiunile golurilor;
 - ordonată crescător în ordinea adreselor;
 - ordonată crescător după dimensiunile golurilor;

- e). ordonată crescător după dimensiunile golurilor și în cadrul acesteia descrescător după adrese.
44. În cazul strategiei de plasare în memorie, Best Fit, se alege, dintre toate zonele de memorie disponibile a căror dimensiune permite alocarea job-ului:
- a). orice zonă;
 - b). zona de la prima adresă;
 - c). prima zonă de memorie cu dimensiune suficient de mare ca job-ul să încapă;
 - d). zona cu cea mai mare dimensiune;
 - e). zona cu cea mai mică dimensiune.
45. În cadrul strategiei de plasare în memorie, Worst Fit, lista zonelor de memorie liberă este:
- a). ordonată crescător după dimensiunile golurilor;
 - b). ordonată crescător în ordinea adreselor;
 - c). ordonată descrescător după dimensiunile golurilor;
 - d). ordonată crescător după dimensiunile golurilor și în cadrul acesteia descrescător după adrese;
 - e). ordonată descrescător în ordinea adreselor.
46. Permisuni de execuție în cazul unui director:
- a). permite modificarea, ștergerea și redenumirea unui fișier;
 - b). permisuni permite actualizarea, ștergerea și redenumirea directorului;
 - c). permite vizualizarea (citirea) conținutului unui fișier;
 - d). permite accesul (comutarea) în directorul respectiv, și efectuarea de operații asupra fișierelor din cadrul său;
 - e). permite executarea fișierului prin tastarea numelui acestuia.
47. Grupul ce conține toți utilizatorii neprivilegiați ai sistemului este:
- a). `starup`;
 - b). `others`;
 - c). `owners`;
 - d). `users`;
 - e). `everyone`.
48. UID -ul este folosit de către sistemul de operare UNIX:
- a). pentru adăugarea oricărei informații pe care acesta o consideră necesară în identificarea utilizatorului;
 - b). pentru a identifica anumite informații ce sunt asociate cu utilizatorul;
 - c). pentru a păstra urma grupului la care utilizatorii aparțin când ei se conectează la sistem (grup de `startup`);
 - d). pentru comanda ce va fi executată când procedura de conectare se termină;
 - e). pentru a specifica directorul în care va fi comutat utilizatorul după ce acesta s-a conectat la sistem (directorul curent).
49. Dacă se dorește ca un cont să nu poată fi folosit pentru accesarea sistemului, se plasează în câmpul de parolă corespunzător lui, din cadrul fișierului `/etc/passwd`:
- a). două puncte;
 - b). un punct;
 - c). caracterul `~` (tilda);
 - d). un spațiu;
 - e). un asterisc.
50. Câmpul `director_implicit` din cadrul unei linii existente în fișierul `/etc/passwd` semnifică:
- a). un număr unic care identifică directorul grupului la care aparține utilizatorul;

- b). directorul ce conține directorul implicit al utilizatorului;
- c). directorul în care utilizatorii sunt plasați când se conectează la sistem (director home);
- d). un director creat de utilizator;
- e). un număr unic care identifică utilizatorul pentru sistemul de operare.

51. Pentru oprirea, într-o manieră ordonată, a calculatorului pe care rulează sistemul UNIX, se utilizează comanda:

- a). shutdown;
- b). combinația de taste <Ctrl><D>;
- c). comanda exit;
- d). comanda bye;
- e). combinația de taste <Ctrl><C>.

52. Accesul la resursele unui sistem UNIX se realizează, de regulă:

- a). prin intermediul fișierelor de dispozitiv;
- b). prin intermediul legăturilor simbolice la fișiere;
- c). prin intermediul unor conturi utilizator ce sunt setate de către administratorul de sistem, ulterior instalării sistemului de operare;
- d). prin intermediul legăturilor hard la fișiere;
- e). prin intermediul unor fișiere de comenzi.

53. Câmpul UID din cadrul unei linii a fișierului /etc/passwd specifică:

- a). un număr unic care identifică utilizatorul pentru sistemul de operare;
- b). un identificator unic pentru utilizator;
- c). un număr unic care identifică grupul la care aparține utilizatorul;
- d). uzual, numele real al utilizatorului, sau o altă informație referitoare la acesta;
- e). directorul în care utilizatorii sunt plasați când se conectează la sistem.

54. Care dintre următoarele conturi sunt setate de către administratorul de sistem, ulterior instalării sistemului de operare:

- a). conturile utilizator;
- b). conturile sistem;
- c). contul utilizatorului root;
- d). conturile standard;
- e). conturile grupurilor standard;

55. Toate informațiile referitoare la conturile utilizatorilor sunt păstrate în fișierul:

- a). /etc/group;
- b). /dev;
- c). /etc;
- d). /etc/passwd;
- e). /usr.

56. În situația în care câmpul parola din cadrul fișierului /etc/passwd nu conține nimic (este gol), la contul respectiv:

- a). nu se permite accesul;
- b). se permite accesul utilizatorului implicit;
- c). se permite accesul doar membrilor grupului din care face parte utilizatorul implicit;
- d). se permite accesul doar utilizatorilor sistem;
- e). se permite un acces nerestricțiv.

57. Schimbarea proprietarului unui fișier sau director se face cu comanda:

- a). `chgrp`;
- b). `chown`;
- c). `chmod`;
- d). `cat`;
- e). `cp`.

58. Permisele de scriere în cazul unui director:

- a). permite actualizarea, ștergerea și redenumirea directorului;
- b). permite accesul (comutarea) în directorul respectiv, și efectuarea de operații asupra fișierelor din cadrul său;
- c). permite afișarea conținutului acestuia;
- d). permite executarea directorului;
- e). permite schimbarea proprietarului directorului.

59. Comanda `chmod o+rw test.txt` modifică permisiunile de acces pentru:

- a). grupul căruia îi aparține utilizatorul;
- b). utilizatorul care este proprietar al fișierului;
- c). grupul standard căruia îi aparține utilizatorul;
- d). alții (alții decât utilizatorul și grupul acestuia);
- e). grupul implicit căruia îi aparține utilizatorul.

60. Pentru ștergerea tuturor permisiunilor asupra fișierului `test.txt` pentru oricine, se va tasta comanda:

- a). `chmod u+rw test.txt`;
- b). `chmod g+w test.txt`;
- c). `chmod u-rw test.txt`;
- d). `chmod g-rw test.txt`;
- e). `chmod a-rw test.txt`;

61. Comanda `chmod 751` stabilește următoarele permisiuni:

- a). pentru proprietar `rw`, pentru grup `rx`, pentru alții `r`;
- b). pentru proprietar `rwX`, pentru grup `rx`, pentru alții `x`;
- c). pentru proprietar `rwX`, pentru grup `wX`, pentru alții `w`;
- d). pentru proprietar `rwX`, pentru grup `r`, pentru alții `wX`;
- e). pentru proprietar `rx`, pentru grup `rx`, pentru alții `wX`.

62. Un sistem de fișiere reprezintă:

- a). modul de organizare și exploatare a informațiilor stocate pe un suport de memorie externă în vederea accesării și prelucrării lor de către sistemul de operare;
- b). o entitate care încapsulează informația de un anumit tip;
- c). un program executabil situat, de regulă, în directorul `/bin`, și care este tratat de către nucleul (kernel) sistemului de operare, ca orice proces utilizator neprivilegiat;
- d). un sistem multiuser și multisesiune, utilizatorii putând avea deschise mai multe sesiuni de lucru pe un același calculator sau pe calculatoare diferite din rețea;
- e). un fișier permanent, ce este privit de către sistemul de operare ca un șir de octeți fără o organizare logică specială, și a cărei structură internă este irelevantă din punctul de vedere al administratorului de sistem.

63. `/dev/hdb3` specifică:

- a). cea de a doua partiție de pe primul hard disc atașat unei unități de control SCSI;
- b). întregul disc master conectat pe prima unitatea de control SCSI;
- c). cea de a treia partiție a unității de hard disc slave de pe prima unitatea de control;
- d). cea de a treia partiție a unității de hard disc master de pe a doua unitate de control;
- e). cea de a treia partiție a unității de hard disc master atașat unei unități de control SCSI.

64. O legătură hard:

- a). este un fișier distinct, marcat cu un cod special, care are ca și conținut numele complet al fișierului indicat;
- b). este un mecanism folosit de UNIX pentru a comunica între două calculatoare gazdă, utilizând pentru aceasta porturi rețea;
- c). este un fișier temporar ce permite comunicația între procese prin mecanismul pipe (conductă);
- d). creează o referință (un pointer) către un fișier deja existent, fără duplicarea conținutului fișierului respectiv;
- e). este un fișier de dispozitiv, care va conține întotdeauna informații despre driver-ul (programul de comandă) al aceluși periferic.

65. Pentru a genera legături hard sau simbolice la fișiere, se utilizează o comandă specifică, numită:

- a). ln;
- b). mknod;
- c). ls;
- d). lpt;
- e). link.

66. Fișierele de dispozitiv rezidă în directorul:

- a). /etc;
- b). /bin;
- c). /dev;
- d). /usr;
- e). /home.

67. O comandă `ls -l` va afișa în cazul unui fișier normal, o informație similară cu următoarea:

- a). drw----- 1 student ise 42 May 12 13:09 fistest
- b). -rw----- 1 student ise 42 May 12 13:09 fistest
- c). crw----- 1 student ise 42 May 12 13:09 fistest
- d). brw----- 1 student ise 42 May 12 13:09 fistest
- e). srw----- 1 student ise 42 May 12 13:09 fistest

68. Un shell este:

- a). un mecanism folosit de UNIX pentru a comunica între două calculatoare gazdă, utilizând pentru aceasta porturi rețea;
- b). un fișier permanent, ce este privit de către sistemul de operare ca un șir de octeți fără o organizare logică specială, și a cărui structură internă este irelevantă din punctul de vedere al administratorului de sistem;
- c). o entitate care încapsulează informația de un anumit tip;
- d). un fișier distinct, marcat cu un cod special, care are ca și conținut numele complet al fișierului indicat;
- e). un program executabil situat, de regulă, în directorul /bin, și care este tratat de către nucleul (kernel) sistemului de operare, ca orice proces utilizator neprivilegiat.

69. Comenzile UNIX sunt

- a). fișiere speciale asociate dispozitivelor periferice;
 - b). legături simbolice;
 - c). biblioteci ale limbajelor de programare instalate;
 - d). programe executabile care pot fi găsite în directoarele `/bin`, `/usr/bin`;
 - e). conturi sistem ce se lansează automat în execuție.
70. Ce conține un fișier special din cadrul sistemului Unix?
- a). informații despre sistemul de fișiere asociat perifericului;
 - b). legături simbolice;
 - c). informații despre driver-ul perifericului asociat fișierului;
 - d). informații despre i-noduri;
 - e). informații destinate administrării sistemului de fișiere.
71. Ce conține de regulă directorul `/dev` din cadrul sistemului de operare Unix?:
- a). comenzi mai puțin utilizate;
 - b). fișiere executabile și utilitare folosite în lucrul cu sistemul de operare;
 - c). biblioteci ale limbajelor de programare instalate;
 - d). date temporare;
 - e). fișiere speciale asociate dispozitivelor periferice.
72. O conductă cu nume este:
- a). un fișier distinct, marcat cu un cod special, care are ca și conținut numele complet al fișierului indicat;
 - b). un mecanism folosit de UNIX pentru a comunica între două calculatoare gazdă, utilizând pentru aceasta porturi rețea;
 - c). o entitate care încapsulează informația de un anumit tip;
 - d). un fișier temporar ce permite comunicația între procese prin mecanismul pipe;
 - e). este un fișier de dispozitiv, care va conține întotdeauna informații despre driver-ul (programul de comandă) al aceluși periferic.
73. Directoarele sunt:
- a). legături ce creează o referință (un pointer) către un fișier deja existent, fără duplicarea conținutului fișierului respectiv;
 - b). fișiere distincte, marcat cu un cod special, care au ca și conținut numele complet al fișierului indicat;
 - c). fișiere permanente, ce sunt privite de către sistemul de operare ca un șir de octeți fără o organizare logică specială, și a căror structură internă este irelevantă din punctul de vedere al administratorului de sistem;
 - d). un mod de organizare și exploatare a informațiilor stocate pe un suport de memorie externă în vederea accesării și prelucrării lor de către sistemul de operare;
 - e). fișiere de un tip special ce conțin o listă de alte fișiere.
74. Protecția contului utilizatorului `root`, dar și a celorlalte conturi utilizator, se realizează prin:
- a). permisiunile de acces;
 - b). asignarea de parole;
 - c). fișiere ascunse;
 - d). duplicarea conținutului fișierelor la care au acces;
 - e). alegerea shell-ului potrivit.
75. În cazul în care un punct este folosit ca prim caracter în cadrul numelui unui fișier, acesta conferă fișierului statutul de:
- a). fișier de dispozitiv;
 - b). legătură simbolică;

- c). fișier ascuns (hidden);
- d). specificator pentru nume de fișier global;
- e). conductă cu nume (pipe).

76. Pentru ca un utilizator să poată copia un fișier, el trebuie să aibă permisiunea de:

- a). de a citi acel fișier;
- b). de a scrie acel fișier;
- c). de a executa acel fișier;
- d). de a executa și de a scrie acel fișier;
- e). nu este nevoie să aibă permisiuni pe acel fișier.

77. Care este efectul comenzii `chmod a-x aplicatii`:

- a). adaugă permisiunea de scriere pentru `others`;
- b). șterge permisiunea de scriere pentru grup;
- c). șterge permisiunea de citire pentru `owner`;
- d). adaugă permisiunea de execuție pentru grup;
- e). șterge permisiunea de execuție pentru toți utilizatorii.

78. Comanda `chmod 536` stabilește următoarele permisiuni:

- a). pentru proprietar `rw`, pentru grup `rx`, pentru alții `r`;
- b). pentru proprietar `rx`, pentru grup `wx`, pentru alții `rw`;
- c). pentru proprietar `rw`, pentru grup `wx`, pentru alții `w`;
- d). pentru proprietar `rw`, pentru grup `r`, pentru alții `wx`;
- e). pentru proprietar `rx`, pentru grup `rx`, pentru alții `wx`.

79. `/dev/sda2` specifică:

- a). prima partiție de pe primul hard disc atașat unei unități de control SCSI;
- b). a doua partiție de pe al doilea hard disc atașat unei unități de control SCSI;
- c). a doua partiție de pe primul hard disc atașat unei unități de control SCSI;
- d). a doua partiție de pe primul hard disc atașat unei unități de control IDE;
- e). prima partiție de pe primul hard disc atașat unei unități de control SCSI.

80. Dezavantajul unui legăturii simbolice este că:

- a). pentru ea (fiind fișier) trebuie creat un `i-nod` separat și, în plus, ocupă spațiu pe disc prin conținutul ei;
- b). o astfel de legătură nu poate exista peste mai multe sisteme de fișiere deoarece ea partajează un `i-nod`;
- c). un director nu poate avea o astfel de legătură;
- d). poate să existe peste mai multe sisteme de fișiere;
- e). se utilizează întotdeauna împreună cu o legătură hard.

Răspunsuri

1 d; 2 b; 3 b; 4 d; 5 c; 6 c; 7 d; 8 a; 9 b; 10 e; 11 e; 12 b; 13 b; 14 c; 15 b; 16 d; 17 c; 18 a; 19 d; 20 b; 21 b; 22 b; 23 c; 24 b; 25 c; 26 e; 27 e; 28 e; 29 c; 30 b; 31 d; 32 b; 33 b; 34 e; 35 d; 36 c; 37 d; 38 e; 39 a; 40 b; 41 e; 42 b; 43 d; 44 e; 45 c; 46 d; 47 d; 48 b; 49 e; 50 c; 51 a; 52 c; 53 a; 54 a; 55 d; 56 e; 57 b; 58 a; 59 d; 60 e; 61 b; 62 a; 63 c; 64 d; 65 a; 66 c; 67 b; 68 e; 69 d; 70 c; 71 e; 72 d; 73 e; 74 b; 75 c; 76 a; 77 e; 78 b; 79 c; 80 a.

GRILE PSI

1. **Cea mai generală definiție a unui sistem informatic îl prezintă ca un:**
 - a. Sistem de colectare, memorare, prelucrare și distribuire a informațiilor care utilizează calculatorul electronic;
 - b. Sistem informațional de tip financiar-contabil, care prelucrează informații economice;
 - c. Sistem integrat ce folosește baze de date distribuite cu teleprelucrare;
 - d. Sistem de colectare și memorare a informațiilor din calculatorul electronic;
 - e. Sistem de prelucrare a informațiilor care utilizează calculatorul electronic.
2. **Definiția sistemului informatic furnizată de Gh. Sabău și I. Lungu prevede că este:**
 - a. Asociat unui sistem de teleprelucrare distribuită sau cu prelucrare pe loturi;
 - b. Un ansamblu de elemente interconectate funcțional în scopul automatizării obținerii informației și fundamentării deciziilor;
 - c. Un sistem care colectează și prelucrează informația fără ajutorul omului;
 - d. Un sistem informațional care se ocupă cu prelucrarea informației la distanță;
 - e. Un sistem informațional care pune la dispoziția decidenților informația necesară prelucrării.
3. **După unele păreri sistemul informatic poate fi avut în vedere ca model extern, dar și ca model intern. Ce reprezintă modelul extern?**
 - a. Se referă la structura fizică orientată către echipamente, unde interesează datele, procedurile de prelucrare, echipamentele, mediile de programare și sistemele de operare;
 - b. Fenomenele și procesele economice, inclusiv modificările produse în activitatea agentului economic;
 - c. Apare în momentul obținerii unui rezultat, fiind destinat actualizării bazelor de date;
 - d. Se referă la o arhitectură funcțională orientată către utilizator, pe care îl interesează informațiile și procesele lor de prelucrare;
 - e. Rezultatul prelucrării colecțiilor de date.
4. **După unele păreri sistemul informatic poate fi avut în vedere ca model extern, dar și ca model intern. Ce reprezintă modelul intern?**
 - a. Se referă la o arhitectură funcțională orientată către utilizator, pe care îl interesează informațiile și procesele lor de prelucrare;
 - b. Fenomenele și procesele economice, inclusiv modificările produse în activitatea agentului economic;
 - c. Apare în momentul obținerii unui rezultat, fiind destinat actualizării bazelor de date;
 - d. Rezultatul prelucrării colecțiilor de date;
 - e. Se referă la structura fizică orientată către echipamente, unde interesează datele, procedurile de prelucrare, echipamentele, mediile de programare și sistemele de operare.
5. **Clasificați sistemele informatice în concordanță cu nivelurile organizatorice?**
 - a. Sisteme informatice departamentale, organizaționale și interorganizaționale;
 - b. Sisteme informatice financiar-contabile, pentru producție, pentru activitatea comercială și pentru management;
 - c. Sisteme informatice tranzacționale, manageriale, suport pentru grup și suport pentru decizii;
 - d. Sisteme informatice bazate pe mainframe-uri, bazate pe PC-uri și distribuite;
 - e. Sisteme operaționale, tactice (manageriale) și strategice.
6. **Clasificați sistemele informatice în concordanță cu arile funcționale?**
 - a. Sisteme informatice departamentale, organizaționale și interorganizaționale;
 - b. Sisteme informatice financiar-contabile, pentru producție, pentru activitatea comercială, pentru gestiunea resurselor umane și pentru management;
 - c. Sisteme informatice tranzacționale, manageriale, suport pentru grup și suport pentru decizii;
 - d. Sisteme informatice bazate pe mainframe-uri, bazate pe PC-uri și distribuite;
 - e. Sisteme operaționale, tactice (manageriale) și strategice.
7. **Clasificați sistemele informatice în concordanță cu suportul furnizat de sistem?**
 - a. Sisteme informatice departamentale, organizaționale și interorganizaționale;

- b. Sisteme informatice financiar-contabile, pentru producție, pentru activitatea comercială și pentru management;
 - c. Sisteme informatice tranzacționale, manageriale, automatizate pentru birouri, suport pentru grup, suport pentru decizii, sisteme executive sau sisteme suport și sisteme suport inteligente;
 - d. Sisteme informatice bazate pe mainframe-uri, bazate pe PC-uri și distribuite;
 - e. Sisteme operaționale, tactice (manageriale) și strategice.
- 8. Clasificați sistemele informatice în concordanță cu arhitectura sistemelor informatice?**
- a. Sisteme informatice departamentale, organizaționale și interorganizaționale;
 - b. Sisteme informatice financiar-contabile, pentru producție, pentru activitatea comercială și pentru management;
 - c. Sisteme informatice tranzacționale, manageriale, suport pentru grup și suport pentru decizii;
 - d. Sisteme informatice bazate pe mainframe-uri, bazate pe PC-uri și distribuite;
 - e. Sisteme operaționale, tactice (manageriale) și strategice.
- 9. Clasificați sistemele informatice după natura activităților pentru care sunt construite?**
- a. Sisteme informatice departamentale, organizaționale și interorganizaționale;
 - b. Sisteme informatice financiar-contabile, pentru producție, pentru activitatea comercială și pentru management;
 - c. Sisteme informatice tranzacționale, manageriale, suport pentru grup și suport pentru decizii;
 - d. Sisteme informatice bazate pe mainframe-uri, bazate pe PC-uri și distribuite;
 - e. Sisteme operaționale, tactice (manageriale) și strategice.
- 10. Sistemele tactice, numite și sisteme manageriale sunt:**
- a. Sistemele care oferă suportul necesar operațiilor zilnice ale unei organizații cum ar fi înregistrarea unei facturi sau a numărului de ore pe care le lucrează un muncitor;
 - b. Sistemele legate de activitățile efectuate de către managerii de la nivelele medii ale întreprinderii, activități precum planificarea organizarea și controlul pe termen scurt și mediu;
 - c. Sistemele legate de situațiile pe termen lung și de deciziile care schimbă semnificativ maniera în care vor fi realizate obiectivele firmei;
 - d. Sistemele informatice necesare activităților și deciziilor operaționale;
 - e. Sistemele care implică doar planificarea de lungă durată.
- 11. Care sunt etapele realizării unui sistem informatic?**
- a. Analiza sistemului informațional existent, proiectarea sistemului informatic, elaborarea și testarea programelor, implementarea sistemului informatic și exploatarea curentă;
 - b. SSADM, Merise, OMT și Prototipizarea;
 - c. Etapa conceptuală, logică și fizică;
 - d. Identificarea operațiilor, stabilirea instanțelor și implementarea prelucrărilor;
 - e. Etape adaptive, ameliorative și inovatoare.
- 12. Care dintre răspunsurile de mai jos nu face parte dintre modelele de realizare a sistemelor informatice?**
- a. Modelul cu extensii;
 - b. Modelul strategic;
 - c. Modelul liniar;
 - d. Modelul cu prototip;
 - e. Modelul ierarhic.
- 13. Care sunt ciclurile metodologiei Merise?**
- a. Ciclul cu prototip, ierarhic și conceptual;
 - b. Ciclul organizatoric, tehnic și funcțional;
 - c. Ciclul de abstractizare, de decizie și de viață al sistemului informatic;
 - d. Ciclul conceptual, logic și fizic;
 - e. Ciclul strategic, liniar și ierarhic.
- 14. Ciclul de abstractizare se dezvoltă pe trei niveluri. Care sunt acestea?**
- a. Nivelul cu prototip, ierarhic și conceptual;
 - b. Nivelul organizatoric, tehnic și funcțional;

- c. Nivelul de abstractizare, de decizie și de viață al sistemului informatic;
 - d. Nivelul conceptual, logic și fizic;
 - e. Nivelul strategic, liniar și ierarhic.
- 15. Ciclul de decizie se compune din mulțimea deciziilor ce se iau pe durata ciclului de viață al sistemului informatic, pe ce tipuri de decizii sunt structurate acestea?**
- a. Decizii conceptuale, logice și fizice;
 - b. Decizii analitice și sintetice;
 - c. Decizii externe și interne;
 - d. Decizii strategice, tactice și operaționale;
 - e. Decizii generale, organizatorice, tehnice și funcționale.
- 16. Care sunt categoriile de strategii de realizare a unui sistem informatic, în funcție de rolul sistemelor informatice în cadrul agentului economic?**
- a. Strategii conceptuale, logice și fizice;
 - b. Strategii ameliorative, inovatoare și adaptive;
 - c. Strategii externe și interne;
 - d. Strategii tactice și operaționale;
 - e. Strategii generale, organizatorice, tehnice și funcționale.
- 17. Ce urmăresc strategiile ameliorative de realizare a unui sistem informatic?**
- a. Automatizarea activităților și operațiilor de rutină sau cu caracter repetitiv, fără să antreneze mari modificări în sistemul agentului economic;
 - b. Urmăresc schimbări importante în organizarea și funcționarea agentului economic, prin valorificarea superioară a calculatorului economic;
 - c. Sunt concepute și realizate astfel încât să fie compatibile cu schimbările care apar în cerințele informaționale și organizatorice și în funcționarea agentului economic;
 - d. Descompunerea unui sistem complex pe niveluri ierarhice, succesiv, până la module elementare, simple și relativ independente care sunt controlate de module coordonatoare;
 - e. Constă în agregarea modulelor de jos în sus punând în evidență legăturile dintre ele până se ajunge la un singur modul.
- 18. Ce urmăresc strategiile inovatoare de realizare a unui sistem informatic?**
- a. Automatizarea activităților și operațiilor de rutină sau cu caracter repetitiv, fără să antreneze mari modificări în sistemul agentului economic;
 - b. Urmăresc schimbări importante în organizarea și funcționarea agentului economic, prin valorificarea superioară a calculatorului economic;
 - c. Sunt concepute și realizate astfel încât să fie compatibile cu schimbările care apar în cerințele informaționale și organizatorice și în funcționarea agentului economic;
 - d. Descompunerea unui sistem complex pe niveluri ierarhice, succesiv, până la module elementare, simple și relativ independente care sunt controlate de module coordonatoare;
 - e. Constă în agregarea modulelor de jos în sus punând în evidență legăturile dintre ele până se ajunge la un singur modul.
- 19. Ce urmăresc strategiile adaptive de realizare a unui sistem informatic?**
- a. Automatizarea activităților și operațiilor de rutină sau cu caracter repetitiv, fără să antreneze mari modificări în sistemul agentului economic;
 - b. Urmăresc schimbări importante în organizarea și funcționarea agentului economic, prin valorificarea superioară a calculatorului economic;
 - c. Sunt concepute și realizate astfel încât să fie compatibile cu schimbările care apar în cerințele informaționale și organizatorice și în funcționarea agentului economic;
 - d. Descompunerea unui sistem complex pe niveluri ierarhice, succesiv, până la module elementare, simple și relativ independente care sunt controlate de module coordonatoare;
 - e. Constă în agregarea modulelor de jos în sus punând în evidență legăturile dintre ele până se ajunge la un singur modul.
- 20. Care sunt categoriile de strategii de realizare a unui sistem informatic, în funcție de modul de abordare a subsistemelor și aplicațiilor ce-l compun?**

- a. Strategii conceptuale, logice și fizice;
 - b. Strategii descendente, ascendente și mixte;
 - c. Strategii ameliorative, inovatoare și adaptive;
 - d. Strategii tactice și operaționale;
 - e. Strategii generale, organizatorice, tehnice și funcționale.
- 21. Ce urmăresc strategiile descendente de realizare a unui sistem informatic?**
- a. Automatizarea activităților și operațiilor de rutină sau cu caracter repetitiv, fără să antreneze mari modificări în sistemul agentului economic;
 - b. Urmăresc schimbări importante în organizarea și funcționarea agentului economic, prin valorificarea superioară a calculatorului economic;
 - c. Sunt concepute și realizate astfel încât să fie compatibile cu schimbările care apar în cerințele informaționale și organizatorice și în funcționarea agentului economic;
 - d. Descompunerea unui sistem complex pe niveluri ierarhice, succesiv, până la module elementare, simple și relativ independente care sunt controlate de module coordonatoare;
 - e. Constă în agregarea modulelor de jos în sus punând în evidență legăturile dintre ele până se ajunge la un singur modul.
- 22. Ce urmăresc strategiile ascendente de realizare a unui sistem informatic?**
- a. Automatizarea activităților și operațiilor de rutină sau cu caracter repetitiv, fără să antreneze mari modificări în sistemul agentului economic;
 - b. Urmăresc schimbări importante în organizarea și funcționarea agentului economic, prin valorificarea superioară a calculatorului economic;
 - c. Sunt concepute și realizate astfel încât să fie compatibile cu schimbările care apar în cerințele informaționale și organizatorice și în funcționarea agentului economic;
 - d. Descompunerea unui sistem complex pe niveluri ierarhice, succesiv, până la module elementare, simple și relativ independente care sunt controlate de module coordonatoare;
 - e. Constă în agregarea modulelor de jos în sus punând în evidență legăturile dintre ele până se ajunge la un singur modul.
- 23. Ce reprezintă analiza sistemelor informaționale economice?**
- a. Este activitatea prin care se realizează cunoașterea sistemului obiect și a cerințelor de informații din sistemul de decizie;
 - b. Este reprezentată de nucleul sistemului informatic, concretizată în programele de actualizare a bazelor de date;
 - c. Este concretizată în tranzacțiile interne și externe ale sistemului informatic;
 - d. Este elaborată concepția logică a sistemului informatic, definirea lui din punct de vedere structural și funcțional;
 - e. Stabilește componentele viitorului sistemului informatic, a ieșirilor, a bazei informaționale de intrare.
- 24. În cadrul analizei sistemului informațional existent există conceptul orientat pe organism, care stă la baza stabilirii elementelor sistemului obiect. Ce determină acest concept?**
- a. Determină structura funcțională, caz în care analiza sistemului informațional existent se va face pe funcții și pe subfuncții ale agentului economic analizat;
 - b. Determină structura organizatorică, caz în care analiza sistemului informațional existent se va face pe compartimente funcționale;
 - c. Determină obiectivelor și a deciziilor curente și potențiale, precum și a atributelor conducerii, corespunzătoare obiectivelor și deciziilor;
 - d. Determină identificarea sau construirea unui model, a unei proceduri de elaborare a fiecărei decizii sau a unui model al procesului de luare a deciziilor;
 - e. Determină testarea sensibilității modelului la acuratețea și disponibilitatea datelor de intrare, specificarea limitelor și disponibilităților datelor cerute de model.
- 25. În cadrul analizei sistemului informațional existent există conceptul orientat pe activitate, care stă la baza stabilirii elementelor sistemului obiect. Ce determină acest concept?**

- a. Determină structura funcțională, caz în care analiza sistemului informațional existent se va face pe funcții și pe subfuncții ale agentului economic analizat;
 - b. Determină structura organizatorică, caz în care analiza sistemului informațional existent se va face pe compartimente funcționale;
 - c. Determină obiectivelor și a deciziilor curente și potențiale, precum și a atributelor conducerii, corespunzătoare obiectivelor și deciziilor;
 - d. Determină identificarea sau construirea unui model, a unei proceduri de elaborare a fiecărei decizii sau a unui model al procesului de luare a deciziilor;
 - e. Determină testarea sensibilității modelului la acuratețea și disponibilitatea datelor de intrare, specificarea limitelor și disponibilităților datelor cerute de model.
- 26. În vederea delimitării corecte a cerințelor informaționale se poate distinge varianta de abordare a sistemului obiect pornind de la analiza deciziilor. Ce reprezintă aceasta?**
- a. Proiectarea unui model al sistemului existent, adică a unui sistem sinonim cu ajutorul căruia să se simuleze funcționarea sistemului real;
 - b. Caută să obțină cerințele informaționale prin analiza datelor folosite curent în sistemul existent, sau a datelor potențial folosibile;
 - c. Determină cerințele de informații pornind de la analiza obiectivelor și a deciziilor ce trebuie luate;
 - d. Sistemul se dezvoltă treptat, în concordanță cu cerințele reale ale utilizatorilor;
 - e. Definește modelul de ansamblu a sistemului informatic pe baza unei analize complexe cu personal numeros.
- 27. În vederea delimitării corecte a cerințelor informaționale se poate distinge varianta de abordare a sistemului obiect pornind de la analiza datelor. Ce reprezintă aceasta?**
- a. Are ca cerință de aplicare modularizarea sistemului, deci obiectivul principal este realizarea modularizării de sus în jos;
 - b. Caută să obțină cerințele informaționale prin analiza datelor folosite curent în sistemul existent, sau a datelor potențial folosibile;
 - c. Determină cerințele de informații pornind de la analiza obiectivelor și a deciziilor ce trebuie luate;
 - d. Baza informațională se transpune în colecții de date, proceduri de creare și actualizare care constituie nucleul sistemului informatic;
 - e. Realizează succesiv modelul conceptual al datelor și modelul conceptual al prelucrărilor.
- 28. Care este avantajul variantei de abordare a sistemului obiect pornind de la analiza deciziilor?**
- a. Simplificare a structurilor decizionale;
 - b. Caută să obțină cerințele informaționale prin analiza datelor folosite curent în sistemul existent, sau a datelor potențial folosibile;
 - c. Determină cerințele de informații pornind de la analiza obiectivelor și a deciziilor ce trebuie luate;
 - d. În urma acestei abordări rezultă un volum mic de date;
 - e. Conduce la un volum mai mare de date, dar are avantajul că deciziile care se vor lua pe baza acestor date vor fi mai puțin afectate de modificările ce pot surveni pe parcurs.
- 29. Care este avantajul variantei de abordare a sistemului obiect pornind de la analiza datelor?**
- a. Simplificare a structurilor funcționale;
 - b. Caută să obțină cerințele informaționale prin analiza datelor folosite curent în sistemul existent, sau a datelor potențial folosibile;
 - c. Determină cerințele de informații pornind de la analiza obiectivelor și a deciziilor ce trebuie luate;
 - d. În urma acestei abordări rezultă un volum mic de date;
 - e. Conduce la un volum mai mare de date, dar are avantajul că deciziile care se vor lua pe baza acestor date vor fi mai puțin afectate de modificările ce pot surveni pe parcurs.
- 30. Care sunt cele 3 faze în care se desfășoară analiza sistemului informațional existent?**
- a. Faze organizatorice, departamentale și funcționale;

- b. Organizarea și conducerea analizei sistemului informațional, realizarea analizei și finalizarea analizei sistemului informațional;
 - c. Analiza, proiectarea și execuția;
 - d. Delimitarea ariei de cuprindere, evidențierea dotării cu tehnică de calcul și fundamentarea soluției de principiu;
 - e. Cercetarea documentelor, cercetarea fișierelor folosite și realizarea situațiilor finale.
- 31. Care este avantajul utilizării în cadrul colectivului analizei sistemului informațional existent al personalului propriu al unității beneficiare?**
- a. Prezintă avantajul redundanței informaționale din noul sistem;
 - b. Din colectiv fac parte doar proiectanții noului sistem informatic;
 - c. Prezintă avantajul cunoașterii sistemului informațional de către colectivul de analiză, ceea ce asigură efectuarea unei analize de calitate într-un termen relativ scurt;
 - d. Nu se folosește din personalul propriu al firmei beneficiare în cadrul colectivului analizei sistemului informațional existent;
 - e. Doar analiștii de sistem fac parte din cadrul colectivului de analiză.
- 32. Care sunt cele două categorii de tehnici de analiză a sistemului informațional economic?**
- a. Tehnici de evaluare și de realizare;
 - b. Tehnici de analiză și de proiectare;
 - c. Tehnici organizaționale și funcționale;
 - d. Tehnici elementare și complexe;
 - e. Tehnici normale și extraordinare.
- 33. Cu ce se ocupă tehnicile elementare de analiză a sistemelor informaționale economice?**
- a. Realizează analiza și proiectarea sistemului informatic;
 - b. Asigură consistența și neredundanța datelor de la sursele de unde provin acestea;
 - c. Permite descrierea și sinteza operațiilor de început ale sistemului informatic;
 - d. Impune metode exacte de respectare a standardelor de lucru;
 - e. Realizează operația de culegere a datelor și asigură o sistematizare redusă a acestora.
- 34. Din ce este compusă grupa tehnicilor elementare de analiză?**
- a. Observarea directă, studierea documentației existente, participarea analistului la lucrări interviul și chestionarul;
 - b. Analiza, proiectarea și execuția;
 - c. Tehnici ameliorative, inovatoare și adaptive;
 - d. Tehnici analitice și sintetice;
 - e. Tehnici tranzacționale, manageriale, strategice și tactice.
- 35. Cu ce se ocupă tehnicile complexe de analiză a sistemelor informaționale economice?**
- a. Se ocupă cu toate activitățile din analiza sistemului informațional existent;
 - b. Realizează toate activitățile de analiză, respectiv culegerea și sistematizarea datelor, precum și evaluarea sistemului informațional;
 - c. Se ocupă cu reinventarea, generarea și memorarea directă a informațiilor;
 - d. Descrie contractul încheiat între furnizorul de software și beneficiarul acestuia;
 - e. Particularizează fiecare aspect al muncii viitoare din cadrul sistemului informatic.
- 36. Sistematizarea datelor, constând nu doar într-o simplă ordonare a lor, ci și într-o sintetizare, se realizează folosind una sau mai multe tehnici de reprezentare. Dați câteva tehnici de reprezentare?**
- a. Tehnici ierarhice, cu extensii și mixte;
 - b. Tehnica Merise, OMT și SSADM;
 - c. Scrierea tehnică, grilele, listele de acțiuni condiționate, tabelele de decizii, etc;
 - d. Tranzacțiile interne și tranzacțiile externe;
 - e. Tehnicile ameliorative, inovatoare și adaptive.
- 37. Care sunt tehnicile complexe de analiză a sistemelor informaționale?**
- a. Metodele diferențiale, realizarea intrinsecă și dimensionarea informațională;
 - b. Prototipizarea și sinteza și agregarea;

- c. Tehnicile ameliorative, inovatoare și adaptive;
 - d. Analiza diagnostic, analiza celulară și analiza S.O.P. (Study Organization Plan);
 - e. Tranzacțiile interne și tranzacțiile externe.
- 38. Ce urmărește să determine evaluarea critică a obiectivelor sistemului informațional?**
- a. Determinarea gradului de utilizare a documentelor tipizate pentru eliminarea din sistem a documentelor netipizate;
 - b. Evidențierea fluxurilor paralele de date și a circuitelor neraționale care trebuie eliminate din noul sistem;
 - c. Descrierea fluxurilor informaționale printr-o organigramă unde se precizează: activitățile, compartimentele sau posturile de lucru;
 - d. Gradul de pregătire a unității analizate pentru proiectarea unui sistem informatic ce are în vedere pregătirea personalului;
 - e. Modul în care sunt rezolvate cerințele informaționale din sistem, măsura în care informațiile solicitate lipsesc sau sunt necorespunzătoare.
- 39. Ce urmărește să determine evaluarea critică a mijloacelor tehnice utilizate în sistemul informațional?**
- a. Trebuie să pună în evidență gradul de dotare cu tehnică de calcul, ponderea acesteia în sistemul informațional, gradul ei de utilizare și oportunitatea folosirii tehnicii existente în noul sistem;
 - b. Evidențierea fluxurilor paralele de date și a circuitelor neraționale care trebuie eliminate din noul sistem;
 - c. Descrierea fluxurilor informaționale printr-o organigramă unde se precizează: activitățile, compartimentele sau posturile de lucru;
 - d. Gradul de pregătire a unității analizate pentru proiectarea unui sistem informatic ce are în vedere pregătirea personalului;
 - e. Modul în care sunt rezolvate cerințele informaționale din sistem, măsura în care informațiile solicitate lipsesc sau sunt necorespunzătoare.
- 40. Ce urmărește să determine evaluarea critică a personalului în sistemul informațional?**
- a. Trebuie să pună în evidență gradul de dotare cu tehnică de calcul, ponderea acesteia în sistemul informațional, gradul ei de utilizare și oportunitatea folosirii tehnicii existente în noul sistem;
 - b. Are rolul de a determina gradul de utilizare al timpului de lucru și gradul de încărcare pe fiecare compartiment și post de lucru;
 - c. Descrierea fluxurilor informaționale printr-o organigramă unde se precizează: activitățile, compartimentele sau posturile de lucru;
 - d. Gradul de pregătire a unității analizate pentru proiectarea unui sistem informatic ce are în vedere pregătirea personalului;
 - e. Modul în care sunt rezolvate cerințele informaționale din sistem, măsura în care informațiile solicitate lipsesc sau sunt necorespunzătoare.
- 41. Ce urmărește să determine evaluarea costului de funcționare a sistemului informațional?**
- a. Trebuie să pună în evidență gradul de dotare cu tehnică de calcul, ponderea acesteia în sistemul informațional, gradul ei de utilizare și oportunitatea folosirii tehnicii existente în noul sistem;
 - b. Are rolul de a determina gradul de utilizare al timpului de lucru și gradul de încărcare pe fiecare compartiment și post de lucru;
 - c. Presupune punerea în relief a măsurii pentru reducerea cheltuielilor cu funcționarea sistemului informațional;
 - d. Gradul de pregătire a unității analizate pentru proiectarea unui sistem informatic ce are în vedere pregătirea personalului;
 - e. Modul în care sunt rezolvate cerințele informaționale din sistem, măsura în care informațiile solicitate lipsesc sau sunt necorespunzătoare.
- 42. Elaborarea variantelor de realizare a noului sistem presupune parcurgerea următoarelor activități?**
- a. Analiza, proiectarea și elaborarea sistemului informatic;

- b. Stabilirea funcțiilor, cerințelor și restricțiilor noului sistem; definirea soluției globale de principiu pentru prelucrarea automată a datelor și evaluarea fiecărei variante de realizare;
 - c. Activități de tipul tranzacțiilor interne și al tranzacțiilor externe;
 - d. Un ansamblu de intrări, prelucrări și ieșiri definite în funcție de obiectivele noului sistem;
 - e. Activități eterogene și activități omogene.
- 43. În abordarea proiectării generale se poate pleca de la mai multe variante. Care sunt acestea?**
- a. Varianta ieșiri-intrări, varianta intrări-ieșiri și varianta mixtă;
 - b. Varianta top-down și varianta bottom-up;
 - c. Varianta proiectării bazelor de date și varianta proiectării bazei informaționale de intrare;
 - d. Varianta structurală și varianta funcțională;
 - e. Varianta codificării și varianta recodificării.
- 44. Cum se realizează varianta ieșiri-intrări în abordarea proiectării generale?**
- a. După stabilirea obiectivelor se inventariază documentele de intrare care circulă în sistemul agentului economic pentru determinarea bazei informaționale de intrare;
 - b. Se pleacă de la obiectivele sistemului informatic și situațiile de ieșire unde se concretizează acestea, analizând modul de obținere a fiecărei informații se determină baza informațională de intrare, apoi se realizează celelalte faze ale proiectării;
 - c. Se realizează din indicatorii ce se pot calcula din atribute;
 - d. Se elaborează documentația și apoi se analizează obiectivele proiectării;
 - e. Se proiectează varianta inițială și apoi varianta finală a bazei informaționale.
- 45. Cum se realizează varianta intrări-ieșiri în abordarea proiectării generale?**
- a. Se elaborează documentația și apoi se analizează obiectivele proiectării;
 - b. Se analizează modul de obținere a fiecărei informații și apoi se determină baza informațională de intrare, după care se realizează celelalte faze ale proiectării;
 - c. După stabilirea obiectivelor se inventariază documentele de intrare care circulă în sistemul agentului economic pentru determinarea bazei informaționale de intrare;
 - d. Se proiectează varianta inițială și apoi varianta finală a bazei informaționale;
 - e. Se codifică și apoi se realizează proiectarea structurală și funcțională.
- 46. Ieșirile sistemului informatic pot fi privite din mai multe puncte de vedere. Care sunt acestea?**
- a. Tranzacții interne și tranzacții externe;
 - b. Alfabetice, numerice și alfanumerice;
 - c. Elementare și complexe;
 - d. Structural, funcțional și tipologic;
 - e. Secvențiale și abreviate.
- 47. Ce reprezintă ieșirile sistemului informatic privite din punct de vedere structural?**
- a. Concretizează obiectivele generale și specifice ale sistemului;
 - b. Pot fi realizate sub formă de liste, grafice, ieșiri către alte sisteme, sub forma unor colecții organizate de date off-line sau on-line;
 - c. Sunt reprezentate de bazele de date, având o structură eterogenă, funcție de necesitățile agentului economic;
 - d. Sunt reprezentate de informații sintetice și analitice;
 - e. Reprezintă a treia componentă din triada ce caracterizează structura oricărui tip de sistem, respectiv INTRĂRI - PRELUCRĂRI - IEȘIRI.
- 48. Ce reprezintă ieșirile sistemului informatic privite din punct de vedere funcțional?**
- a. Pot fi realizate sub formă de liste, grafice, ieșiri către alte sisteme, sub forma unor colecții organizate de date off-line sau on-line;
 - b. Concretizează obiectivele generale și specifice ale sistemului;
 - c. Sunt reprezentate de bazele de date, având o structură eterogenă, funcție de necesitățile agentului economic;
 - d. Sunt reprezentate de informații sintetice și analitice;
 - e. Reprezintă a treia componentă din triada ce caracterizează structura oricărui tip de sistem, respectiv INTRĂRI - PRELUCRĂRI - IEȘIRI.

- 49. Ce reprezintă ieșirile sistemului informatic privite din punct de vedere tipologic?**
- Pot fi realizate sub formă de liste, grafice, ieșiri către alte sisteme, sub forma unor colecții organizate de date off-line sau on-line;
 - Concretizează obiectivele generale și specifice ale sistemului;
 - Sunt reprezentate de bazele de date, având o structură eterogenă, funcție de necesitățile agentului economic;
 - Sunt reprezentate de informații sintetice și analitice;
 - Reprezintă a treia componentă din triada ce caracterizează structura oricărui tip de sistem, respectiv INTRĂRI - PRELUCRĂRI - IEȘIRI.
- 50. De una din următoarele aspecte trebuie să se țină cont în cadrul proiectării situațiilor de ieșire:**
- Existența bazelor de date distribuite funcționale;
 - Realizarea de copii de siguranță necesare în cazul situațiilor neprevăzute;
 - Destinația situației și momentul când se estimează că se poate obține, nivelul de detaliere a informației și anume: informații mai analitice la nivelurile inferioare ale conducerii și informații de sinteză la nivelul superior;
 - Existența totalurilor pe fiecare nivel analitic în parte;
 - Verificarea datelor în prealabil, mai ales a celor cu caracter criptat.
- 51. Ieșirile sistemului informatic pot fi clasificate din punct de vedere al destinației acestora în:**
- Situații de ieșire ce conțin informații cu caracter sintetic și situații de ieșire ce conțin informații cu caracter analitic;
 - Situații de ieșire pentru activitatea de producție, pentru activitatea de personal, pentru activitatea comercială, pentru activitatea financiar-contabilă;
 - Situații de ieșire destinate altor sisteme și situații de ieșire destinate sistemului propriu;
 - Situații cu caracter operativ, cu caracter periodic și cu caracter aleator;
 - Situații de stare, statistice și previzionale.
- 52. Ieșirile sistemului informatic pot fi clasificate din punct de vedere al specificului funcțiilor generale ale agentului economic în:**
- Situații de ieșire ce conțin informații cu caracter sintetic și situații de ieșire ce conțin informații cu caracter analitic;
 - Situații de ieșire pentru activitatea de producție, pentru activitatea de personal, pentru activitatea comercială, pentru activitatea financiar-contabilă;
 - Situații de ieșire destinate altor sisteme și situații de ieșire destinate sistemului propriu;
 - Situații cu caracter operativ, cu caracter periodic și cu caracter aleator;
 - Situații de stare, statistice și previzionale.
- 53. Ieșirile sistemului informatic pot fi clasificate din punct de vedere al gradului de sintetizare a indicatorilor în:**
- Situații de ieșire ce conțin informații cu caracter sintetic și situații de ieșire ce conțin informații cu caracter analitic;
 - Situații de ieșire pentru activitatea de producție, pentru activitatea de personal, pentru activitatea comercială, pentru activitatea financiar-contabilă;
 - Situații de ieșire destinate altor sisteme și situații de ieșire destinate sistemului propriu;
 - Situații cu caracter operativ, cu caracter periodic și cu caracter aleator;
 - Situații de stare, statistice și previzionale.
- 54. Ieșirile sistemului informatic pot fi clasificate din punct de vedere al momentului generării în:**
- Situații de ieșire ce conțin informații cu caracter sintetic și situații de ieșire ce conțin informații cu caracter analitic;
 - Situații de ieșire pentru activitatea de producție, pentru activitatea de personal, pentru activitatea comercială, pentru activitatea financiar-contabilă;
 - Situații de ieșire destinate altor sisteme și situații de ieșire destinate sistemului propriu;
 - Situații cu caracter operativ, cu caracter periodic și cu caracter aleator;
 - Situații de stare, statistice și previzionale.

- 55. Ieșirile sistemului informatic pot fi clasificate din punct de vedere al intervalului de referință în:**
- Situații de ieșire ce conțin informații cu caracter sintetic și situații de ieșire ce conțin informații cu caracter analitic;
 - Situații de ieșire pentru activitatea de producție, pentru activitatea de personal, pentru activitatea comercială, pentru activitatea financiar-contabilă;
 - Situații de ieșire destinate altor sisteme și situații de ieșire destinate sistemului propriu;
 - Situații cu caracter operativ, cu caracter periodic și cu caracter aleator;
 - Situații de stare, statistice și previzionale.
- 56. Care sunt cele două subfaze pentru proiectarea bazei informaționale de intrare?**
- Faza de analiză și faza proiectării generale;
 - Concretizarea obiectivelor generale și concretizarea obiectivelor specifice ale proiectării;
 - Se elaborează documentația și apoi se analizează obiectivele proiectării;
 - Se proiectează varianta inițială și apoi varianta finală a bazei informaționale;
 - Determinarea conținutului bazei informaționale de intrare și a algoritmilor folosiți și structurarea bazei informaționale în entități.
- 57. Ce presupune structurarea bazei informaționale de intrare în entități?**
- Presupune analiza bazei informaționale de intrare determinată anterior acestui proces și gruparea atributelor în entități;
 - Se proiectează varianta inițială și apoi varianta finală a bazei informaționale;
 - Presupune faza de analiză și faza proiectării generale;
 - Se concretizează de fapt în analiza și proiectarea de detaliu a sistemului informatic;
 - Determinarea tuturor indicatorilor sintetici și analitici folosiți în faza de proiectare.
- 58. În cadrul unei entități pot exista mai multe tipuri de atribute funcție de poziția unui atribut în raport cu acea entitate și care îi conferă acestuia un mod specific de comportare. Care sunt aceste tipuri de atribute?**
- Sintetice și analitice;
 - Cu caracter permanent, variabil și de stare;
 - Cu caracter operativ, cu caracter periodic și cu caracter aleator;
 - Atribute de stare, statistice și previzionale;
 - Atribute off-line și on-line.
- 59. Ce reprezintă atributele permanente în cadrul unei entități?**
- Reprezintă o operație de separare a ansamblului unic de date în mai multe grupe omogene;
 - Se caracterizează prin aceea că valoarea lor variază de la o prelucrare la alta reflectând fenomene și procese economice ce s-au desfășurat de la prelucrarea precedentă. Ele sunt introduse în sistem la fiecare prelucrare și intră în componența entităților variabile;
 - Rămân neschimbate o perioadă mai mare de timp, se preiau în sistem o singură dată și se actualizează la intervale mai mari de timp. Ele intră în componența entităților permanente;
 - Caracterizează nivelul unui element la un moment dat și reprezintă rezultatul unei prelucrări. Ele sunt stocate în baza informațională de unde se preiau periodic, se prelucrează și se înregistrează la noua valoare;
 - Sunt reprezentate de atribute comune care servesc la realizarea de asocieri necesare bazei informaționale de intrare.
- 60. Ce reprezintă atributele variabile în cadrul unei entități?**
- Reprezintă o operație de separare a ansamblului unic de date în mai multe grupe omogene;
 - Sunt reprezentate de atribute comune care servesc la realizarea de asocieri necesare bazei informaționale de intrare;
 - Rămân neschimbate o perioadă mai mare de timp, se preiau în sistem o singură dată și se actualizează la intervale mai mari de timp. Ele intră în componența entităților permanente;
 - Se caracterizează prin aceea că valoarea lor variază de la o prelucrare la alta reflectând fenomene și procese economice ce s-au desfășurat de la prelucrarea precedentă. Ele sunt introduse în sistem la fiecare prelucrare și intră în componența entităților variabile;

- e. Caracterizează nivelul unui element la un moment dat și reprezintă rezultatul unei prelucrări. Ele sunt stocate în baza informațională de unde se preiau periodic, se prelucrează și se înregistrează la noua valoare.
- 61. Ce reprezintă atributele de stare în cadrul unei entități?**
- Sunt reprezentate de atribute comune care servesc la realizarea de asocieri necesare bazei informaționale de intrare;
 - Reprezintă o operație de separare a ansamblului unic de date în mai multe grupe omogene;
 - Rămân neschimbate o perioadă mai mare de timp, se preiau în sistem o singură dată și se actualizează la intervale mai mari de timp. Ele intră în componența entităților permanente;
 - Se caracterizează prin aceea că valoarea lor variază de la o prelucrare la alta reflectând fenomene și procese economice ce s-au desfășurat de la prelucrarea precedentă. Ele sunt introduse în sistem la fiecare prelucrare și intră în componența entităților variabile;
 - Caracterizează nivelul unui element la un moment dat și reprezintă rezultatul unei prelucrări. Ele sunt stocate în baza informațională de unde se preiau periodic, se prelucrează și se înregistrează la noua valoare.
- 62. Codificarea se regăsește ca activitate cu implicații mari în prelucrarea automată a datelor sub două aspecte. Care sunt acestea?**
- O codificare internă a datelor în calculatorul electronic și o codificare externă, ceea ce permite introducerea în prelucrare a unor date cât mai împachetate, mai formalizate;
 - Se regăsește sub forma codurilor cu lungime fixă și a celor cu lungime variabilă;
 - Se regăsește sub forma codurilor complexe și a codurilor elementare;
 - Cele două aspecte principale sunt stabilitatea și unicitatea;
 - Se regăsește sub forma lungimii codului și a capacității sistemului de coduri.
- 63. Care sunt cerințele existenței unui sistem de coduri?**
- Unicitatea, stabilitatea, elasticitatea, conciziunea;
 - Unicitatea, stabilitatea, elasticitatea, conciziunea, claritatea, semnificația și codul să fie operațional;
 - Unicitatea, stabilitatea, elasticitatea, conciziunea, claritatea, semnificația și codul să fie sintetic;
 - Unicitatea, stabilitatea, elasticitatea, conciziunea, claritatea;
 - Unicitatea, stabilitatea, elasticitatea, conciziunea, claritatea, semnificația și codul să fie analitic.
- 64. Ce reprezintă funcția de caracterizare a unui sistem de coduri?**
- Este reprezentată de diversitatea și complexitatea colecțiilor de date, specificul operațiilor de regăsire, sortare, indexare etc. au condus la apariția unei palete variate de coduri;
 - Facilitează introducerea în memorie a acestora, reducerea timpului de prelucrare, inclusiv ușurința folosirii codului de către personalul din compartimentele funcționale implicate în funcționarea sistemului informatic;
 - Presupune existența unei chei de control (formată din unul sau mai multe caractere) pe baza căreia folosind metode și algoritmi specifici să se poată verifica integral corectitudinea simbolurilor care intră în structura codului;
 - Oferă posibilitatea regăsirii mai rapide a elementelor prin intermediul codurilor asociate lor decât prin folosirea completă a semanticii acestora;
 - Asigură exprimarea într-o formă concisă, unică și stabilă în timp, a conținutului semantic al fiecărui element codificat prin intermediul codului asociat acestuia.
- 65. Ce reprezintă funcția de identificare a unui sistem de coduri?**
- Este reprezentată de diversitatea și complexitatea colecțiilor de date, specificul operațiilor de regăsire, sortare, indexare etc. au condus la apariția unei palete variate de coduri;
 - Facilitează introducerea în memorie a acestora, reducerea timpului de prelucrare, inclusiv ușurința folosirii codului de către personalul din compartimentele funcționale implicate în funcționarea sistemului informatic;
 - Presupune existența unei chei de control (formată din unul sau mai multe caractere) pe baza căreia folosind metode și algoritmi specifici să se poată verifica integral corectitudinea simbolurilor care intră în structura codului;

- d. Oferă posibilitatea regăsirii mai rapide a elementelor prin intermediul codurilor asociate lor decât prin folosirea completă a semanticii acestora;
- e. Asigură exprimarea într-o formă concisă, unică și stabilă în timp, a conținutului semantic al fiecărui element codificat prin intermediul codului asociat acestuia.

66. Ce reprezintă funcția de control a unui sistem de coduri?

- a. Este reprezentată de diversitatea și complexitatea colecțiilor de date, specificul operațiilor de regăsire, sortare, indexare etc. au condus la apariția unei palete variate de coduri;
- b. Facilitează introducerea în memorie a acestora, reducerea timpului de prelucrare, inclusiv ușurința folosirii codului de către personalul din compartimentele funcționale implicate în funcționarea sistemului informatic;
- c. Presupune existența unei chei de control (formată din unul sau mai multe caractere) pe baza căreia folosind metode și algoritmi specifici să se poată verifica integral corectitudinea simbolurilor care intră în structura codului;
- d. Oferă posibilitatea regăsirii mai rapide a elementelor prin intermediul codurilor asociate lor decât prin folosirea completă a semanticii acestora;
- e. Asigură exprimarea într-o formă concisă, unică și stabilă în timp, a conținutului semantic al fiecărui element codificat prin intermediul codului asociat acestuia.

67. Ce reprezintă funcția de manipulare a unui sistem de coduri?

- a. Este reprezentată de diversitatea și complexitatea colecțiilor de date, specificul operațiilor de regăsire, sortare, indexare etc. au condus la apariția unei palete variate de coduri;
- b. Facilitează introducerea în memorie a acestora, reducerea timpului de prelucrare, inclusiv ușurința folosirii codului de către personalul din compartimentele funcționale implicate în funcționarea sistemului informatic;
- c. Presupune existența unei chei de control (formată din unul sau mai multe caractere) pe baza căreia folosind metode și algoritmi specifici să se poată verifica integral corectitudinea simbolurilor care intră în structura codului;
- d. Oferă posibilitatea regăsirii mai rapide a elementelor prin intermediul codurilor asociate lor decât prin folosirea completă a semanticii acestora;
- e. Asigură exprimarea într-o formă concisă, unică și stabilă în timp, a conținutului semantic al fiecărui element codificat prin intermediul codului asociat acestuia.

68. În cadrul proiectării codurilor, la ce se referă proiectarea datelor specifice procedurilor?

- a. Cuprinde proceduri de actualizare a fișierelor permanente, proceduri de actualizare a fișierelor variabile, proceduri de actualizare a datelor (fișierelor) de stare, proceduri de obținere a ieșirilor;
- b. Se referă la intrările în proceduri ce sunt constituite din parametri de apel și date de intrare din fișierele de bază;
- c. Se referă la ieșirile din proceduri care sunt constituite din fișierele create sau situațiile de ieșire obținute și trebuie descrise, ca și intrările, cu specificatorul complet de fișier;
- d. Se folosește metoda top-down, pornindu-se de la cele două funcții principale pe care trebuie să le asigure un sistem informatic economic, respectiv actualizarea fișierelor de bază și obținerea ieșirilor proiectate;
- e. Intrările în proceduri, ieșirile din proceduri și fișierele intermediare.

69. Ce cuprinde proiectarea de detaliu a procedurilor?

- a. Proiectarea datelor specifice procedurilor și proiectarea prelucrărilor specifice procedurilor;
- b. Se folosește metoda top-down, pornindu-se de la cele două funcții principale pe care trebuie să le asigure un sistem informatic economic, respectiv actualizarea fișierelor de bază și obținerea ieșirilor proiectate;
- c. Se referă la ieșirile din proceduri care sunt constituite din fișierele create sau situațiile de ieșire obținute și trebuie descrise, ca și intrările, cu specificatorul complet de fișier;
- d. Se referă la intrările în proceduri ce sunt constituite din parametri de apel și date de intrare din fișierele de bază;
- e. Cuprinde proceduri de actualizare a fișierelor permanente, proceduri de actualizare a fișierelor variabile, proceduri de actualizare a datelor (fișierelor) de stare, proceduri de obținere a ieșirilor.

- 70. Ce caracter pot avea operațiile de control în cazul validării datelor din cadrul proiectării procedurilor?**
- Caracter analitic și caracter sintetic, funcție de importanța informației;
 - Caracter sintactic, când se verifică forma sau caracter semantic, când se verifică fondul;
 - Caracter secvențial și caracter aleator;
 - Caracter manual și caracter automat;
 - Caracter de identificare și caracter de caracterizare.
- 71. Care din activitățile următoare nu se regăsește în cadrul celor ce trebuie realizate în etapa de implementare?**
- Crearea bazelor de date pentru realizarea proiectării de detaliu;
 - Asigurarea condițiilor de punere în funcțiune;
 - Funcționarea experimentală și punerea în funcțiune a sistemului proiectat;
 - Definitivarea documentației sistemului informatic;
 - Recepționarea și omologarea noului sistem.
- 72. În funcție de ordinea de testare a subsistemelor componente implementarea poate fi?**
- Aleatoare sau secvențială, funcție de prioritatea modulelor din cadrul sistemului proiectat;
 - Bazată pe proiectarea generală a sistemului informatic sau pe proiectarea de detaliu a sistemului informatic;
 - Simultană, când toate subsistemele se testează odată, fără acordarea de priorități sau în serie, care presupune testarea pe bază de prioritate a subsistemelor, de obicei testându-se mai întâi subsistemele cu frecvența cea mai mare;
 - Pentru fișiere, pentru programe, pentru module sau pentru proceduri;
 - Cea orientată spre date sau cea orientată spre utilizator funcție de preferințele beneficiarului sistemului informatic.
- 73. Prin ce se concretizează documentația finală a unui sistem informatic?**
- Manualul hardware și manualul software al aplicației realizate;
 - Se concretizează în manualul de prezentare, manualul de utilizare și manualul de exploatare;
 - Se concretizează din manualul software al aplicației realizate;
 - Se referă la procesele verbale de predare primire a sistemului informatic;
 - Cuprinde partea de mentenanță a sistemului informatic cu ajutorul inginerilor de sistem.
- 74. Cheltuielile efectuate cu un sistem informatic pot fi grupate în două categorii. Care sunt acestea?**
- Cheltuieli cu achiziționarea hardware-ului și cheltuieli cu realizarea software-ului;
 - Cheltuieli cu software-ul de bază și cheltuieli cu realizarea sistemului informatic;
 - Cheltuieli de realizare și cheltuieli de întreținere-exploatare;
 - Cheltuieli de materiale și cheltuieli de personal;
 - Cheltuieli variabile și cheltuieli fixe.
- 75. Ce indicator reprezintă formula următoare $k_{eff} = \frac{E_{BJ}}{C_{RJ}}$?**
- Termenul de recuperare a investiției suplimentare;
 - Termenul de recuperare a cheltuielilor totale cu realizarea unui subsistem informatic sau a sistemului informatic în ansamblu;
 - Coeficientul eficienței comparate;
 - Coeficientul eficienței la 1000 lei cheltuieli;
 - Coeficientul eficienței economice.
- 76. Ce indicator reprezintă formula următoare $T_{rs} = \frac{1}{k_{comp}} = \frac{C_{R2} - C_{R1}}{E_{B2} - E_{B1}}$?**
- Termenul de recuperare a investiției suplimentare;
 - Termenul de recuperare a cheltuielilor totale cu realizarea unui subsistem informatic sau a sistemului informatic în ansamblu;
 - Coeficientul eficienței comparate;
 - Coeficientul eficienței la 1000 lei cheltuieli;

- e. Coeficientul eficienței economice.
- 77. Calitatea unui sistem informatic se poate privi din două puncte de vedere. Care sunt acestea?**
- a. Din punctul de vedere al erorilor detectate și din punctul de vedere al celor corectate;
 - b. Din punctul de vedere al software-ului realizat și din punctul de vedere al hardware-ului achiziționat;
 - c. Din punctul de vedere al costului de realizare al sistemului informatic și din punctul de vedere al prețului plătit de beneficiar;
 - d. Din punctul de vedere al erorilor detectate și din punctul de vedere al timpului trecut până la rezolvarea acestora;
 - e. Din punct de vedere al utilizatorilor sistemului și din punctul de vedere al celor ce au realizat sistemul.
- 78. Ce reprezintă concepția cu privire la calitate cea orientată tehnic către producție?**
- a. Consideră calitatea ca fiind aptitudinea unui produs sau a unui serviciu de a satisface nevoile utilizatorului;
 - b. Cercetează existența unor abateri ale caracteristicilor produsului realizat față de nivelurile prevăzute în documentația tehnică întocmită înainte de începerea realizării produsului;
 - c. Cea care urmărește costul de realizare al sistemului informatic și în același timp prețul plătit de beneficiar;
 - d. Cea care urmărește numărul erorilor detectate și timpul scurs până la rezolvarea acestora;
 - e. Cea care urmărește un indicator al eficienței comparate cât mai bun.
- 79. Calitatea unui sistem informatic se poate privi din două puncte de vedere. Care sunt acestea?**
- a. Consideră calitatea ca fiind aptitudinea unui produs sau a unui serviciu de a satisface nevoile utilizatorului;
 - b. Cercetează existența unor abateri ale caracteristicilor produsului realizat față de nivelurile prevăzute în documentația tehnică întocmită înainte de începerea realizării produsului;
 - c. Cea care urmărește costul de realizare al sistemului informatic și în același timp prețul plătit de beneficiar;
 - d. Cea care urmărește numărul erorilor detectate și timpul scurs până la rezolvarea acestora;
 - e. Cea care urmărește un indicator al eficienței comparate cât mai bun.
- 80. Care dintre următoarele caracteristici nu se regăsesc printre caracteristicile care permit descrierea calității unui sistem informatic conform normei ISO 9126?**
- a. Capacitatea funcțională;
 - b. Facilitatea în utilizare;
 - c. Redundanța;
 - d. Fiabilitatea;
 - e. Portabilitatea.

RĂSPUNSURI CORECTE PSI

Nr. intreb	Rasp. corect
1	A
2	B
3	D
4	E
5	A
6	B
7	C
8	D
9	E
10	B
11	A
12	B
13	C
14	D
15	E
16	B
17	A
18	B
19	C
20	B
21	D
22	E
23	A
24	B
25	A
26	C
27	B
28	D
29	E
30	B
31	C
32	D
33	E
34	A
35	B
36	C
37	D
38	E
39	A
40	B
41	C
42	B
43	A
44	B
45	C
46	D
47	E
48	B
49	A
50	C
51	C
52	B
53	A
54	D
55	E
56	E

Nr. intreb	Rasp. corect
57	A
58	B
59	C
60	D
61	E
62	A
63	B
64	E
65	D
66	C
67	B
68	E
69	A
70	B
71	A
72	C
73	B
74	C
75	E
76	A
77	E
78	B
79	A
80	C

TESTE GRILA – PROGRAMARE INTERNET

1) FTP reprezinta un protocol pentru:

- a) accesul la distanta la un calculator conectat la Internet, cu conditia ca utilizatorul sa aiba drept de acces la calculatorul respectiv
- b) transferul de fisiere intre calculatoare
- c) posta electronica
- d) denumirea simbolica a calculatoarelor conectate la Internet
- e) transferul sigur si fiabil al datelor intre procesele aplicatiilor ce ruleaza pe calculatoare conectate la Internet

2). DNS reprezinta un protocol pentru:

- a) accesul la distanta la un calculator conectat la Internet, cu conditia ca utilizatorul sa aiba drept de acces la calculatorul respectiv
- b) transferul de fisiere intre calculatoare
- c) posta electronica
- d) denumirea simbolica a calculatoarelor conectate la Internet
- e) transferul sigur si fiabil al datelor intre procesele aplicatiilor ce ruleaza pe calculatoare conectate la Internet

3). Un server reprezinta:

- a) un calculator (program in executie) din cadrul unei retele, care furnizeaza servicii altor calculatoare (programe in executie) din cadrul retelei
- b) un calculator (program in executie) dintr-o retea, care beneficiaza de serviciile altui calculator (sau program)
- c) un canal abstract prin care un calculator (program care ruleaza pe calculator) poate comunica cu exteriorul
- d) o pereche formata dintr-o adresa de IP si un port
- e) o componenta electronica dintr-o retea

4). Un soclu reprezinta:

- a) un calculator (program in executie) din cadrul unei retele, care furnizeaza servicii altor calculatoare (programe in executie) din cadrul retelei
- b) o componenta electronica dintr-o retea
- c) un calculator (program in executie) dintr-o retea, care beneficiaza de serviciile altui calculator (sau program)
- d) un canal abstract prin care un calculator (program care ruleaza pe calculator) poate comunica cu exteriorul
- e) o pereche formata dintr-o adresa de IP si un port

5). Un port reprezinta:

- a) o componenta electronica dintr-o retea
- b) un calculator (program in executie) din cadrul unei retele, care furnizeaza servicii altor calculatoare (programe in executie) din cadrul retelei
- c) un calculator (program in executie) dintr-o retea, care beneficiaza de serviciile altui calculator (sau program)
- d) un canal abstract prin care un calculator (program care ruleaza pe calculator) poate comunica cu exteriorul
- e) o pereche formata dintr-o adresa de IP si un port

6). Avantajul arhitecturii in 3 straturi fata de cea in 2 straturi este:

- a) separa logica de prezentare de logica problemei

- b) sistemul este autocontinut
- c) nu necesita cunostinte despre bazele de date
- d) separa logica datelor de logica de prezentare si logica problemei
- e) necesita expertiza in plus

7). Dezavantajul arhitecturii in 3 straturi fata de cea in 2 straturi este:

- a) separa logica de prezentare de logica problemei
- b) sistemul este autocontinut
- c) separa logica datelor de logica de prezentare si logica problemei
- d) nu separa logica de prezentare de logica problemei
- e) necesita expertiza in plus

8). In cadrul modelului transferului de mesaje:

- a) clientii si serverele comunica prin transfer de mesaje de-a lungul unor canale de comunicatie
- b) exista obiecte rezidente pe un calculator, care pot fi invocate de obiecte rezidente pe alte calculatoare
- c) se asociaza unele portiuni de cod cu anumite evenimente, iar codul va fi executat automat la declansarea evenimentelor respective
- d) utilizatorii comunica prin mesaje cu serverul
- e) procesele comunica prin intermediul unor zone partajate numite *spatii*

9). In cadrul modelului obiectelor distribuite:

- a) clientii si serverele comunica prin transfer de mesaje de-a lungul unor canale de comunicatie
- b) exista obiecte rezidente pe un calculator, care pot fi invocate de obiecte rezidente pe alte calculatoare
- c) utilizatorii comunica prin mesaje cu serverul
- d) se asociaza unele portiuni de cod cu anumite evenimente, iar codul va fi executat automat la declansarea evenimentelor respective
- e) procesele comunica prin intermediul unor zone partajate numite *spatii*

10). In cadrul modelului evenimentelor:

- a) clientii si serverele comunica prin transfer de mesaje de-a lungul unor canale de comunicatie
- b) exista obiecte rezidente pe un calculator, care pot fi invocate de obiecte rezidente pe alte calculatoare
- c) se asociaza unele portiuni de cod cu anumite evenimente, iar codul va fi executat automat la declansarea evenimentelor respective
- d) procesele comunica prin intermediul unor zone partajate numite *spatii*
- e) utilizatorii comunica prin mesaje cu serverul

11). Termenul de multimedia se refera la:

- a) integrarea mai multor tipuri de media in cadrul aceluiasi model de document: text, grafica, imagine, video, etc.
- b) crearea de legaturi intre documente, folosind un mecanism propriu modelului de document
- c) documente care contin legaturi la documente stocate pe alte calculatoare din cadrul unei retele
- d) un document folosit intr-o pagina Web
- e) Un site de prezentare

12). Termenul de hiperdocument se refera la:

- a) integrarea mai multor tipuri de media in cadrul aceluiasi model de document: text, grafica, imagine, video, etc.
- b) crearea de legaturi intre documente, folosind un mecanism propriu modelului de document
- c) documente care contin legaturi la documente stocate pe alte calculatoare din cadrul unei retele
- d) Un site de prezentare
- e) un document folosit intr-o pagina Web

13). Un program de tip softbot reprezinta:

- a) un program care permite regasirea si afisarea paginilor WWW in scopul vizualizarii continutului lor de catre un agent uman
- b) un program care localizeaza diverse pagini WWW in scopul crearii unui index
- c) un identificator simbolic al unei resurse
- d) un limbaj de marcare
- e) Un site de prezentare

14). Un URL reprezinta:

- a) un program care permite regasirea si afisarea paginilor WWW in scopul vizualizarii continutului lor de catre un agent uman
- b) un program de tip softbot care localizeaza diverse pagini WWW in scopul crearii unui index
- c) un identificator simbolic al unei resurse
- d) un document folosit intr-o pagina Web
- e) un limbaj de marcare

15). HTML reprezinta:

- a) un program care permite regasirea si afisarea paginilor WWW in scopul vizualizarii continutului lor de catre un agent uman
- b) un document folosit intr-o pagina Web
- c) un program de tip softbot care localizeaza diverse pagini WWW in scopul crearii unui index
- d) un identificator simbolic al unei resurse
- e) un limbaj de marcare

16). Datele dintr-un document HTML reprezinta:

- a) continutul propriu-zis al documentului
- b) informatii referitoare la structura documentului
- c) o definitie de tip de document
- d) un identificator simbolic al unei resurse
- e) un document tranzitional

17). Marcajele dintr-un document HTML reprezinta:

- a) continutul propriu-zis al documentului
- b) informatii referitoare la structura documentului
- c) un identificator simbolic al unei resurse
- d) o definitie de tip de document
- e) un document tranzitional

18). Un document tranzitional HTML reprezinta:

- a) continutul propriu-zis al documentului
- b) informatii referitoare la structura documentului
- c) o definitie de tip de document
- d) un identificator simbolic al unei resurse
- e) un document pentru verificarea documentelor HTML

19). Care din urmatoarele afirmatii este adevarata in cazul unui document HTML?

- a) elementele de nivel bloc sunt formate intotdeauna incepand cu o linie noua
- b) elementele incluse sunt formate intotdeauna incepand cu o linie noua
- c) elementele incluse nu pot fi despartite pe mai multe linii
- d) elementele de nivel bloc pot fi despartite pe mai multe linii
- e) elementele de nivel bloc nu sunt formate intotdeauna incepand cu o linie noua

20). Care din urmatoarele afirmatii este adevarata in cazul unui document HTML?

- a) elementele de nivel bloc nu sunt formate intotdeauna incepand cu o linie noua
- b) elementele incluse sunt formate intotdeauna incepand cu o linie noua
- c) elementele incluse nu pot fi despartite pe mai multe linii
- d) elementele de nivel bloc pot fi despartite pe mai multe linii
- e) elementele de nivel bloc nu pot fi despartite pe mai multe linii

21). Care din urmatoarele afirmatii este adevarata in cazul unui document HTML?

- a) un element de nivel bloc poate contine atat elemente incluse, cat si elemente de nivel bloc
- b) un element inclus poate contine elemente incluse, date, elemente de nivel bloc
- c) elementele de nivel bloc nu mostenesc directionalitatea de la elementul cuprinzator
- d) elementele incluse mostenesc directionalitatea de la elementul cuprinzator
- e) un element de nivel bloc poate contine doar elemente incluse

22). Termenul de *cookie* se refera la:

- a) mecanismul prin care o aplicatie WWW pe partea de server poate stoca si regasi informatii pe partea de client
- b) mecanismul prin care o aplicatie WWW pe partea de client poate stoca si regasi informatii pe partea de server
- c) mecanismul de creare a unei pagini Web interactive
- d) mecanismul de afisare a informatiilor intr-un browser
- e) mecanismul de comunicare catre client a unor informatii care nu au fost incluse in linia de start

23). Pentru un mesaj HTTP, un antet general:

- a) se refera la mesaj, nu la entitatea transmisa
- b) se refera la entitatea transmisa sau referita
- c) este un mesaj prin care serverul comunica clientului informatii care nu au fost incluse in linia de start
- d) specifica o pagina Web interactiva
- e) nici-o afirmatie nu este adevarata

24). Termenul de *programare pe partea de client* se refera la:

- a) executarea de programe la client in scopul cresterii gradului de interactivitate a paginilor WWW
- b) executarea de programe la server in scopul de a genera date si rapoarte care sunt trimise clientului
- c) utilizarea de pagini Web create pe server
- d) utilizarea de pagini Web create pe calculatorul client
- e) mecanismul de afisare a informatiilor intr-un browser

25). Termenul de *programare pe partea de server* se refera la:

- a) executarea de programe la client in scopul cresterii gradului de interactivitate a paginilor WWW
- b) executarea de programe la server in scopul de a genera date si rapoarte care sunt trimise clientului
- c) utilizarea de pagini Web create pe server
- d) utilizarea de pagini Web create pe calculatorul client
- e) mecanismul de afisare a informatiilor intr-un browser

26). In cadrul DHTML, CSS se refera la:

- a) includerea unor elemente multimedia intr-un document HTML
- b) stilul de prezentare al unui document HTML
- c) incorporarea in cadrul unei pagini a unor programe in format sursa, ce vor fi executate la client
- d) incorporarea in cadrul unei pagini a unor programe in format obiect, ce vor fi executate la client
- e) liantul dintre limbajul de marcare HTML sau XML si celelalte elemente ale DHTML

27). In cadrul DHTML, scripting-ul la client se refera la:

- a) includerea unor elemente multimedia intr-un document HTML
- b) stilul de prezentare al unui document HTML
- c) incorporarea in cadrul unei pagini a unor programe in format sursa, ce vor fi executate la client
- d) incorporarea in cadrul unei pagini a unor programe in format obiect, ce vor fi executate la client
- e) liantul dintre limbajul de marcare HTML sau XML si celelalte elemente ale DHTML

28). In cadrul DHTML, obiectele se refera la:

- a) stilul de prezentare al unui document HTML
- b) incorporarea in cadrul unei pagini a unor programe in format sursa, ce vor fi executate la client
- c) includerea unor elemente multimedia intr-un document HTML
- d) incorporarea in cadrul unei pagini a unor programe in format obiect, ce vor fi executate la client
- e) liantul dintre limbajul de marcare HTML sau XML si celelalte elemente ale DHTML

29). In cadrul DHTML, modelul obiectelor document (DOM) se refera la:

- a) stilul de prezentare al unui document HTML
- b) incorporarea in cadrul unei pagini a unor programe in format sursa, ce vor fi executate la client
- c) incorporarea in cadrul unei pagini a unor programe in format obiect, ce vor fi executate la client
- d) includerea unor elemente multimedia intr-un document HTML
- e) liantul dintre limbajul de marcare HTML sau XML si celelalte elemente ale DHTML

30). O foaie de stil pentru un document HTML este:

- a) o multime de reguli care se aplica unui document HTML in scopul generarii unei prezentari a documentului
- b) un set de programe in format obiect executate la client
- c) o multime de asignari de valori unor variabile
- d) o calitate sau caracteristica pe care o poate avea un element al unui document HTML
- e) un set de actiuni

31). Stilul unui document HTML este:

- a) o multime de reguli care se aplica unui document HTML in scopul generarii unei prezentari a documentului
- b) un set de programe in format obiect executate la client
- c) o multime de asignari de valori unor variabile
- d) o calitate sau caracteristica pe care o poate avea un element al unui document HTML
- e) un set de actiuni

32). O proprietate in HTML este:

- a) o multime de reguli care se aplica unui document HTML in scopul generarii unei prezentari a documentului
- b) o multime de asignari de valori unor variabile
- c) un program in format obiect executat la client
- d) o calitate sau caracteristica pe care o poate avea un element al unui document HTML
- e) un set de actiuni

33). Un selector intr-un document HTML este:

- a) o multime de reguli care se aplica unui document HTML in scopul generarii unei prezentari a documentului
- b) o multime de asignari de valori unor variabile
- c) o calitate sau caracteristica pe care o poate avea un element al unui document HTML
- d) un program in format obiect executat la client
- e) un set de conditii dintr-o regula de stil

34). Care din afirmatiile urmatoare este adevarata?

- a) Java are legare la compilare
- b) JavaScript are legare la compilare
- c) JavaScript foloseste un model obiectual bazat pe clase
- d) Java foloseste un model obiectual bazat pe prototipuri
- e) Java foloseste un interpretor

35). Care din afirmatiile urmatoare este adevarata?

- a) Java are legare la executie
- b) JavaScript are legare la compilare
- c) JavaScript foloseste un model obiectual bazat pe clase
- d) JavaScript foloseste un model obiectual bazat pe prototipuri
- e) Java foloseste un interpretor

36). Modelul de programare folosit la programarea interfetelor grafice se numeste:

- a) programare orientata pe evenimente
- b) programare orientata pe obiecte
- c) programare orientata pe mesaje
- d) programare functionala
- e) programare logica

37). Care din afirmatiile urmatoare este adevarata?

- a) este indicat sa se separe partea de continut de partea de prezentare dintr-o pagina
- b) descarcarea unor programe in cod obiect si rularea lor in programul navigator nu duce la incetinirea incarcarii paginii
- c) este indicata folosirea cat mai des a scriptingului la client
- d) nu este indicat sa se separe partea de continut de partea de prezentare dintr-o pagina
- e) nici-o afirmatie nu este adevarata

38). Gazdele virtuale usureaza activitatea de administrare a serverului deoarece:

- a) exista mai multe instante ale serverului
- b) exista mai multe configuratii ale serverului
- c) se monitorizeaza mai multe servere
- d) se monitorizeaza executia unui singur server
- e) nici-o afirmatie nu este adevarata

39). Gazdele virtuale ingreuneaza activitatea de administrare a serverului deoarece:

- a) exista mai multe instante ale serverului
- b) exista mai multe configuratii ale serverului
- c) se monitorizeaza executia mai multor servere
- d) nu exista decat un singur server
- e) nici-o afirmatie nu este adevarata

40). Avantajul gazdelor virtuale non-IP este:

- a) este nevoie de o adresa de IP distincta pentru fiecare gazda virtuala
- b) asignarea mai multor adrese de IP unei singure masini nu creeaza probleme retelei
- c) este nevoie de o singura adresa de IP pentru deservirea tuturor gazdelor virtuale
- d) se poate aplica pentru orice versiune de HTTP
- e) nu este necesara o singura adresa de IP pentru deservirea tuturor gazdelor virtuale

41). Dezavantajul gazdelor virtuale IP este:

- a) este nevoie de o adresa de IP distincta pentru fiecare gazda virtuala
- b) asignarea mai multor adrese de IP unei singure masini nu creeaza probleme retelei
- c) este nevoie de o singura adresa de IP pentru deservirea tuturor gazdelor virtuale
- d) se poate aplica numai pentru anumite versiuni de HTTP
- e) o singura adresa IP deserveste toate gazdele virtuale

42). In cazul serverelor de origine:

- a) cererile HTTP sunt tratate local de server
- b) acestea actioneaza ca intermediari intre client si serverul de origine
- c) cererile sunt filtrate
- d) cererile HTTP nu sunt tratate local de server
- e) se trece de la un protocol nesigur la unul sigur

43). SSI este:

- a) o interfata pentru comunicarea dintre un server de informatii (cum este cazul unui server WWW) si un program de aplicatie
- b) o tehnologie ce permite includerea de informatii intr-o pagina WWW inainte de trimiterea paginii la client
- c) o interfata independenta de limbaj
- d) un set de reguli referitoare la formatarea unui document HTML
- e) un standard pentru trimiterea de informatii catre client

44). CGI este:

- a) o interfata pentru comunicarea dintre un server de informatii (cum este cazul unui server WWW) si un program de aplicatie
- b) o tehnologie ce permite includerea de informatii intr-o pagina WWW inainte de trimiterea paginii la client
- c) un set de reguli referitoare la formatarea unui document HTML
- d) o interfata dependenta de limbajul folosit
- e) o tehnologie care nu implica apelul nici unui program extern

45). O aplicatie Web orientata pe prezentare:

- a) contine pagini WWW interactive
- b) este o interfata independenta de limbajul folosit
- c) nu este capabila sa genereze continut dinamic ca raspuns la cererile clientilor
- d) implementeaza un punct terminal pentru un serviciu WWW
- e) este o componenta Web

46). O aplicatie Web orientata pe serviciu:

- a) contine pagini WWW interactive
- b) este capabila sa genereze continut dinamic ca raspuns la cererile clientilor
- c) implementeaza un punct terminal pentru un serviciu WWW
- d) este o interfata independenta de limbajul folosit
- e) este o componenta Web

47). In vederea pastrarii starii comunicarii dintre un client si un server WWW in cadrul unei sesiuni de lucru, folosirea campurilor ascunse presupune:

- a) transmiterea repetata de la server catre client si apoi de la client catre server a istoricului interactiunii sub forma unei multimi de campuri ascunse intr-un formular HTML
- b) adaugarea de informatii suplimentare la un URL pentru a identifica in mod unic o sesiune
- c) identificarea si autentificarea utilizatorului la primul sau acces asupra serverului

- d) folosirea unor informatii pe care serverul WWW le poate stoca pe calculatorul clientului
- e) ascunderea informatiilor referitoare la sursa paginii Web

48). In vederea pastrarii starii comunicarii dintre un client si un server WWW in cadrul unei sesiuni de lucru, rescrierea URL-ului presupune:

- a) transmiterea repetata de la server catre client si apoi de la client catre server a istoricului interactiunii sub forma unei multimi de campuri ascunse intr-un formular HTML
- b) adaugarea de informatii suplimentare la un URL pentru a identifica in mod unic o sesiune
- c) identificarea si autentificarea utilizatorului la primul sau acces asupra serverului
- d) ascunderea informatiilor referitoare la sursa paginii Web
- e) folosirea unor informatii pe care serverul WWW le poate stoca pe calculatorul clientului

49). In vederea pastrarii starii comunicarii dintre un client si un server WWW in cadrul unei sesiuni de lucru, folosirea autentificarii HTTP presupune:

- a) transmiterea repetata de la server catre client si apoi de la client catre server a istoricului interactiunii sub forma unei multimi de campuri ascunse intr-un formular HTML
- b) adaugarea de informatii suplimentare la un URL pentru a identifica in mod unic o sesiune
- c) identificarea si autentificarea utilizatorului la primul sau acces asupra serverului
- d) ascunderea informatiilor referitoare la sursa paginii Web
- e) folosirea unor informatii pe care serverul WWW le poate stoca pe calculatorul clientului

50). In vederea pastrarii starii comunicarii dintre un client si un server WWW in cadrul unei sesiuni de lucru, folosirea cookie-urilor presupune:

- a) transmiterea repetata de la server catre client si apoi de la client catre server a istoricului interactiunii sub forma unei multimi de campuri ascunse intr-un formular HTML
- b) adaugarea de informatii suplimentare la un URL pentru a identifica in mod unic o sesiune
- c) identificarea si autentificarea utilizatorului la primul sau acces asupra serverului
- d) folosirea unor informatii pe care serverul WWW le poate stoca pe calculatorul clientului
- e) ascunderea informatiilor referitoare la sursa paginii Web

51). In cadrul prelucrarilor efectuate de un miniserver, metodele init() si destroy() implementeaza:

- a) acele prelucrari care asigura persistenta informatiei stocate de miniserver
- b) prelucrarile legate de tratarea cererilor primite de la client
- c) aflarea de informatii de configurare
- d) ascunderea informatiilor referitoare la sursa paginii Web
- e) nimic, deoarece nu exista

52). In cadrul prelucrarilor efectuate de un miniserver, metodele service() si doYYY() implementeaza:

- a) prelucrarile care asigura persistenta informatiei stocate de miniserver
- b) prelucrarile legate de tratarea cererilor primite de la client
- c) aflarea de informatii de configurare
- d) ascunderea informatiilor referitoare la sursa paginii Web
- e) nimic, deoarece nu exista

53). In cadrul prelucrarilor efectuate de un miniserver, metoda getServletConfig() implementeaza:

- a) prelucrarile care asigura persistenta informatiei stocate de miniserver
- b) prelucrarile legate de tratarea cererilor primite de la client
- c) aflarea de informatii de configurare
- d) nimic, deoarece nu exista

e) ascunderea informatiilor referitoare la sursa paginii Web

54). In cadrul JDBC, interfata DriverManager:

- a) gestioneaza programele de control disponibile pentru conectarea la diverse baze de date
- b) este utila pentru crearea si executarea de cereri SQL
- c) permite accesul la informatiile de configurare
- d) se utilizeaza pentru reprezentarea multimii de rezultate obtinute in urma executiei unei interogari SQL
- e) este utila pentru pentru reprezentarea metadatelor referitoare la o multime de rezultate

55). In cadrul JDBC, interfata Statement:

- a) gestioneaza programele de control disponibile pentru conectarea la diverse baze de date
- b) este utila pentru crearea si executarea de cereri SQL
- c) permite accesul la informatiile de configurare
- d) se utilizeaza pentru reprezentarea multimii de rezultate obtinute in urma executiei unei interogari SQL
- e) este utila pentru pentru reprezentarea metadatelor referitoare la o multime de rezultate

56). In cadrul JDBC, interfata ResultSet:

- a) gestioneaza programele de control disponibile pentru conectarea la diverse baze de date
- b) este utila pentru crearea si executarea de cereri SQL
- c) permite accesul la informatiile de configurare
- d) se utilizeaza pentru reprezentarea multimii de rezultate obtinute in urma executiei unei interogari SQL
- e) este utila pentru pentru reprezentarea metadatelor referitoare la o multime de rezultate

57). In cadrul JDBC, interfata ResultSetMetaData:

- a) gestioneaza programele de control disponibile pentru conectarea la diverse baze de date
- b) este utila pentru crearea si executarea de cereri SQL
- c) se utilizeaza pentru reprezentarea multimii de rezultate obtinute in urma executiei unei interogari SQL
- d) permite accesul la informatiile de configurare
- e) este utila pentru pentru reprezentarea metadatelor referitoare la o multime de rezultate

58).Care din afirmatiile urmatoare nu este adevarata?

- a) Metadatele sunt date pentru descrierea altor date
- b) Metadatele sunt utile atunci cand nu se cunoaste structura bazei de date
- c) Metadatele sunt utile pentru testarea existentei anumitor facilitati
- d) Metadatele nu pot fi folosite pentru implementarea unor programe utilitare
- e) Datele se pot descrie cu ajutorul unor structuri numite metadate

59). Care din afirmatiile urmatoare este adevarata?

- a) Metadatele nu exista
- b) Metadatele sunt date pentru descrierea altor date
- c) Metadatele sunt utile atunci cand se cunoaste structura bazei de date
- d) Metadatele nu sunt utile pentru testarea existentei anumitor facilitati
- e) Metadatele nu pot fi folosite pentru implementarea unor programe utilitare

60). In cadrul rezervei de conexiuni, prealocarea conexiunilor pe constructorul clasei inseamna ca:

- a) constructorul primeste un numar maxim de conexiuni, un numar initial de conexiuni si un indicator daca rezerva sa astepte sau nu cand se cere o conexiune si nu exista nici una libera

- b) nu este necesara gestiunea ulterioara a alocarilor de conexiuni
- c) daca se cere o conexiune si exista o conexiune libera, atunci ea este alocata si trecuta in lista conexiunilor ocupate
- d) daca se cere o conexiune, nu exista nici una libera si nu s-a atins numarul maxim de conexiuni, atunci se creeaza o noua conexiune pe un fir separat
- e) asteptarea pentru eliberarea unei conexiuni

61). In cadrul rezervei de conexiuni, gestiunea conexiunilor disponibile inseamna ca:

- a) constructorul primeste un numar maxim de conexiuni, un numar initial de conexiuni si un indicator daca rezerva sa astepte sau nu cand se cere o conexiune si nu exista nici una libera
- b) daca se cere o conexiune si exista o conexiune libera, atunci ea este alocata si trecuta in lista conexiunilor ocupate
- c) daca se cere o conexiune, nu exista nici una libera si nu s-a atins numarul maxim de conexiuni, atunci se creeaza o noua conexiune pe un fir separat
- d) nu este necesara alocarea ulterioara de conexiuni
- e) este necesara asteptarea pentru eliberarea unei conexiuni

62). In cadrul rezervei de conexiuni, alocarea de noi conexiuni inseamna ca:

- a) constructorul primeste un numar maxim de conexiuni, un numar initial de conexiuni si un indicator daca rezerva sa astepte sau nu cand se cere o conexiune si nu exista nici una libera
- b) daca se cere o conexiune si exista o conexiune libera, atunci ea este alocata si trecuta in lista conexiunilor ocupate
- c) daca se cere o conexiune, nu exista nici una libera si nu s-a atins numarul maxim de conexiuni, atunci se creeaza o noua conexiune pe un fir separat
- d) asteptarea pentru eliberarea unei conexiuni
- e) nu este necesara gestiunea ulterioara a alocarilor de conexiuni

63). In starea de autovalidare a unei baze de date:

- a) nu orice cerere de actualizare declanseaza automat modificarea continutului bazei de date
- b) orice cerere de actualizare declanseaza automat modificarea continutului bazei de date
- c) modificarea continutului bazei de date are loc atunci cand se executa comanda commit
- d) modificarea continutului bazei de date are loc atunci cand se executa comanda rollback
- e) nu exista starea de autovalidare

64). In starea de validare manuala a unei baze de date:

- a) orice cerere de actualizare declanseaza automat modificarea continutului bazei de date
- b) modificarea continutului bazei de date are loc atunci cand se executa comanda autocommit
- c) modificarea continutului bazei de date are loc atunci cand se executa comanda commit
- d) modificarea continutului bazei de date are loc atunci cand se executa comanda rollback
- e) nu exista starea de validare manuala

65). Intr-un document XML, o radacina:

- a) indica faptul ca este vorba de un document XML
- b) contine o definitie de tip sau o referinta la definitia de tip
- c) contine in interiorul sau toate celelalte elemente
- d) este o portiune de text util in prezentarea unei pagini Web
- e) este o portiune de text optional

66). SAX:

- a) defineste o multime de interfete pentru un analizor sintactic bazat pe evenimente
- b) traduce un document XML intr-o structura de date abstracta de tip arbore
- c) defineste o multime de interfete pentru un analizor sintactic care nu este bazat pe evenimente

- d) este o specificatie independenta de limbaj
- e) transforma un arbore intr-un document XML

67). DOM:

- a) defineste o multime de interfete pentru un analizor sintactic bazat pe evenimente
- b) traduce un document XML intr-o structura de date abstracta de tip arbore
- c) defineste o multime de interfete pentru un analizor sintactic care nu este bazat pe evenimente
- d) este o specificatie dependenta de limbaj
- e) transforma un arbore intr-un document XML

68). Care din afirmatiile urmatoare nu este adevarata?

- a) XSL este un limbaj de prelucrare a documentelor XML
- b) radacina XML contine in interiorul sau toate celelalte elemente
- c) XML permite transformarea unui document XSLT intr-un alt document XSLT sau document text
- d) XML-FO permite descrierea formatului de prezentare a unui document XML
- e) XSL este o aplicatie XML

69). In XSLT, motorul transformarii este:

- a) documentul XML de intrare
- b) documentul XML de iesire
- c) un document XSLT
- d) procesorul XSLT
- e) o structura de tip arbore

70). Pentru XSLT, o instructiune de prelucrare reprezinta:

- a) nodul de start al documentului
- b) un element al sursei XML
- c) textul unei instructiuni de prelucrare din sursa XML
- d) un atribut al unui element din sursa XML
- e) o linie din documentul XML de iesire

71). In cadrul CSS, in regula de stil $P \{color:green\}$, partea P reprezinta:

- a) un selector
- b) o declaratie
- c) o asignare
- d) o proprietate
- e) o valoare

72). In cadrul CSS, in regula de stil $P \{color:green\}$, partea $\{color:green\}$ reprezinta:

- a) un selector
- b) o declaratie
- c) o asignare
- d) o proprietate
- e) o valoare

73). In cadrul CSS, in regula de stil $P \{color:green\}$, partea $color:green$ reprezinta:

- a) un selector
- b) o declaratie
- c) o asignare
- d) o proprietate
- e) o valoare

74). In cadrul CSS, in regula de stil $P \{color:green\}$, partea *color* reprezinta:

- a) un selector
- b) o declaratie
- c) o asignare
- d) o proprietate
- e) o valoare

75). In cadrul CSS, in regula de stil $P \{color:green\}$, partea *green* reprezinta:

- a) un selector
- b) o declaratie
- c) o asignare
- d) o proprietate
- e) o valoare

76). In cadrul CSS, codul:

```
<HTML>
  <HEAD>
    <TITLE>
      Ce reprezinta?
    </TITLE>
  </HEAD>
  <BODY>
    <P STYLE = "font-size: 20pt">Text</P>
    <P STYLE = "font-size: 20pt; color: #0000FF">Text</P>
  </BODY>
</HTML>
```

reprezinta:

- a) specificarea stilului in cadrul unui element individual HTML
- b) specificarea stilului in antetul unui fisier HTML
- c) specificarea stilului intr-un fisier de stil separat
- d) nu specifica stilul
- e) specificarea preluarii stilului implicit din browser

77). In cadrul CSS, codul:

```
<HTML>
  <HEAD>
    <TITLE>Ce reprezinta?
  </TITLE>
  <STYLE TYPE = "text/css">
    EM { background-color: #8000FF; color: white }
    H1 { font-family: Arial, sans-serif }
    P { font-size: 18pt }
    .blue { color: blue }
  </STYLE>
</HEAD>
<BODY>
  <H1 CLASS = "blue">Un antet</H1>
  <P>Un paragraf</P>
  <H1>Alt antet</H1>
  <P CLASS = "blue">Alt <EM>paragraf</EM></P>
</BODY>
</HTML>
```

reprezinta:

- a) specificarea stilului in cadrul unui element individual HTML
- b) specificarea stilului in antetul unui fisier HTML
- c) specificarea stilului intr-un fisier de stil separat
- d) nu specifica stilul
- e) specificarea preluarii stilului implicit din browser

78). In cadrul CSS, codul:

```

<HTML>
<HEAD>
<TITLE>
  Ce reprezinta?
  <LINK REL="stylesheet"
  TYPE="text/css"
  HREF="css3.css"/>
</TITLE>
</HEAD>
<BODY>
  <H1 CLASS = "blue">Un antet</H1>
  <P>Un paragraf</P>
  <H1>Alt antet</H1>
  <P CLASS =
  "blue">Alt<EM>paragraf</EM></P>
</BODY>
</HTML>

```

reprezinta:

- a) specificarea stilului in cadrul unui element individual HTML
- b) specificarea stilului in antetul unui fisier HTML
- c) preluarea stilului dintr-un fisier de stil separat
- d) nu specifica stilul
- e) specificarea preluarii stilului implicit din browser

79). Codul XML

```
<?xml version="1.0" standalone="yes"
  encoding="UTF-8" ?>
<!-- Ce reprezinta -->
<ARBBIN>
  <INFO>a</INFO>
  <ARBBIN>
    <INFO>b</INFO>
    <ARBBIN/>
    <ARBBIN>
      <INFO>d</INFO>
      <ARBBIN/>
      <ARBBIN/>
    </ARBBIN>
  </ARBBIN>
</ARBBIN>
<ARBBIN>
  <INFO>c</INFO>
  <ARBBIN>
    <INFO>e</INFO>
    <ARBBIN/>
    <ARBBIN/>
  </ARBBIN>
  <ARBBIN/>
</ARBBIN>
</ARBBIN>
```

reprezinta:

- a) un document XML ce defineste un arbore binar
- b) o definitie de tip de document pentru un document XML
- c) reprezentarea informatiei din cadrul unui nod al unui arbore XML ca valoare a unui *atribut*
- d) o transformare XSLT aplicata unui document XML
- e) un sablon XSL pentru transformarea unu document XML

80). Codul XML

```
<?xml version="1.0" standalone="yes" ?>
<!-- Un arbore binar reprezentat in XML -->
<!DOCTYPE ARBBIN [
  <!ELEMENT ARBBIN (INFO,ARBBIN,ARBBIN)? >
  <!ELEMENT INFO (#PCDATA) >
]>
<ARBBIN> ...
```

reprezinta:

- a) un document XML ce defineste un arbore binar
- b) o definitie de tip de document pentru un document XML
- c) reprezentarea informatiei din cadrul unui nod al unui arbore XML ca valoare a unui *atribut*
- d) o transformare XSLT aplicata unui document XML
- e) un sablon XSL pentru transformarea unu document XML

RASPUNSURI CORECTE – PROGRAMARE INTERNET

Nr. intrebare	Raspuns corect	Punctaj
1	b	1
2	d	1
3	a	1
4	e	1
5	d	1
6	d	1
7	e	1
8	a	1
9	b	1
10	c	1
11	a	1
12	b	1
13	b	1
14	c	1
15	e	1
16	a	1
17	b	1
18	e	1
19	a	1
20	e	1
21	a	1
22	a	1
23	a	1
24	a	1
25	b	1
26	b	1
27	c	1
28	d	1
29	e	1
30	a	1
31	c	1
32	d	1
33	e	1
34	a	1
35	d	1
36	a	1
37	a	1
38	d	1
39	e	1
40	c	1
41	a	1
42	a	1
43	b	1
44	a	1
45	a	1
46	c	1
47	a	1

48	b	1
49	c	1
50	d	1
51	a	1
52	b	1
53	c	1
54	a	1
55	b	1
56	d	1
57	e	1
58	d	1
59	b	1
60	a	1
61	b	1
62	c	1
63	b	1
64	c	1
65	c	1
66	a	1
67	b	1
68	c	1
69	d	1
70	c	1
71	a	3
72	b	3
73	c	3
74	d	3
75	e	3
76	a	3
77	b	3
78	c	3
79	a	3
80	b	3