UNIVERSITATEA DE STAT DIN MOLDOVA

FACULTATEA de Fizică și Inginerie

APROB APROB

Catedra Consiliul facultății

*Fizica Aplicată și Informatica*  *Fizică și Inginerie*

Pr.verbal Nr. din Pr .verbal Nr. din

***Tehnologii Informaţionale***

Curriculum

pentru examenul de licență

**(specialitatea Tehnologii Informaţionale)**

CHIȘINĂU 2018

1. **Preliminarii**

Scopul examenului de licență constă în demonstrarea cunoștințelor și competențelor integrate la realizarea programului de studii în domeniul 526 Ingineria Sistemelor şi Calculatoarelor, specialitatea 526.2 – Tehnologii Informaționale.

Prin această evaluare finală, absolventul manifestă capacități de a specifica şi rezolva probleme de proiectare a sistemelor informaționale profesional orientate, utilizînd diferite metode și soluții, de a formula problema proiectării sistemice şi integrarea reţelelor locale şi globale, de a expune şi rezolva probleme, legate de organizarea dialogului dintre om şi sistemul informațional. La susținerea examenului de licență absolventul demonstrează nivelul cunoștințelor obținute, crearea şi implementarea sistemelor informaţionale profesional orientate in domeniul dat de activitate. În același timp, examenul de licență arată raționamentele studentului, privind problemele relevante de ordin social și etic.

Susținerea examenului de licență finalizează studiile superioare de licență (ciclul I) cu acordarea titlului de *Licenţiat în Inginerie şi activităţi inginereşti*.

Absolvenții, care au dezvoltat competențele prevăzute în planul de învățămînt, au acces la studii superioare de masterat, ciclul II.

1. **Competențele evaluate**

***Competențe generale:***

* Conceperea şi realizarea produselor hardware şi/sau software competitive pe piaţă,
* Alegerea unui design contemporan al produselor hardware şi/sau software şi al mediului industrial,
* Promovarea inovaţiilor şi a transferului tehnologic,
* Promovarea tehnologiilor informaţionale performante,
* Utilizarea calculatoarelor, sistemelor de conducere şi informatice avansate,
* Utilizarea structurilor organizatorice performante,
* Managementul eficient;

***Competențe specifice:***

* + Specificarea şi rezolvarea problemelor de proiectare a sistemelor informaţionale profesional orientate, utilizînd diferite metode şi soluţii;
  + Formularea problemei proiectării sistemice şi integrarea reţelelor locale şi globale;
  + Expunerea şi rezolvarea problemelor, legate de organizarea dialogului dintre om şi sistemul informaţional;
  + Alegerea resurselor de interfaţare la construirea unor sisteme informaţionale complexe,
  + Formularea cerinţelor tehnico-economice principale pentru proiectarea sistemelor informaţionale profesional orientate;
  + Crearea şi implementarea sistemelor informaţionale profesional orientate in domeniul dat de activitate;
  + Elaborarea documentaţiei tehnico-normative, a sistemelor de standardizare şi certificare;
  + Elaborarea politicii de preţuri in utilizarea sistemelor informaţionale profesional orientate.

1. **Subiecte/ Probleme**

|  |  |  |
| --- | --- | --- |
| ***Finalități*** | ***Repere de conținut*** | |
| 1. **Structura Sistemelor de Calcul** | | |
| ***Arhitectura calculatorului*** | | |
| * să definească bazele de numeraţie şi metodele de convertire a numerelor între diferite baze de numeraţie; * să aplice metodele de reprezentare a numerelor în virgulă fixă utilizînd codul direct, codul invers şi codul complementar; * să aplice metodele de reprezentare a numerelor în virgulă mobilă; | * 1. **Forme de reprezentare a informaţiei în calculator**   Reprezentarea în virgulă fixă a numerelor. Codul direct. Codul complementar. Codul invers.  Reprezentarea în virgulă mobilă a numerelor. | |
| * să definească noţiunea de calculator digital; * să identifice principiile care stau la baza modelului von Neumann a unui calculator numeric; * să determine componentele unui calculator numeric în baza principiilor care stau la baza modelului von Neumann; * să reproducă schema funcţională a calculatorului numeric în baza modelului von Neumann. | * 1. **Arhitectura sistemelor de calcul după modelul von Neumann**   Definirea calculatorului digital. Principiul „programului memorat”. Structura de bază a unui calculator digital conform modelului von Neumann. Principiul de funcţionare a unui calculator digital. | |
| ***Limbaje de asamblare*** | | |
| * să descrie structura procesoarelor din familia Intel x86; * să construiască schema logică a structurii interne a procesorului; * să aplice instrucțiunile de transfer și interschimbare între registre; | * 1. **Registrele procesorului Intel x86**   Registrele procesorului Intel x86 pe 8, 16 și 32 de biți. Instrucțiuni de transfer între registre. Instrucțiuni de interschimbare între registre. | |
| * să definească modurile de adresare la memorie.să argumenteze necesitatea împărțirii memoriei calculatorului în segmente; * să deducă formula de aflare a adresei logice din adresa efectivă; | * 1. **Transferuri de date și operații aritmetice cu memoria**   Componentele unei adrese logice. Componentele adresei efective. Moduri de adresare. | |
| ***Sisteme de operare*** | | |
| * să definească destinaţia RAID (Redundant Arraz of Inexpensive/Independent Disks). Tipuri de RAID; * să descrie structura unui fişier Linux. Timpii atime, ctime şi mtime; * să explice procesul de încărcare a SO Linux, Boot Loaderul şi Boot Manager. | 1. Utilizarea hard-discurilor în SO Linux   Destinaţia RAID şi tipurile. SO Linux, Boot Loaderul şi Boot Manager. | |
| ***Proiectarea sistemelor de operare*** | | |
| * Să descrie componentele SO; * Să detrmine structura SO;   Să proiecteze şi să implimenteze un SO. | **1. Structura Sistemelor de operare**  Componentele sistemelor de operare  Sarcinile sistemului de operare  Structura sistemelor de operare  Proiectarea şi implementarea sistemelor de operare | |
| * Să definească noţiunea de proces şi tare a procesului, secţiune critică, situaţie moartă; * Să explice operaţiile ce au loc asupra proceselor; * Să descrie diferite tipuri şi nivele de planificare a proceselor, diferiţi algoritmi de sincronizare; * Să cunoască diferiţi algoritmi de planificare şi diferite modalităţi de cooperare a proceselor, diferite mecanisme de sincronizare. | **2. Procese**  Noţiunea de proces. Definiţii  Planificarea unităţii centrale (UC)  Starea de interblocare | |
| * Să definească noţiunea de memorie, diferite tipuri de organizarea fizică a memoriei; * Să descrie schemele de gestiune a memoriei; * Să creeze diferite strategii de gestionare a memoriei; * Să creeze diferite scheme de gestiune a memoriei în sistemul de operare. | **3. Gestionarea memoriei**  Suprapuneri (Overlays)  Încărcarea dinamică  Gestionarea memoriei de către sistemul de operare; | |
| * Să defmeasca noţiunele de sistem de fişiere şi principalele lui funcţii; * Să evidenţieze organizarea fişierelor şi accesul către ele; * Să gestioneze fişierele şi directoriile în sistemul de operare; * Să opereze cu fişierele şi directoriile. | **4. Sistemul de fișiere**  Structuri şi tipuri de fişiere  Operaţii cu fişiere  Directoare  Implementarea sistemului de fişiere  Acces concurent la nivel de process  Protecţie fişierilor  Performanţele sistemelor de fişiere | |
| ***Tehnici și protocale de comunicare*** | | |
| ***-*** Să cunoască protocoalele nivelului legătură de date.  ***-*** Să cunoască metodele de detectare si corectare a erorilor de transmisie.  ***-*** Sa utilizeze metodele de detectare/corectare a erorilor de transmisie.  ***-*** Să implementeze măsuri de reducere a influențelor factorilor perturbatori în rețea. | **1. Protocoale de nivel legătura de date**  Serviciile oferite de nivelul legătură de date  Metode de detectare a erorilor de transmisie a  datelor prin rețea  Codul Hamming de corectare a erorilor de transmisie.  Protocolul cu oprire şi aşteptare (Stop and Wait)  Protocolul cu fereastră glisantă (Sliding Window Protocol)  Familia de protocoale HDLC (SDLC, LAPB, LAPD, PPP, LLC, s.a.) | |
| * Să cunoască funcțiile si specificul protocoalelor nivelului legatură de date. * Să planifice transferul fiabil al informației între două sisteme terminale. * Să asigure ordinea corectă de transmitere a pachetelor de date. | **2. Protocoale de nivel transport**  Protocoalele TCP, UDP, RTP, Wireless TCP  Probleme de performanță și soluții  Utilizarea serviciilor de transport  Modelul client-server  Stabilirea și eliberarea conexiunii de transport | |
| ***-***  Să cunoască protocoalele nivelului retea.  ***-***  Să cunoască metodele de determinare a căii optime pentru realizarea transferului.  - Sa utilizeze protocoalele de rutare a datelor intre retele.  - Să selecteze protocolul adecvat pentru comunicarea intre rețele în condițiile specificate. | **3. Protocoale de nivel rețea**  Protocolul IP, IPv4.  Adresarea in Internet  Funcţionarea subreţelelor si calculul claselor de adresare | |
| ***Proiectarea rețelelor de calculatoare*** | | |
| – Se relateze despre caracteristicile de bază, domeniile de utilizare a reţelelor, diagramele logice şi fizice de reţea, instrumente de realizare a diagramelor;  – Să definească conceptul de reţea, subreţea, inter-reţea, nod intermediar şi nod final;  – Să relateze despre trei moduri de adresare în reţele (MAC, IP, DNS), clasele adreselor IP, numărul maximal de reţele şi noduri pe clase, ierarhia numelor în cadrul DNS, rutarea CIDR;  – Să explice funcţiile fiecărui nivel conform modelului OSI;  – Să explice funcţiile fiecărui nivel TCP/IP şi protocoalelor TCP, UDP, IP, NetBIOS, ARP/RARP;  – Să descrie principiile, modurile şi utilajele de interconectare a reţelelor conform nivelelor OSI (hub, switch/bridge, router);  – Să descrie principiile de funcţionare a punţilor/comutatoarelor, routerelor (tabelul de rutare, rutarea statică/dinamică, algoritmi de rutare);  – Să caracterizeze mediile de cablate de transmisie (cablul coaxial, torsadat, de fibra optică) şi fără fir (unde radio);  – Să relateze despre modurile de asigurare securităţii în reţele la diferite niveluri OSI; | | 1. Definiţia reţelei de calculatoare, termeni uzuali, clasificare reţele. 2. Modele de referinţă ISO OSI şi TCP/IP. 3. Topologii de reţea. 4. Adresarea în rețele. IP adresarea TCP/IP. |
| ***Rețele de transport de date*** | | |
| – Să descrie principiile de funcţionare a reţelei de performanţă înaltă și caracteristici-lor principale lor: 10G/40G/100G Ethernet, FibreChannel și Infiniband.  – Să descrie principiile de funcţionare a reţelei magistrale.  – Să explice principiile de multiplexarea (MUX) și demultiplexarea (DEMUX).  – Să explice principiile de multiplexarea FDM, TDM, WDM, CWDM și DWDM.  - Să descrie principiile de funcţionare a reţelei PDH, SONET/SDH, DWDM și OTN. | 1. **Reţele de performanţă înaltă.**   Rețele 10G/40G/100G Ethernet, FibreChannel și Infiniband.   1. **Reţele magistrale.**   Rețele PDH, SONET/SDH, DWDM și OTN. | |
| ***Sisteme de operare pentru rețea*** | | |
| * Să determine prioritatea proceselor în UNIX * Să determine necesitatea procesoarelor de comenzi shell pentru sistemele de opeare. | **1. Procesoare de comenzi**   * Procesoare de comenzi shell:sh, csh, ksh, bash; prezentări generale. * Comenzi utile în context shell şi gestiunea exterioară a proceselor. | |
| * Să determine necesitatea nucleului în UNIX, alocarea spatiului fişierelor de sistem. Schema arbore. * Să studieze gestiunea, dispecerizarea şi programarea aplicaţiilor multitasking folosind procese sub Unix si sub Windows * Să stabilească schema de legare a fişierelor   - Să analizeze toate tipurile de fişiere şi formatul lor | **2. Structura sistemului Unix/Linux**  - Structura superioarăa sistemului de directoare Unix  - Sistemul de operare Unix: apeluri sistem, structuri interne.  - Fişiere şi procese sub Unix: schema de legare între ele.  - Sistemele de fişiere pentru sisteme de operare | |
| ***Proiectarea interfețelor de utilizator*** | | |
| * Să definească noţiunea de interacţiune om-maşină; * Să clasifice ciclul de viaţă al interfeţelor om-maşină; * Să descrie modelele din ingineria software, modelele ale interfeţei om-calculator; | **Ingineria interfeţelor om - maşină**  Domeniul interacţiunii om-maşină  Ciclul de viaţă a interfeţelor om-maşină  Abordări în proiectarea asistată a interfeţei | |
| * Să identifice setul de criterii ergonomice; * Să descrie criteriile ergonomice; * Să aplice criteriile ergonomice la elaborarea interfeţelor de utilizator | **Criterii ergonomice de proiectare a UI**  Importanţa criteriilor ergonomice  Ghiduri de reguli | |
| **II. Limbaje de programare** | | |
| **Limbajul de programare C** | | |
| * să definească tipurile de instrucțiuni ale limbajului C; * să aplice instrucțiunea if a limbajului C; * să aplice instrucțiunea switch a limbajului C; * să aplice operatorul ”?” * să definească operatorii logici ai limbajului C; * să explice conceptul de ”adevărat” și ”fals” în limbajul C. | 1. **Instrucțiunile limbajului C. Instrucțiuni de selecție.**   Grupurile de instrucțiuni ale standardului ANSI C. Operatorii logici ai limbajului C. Instrucțiunea *if.* Instrucțiunea *switch*. Operatorul ”?”. | |
| – să difinească recunoaşterea şi utilizarea tipurilor de date predefinite specifice unui limbaj de programare;  – să definească utilizarea unor tipuri de date proprii;  – să aplice operatorii predefiniţi elementari;   * – să aplice elementele sintactice specifice limbajului de programare studiat; | **1. Tabloul şi şirurile**  - Tipul tablou.  -Tipul şir de caractere. Operatori şi funcţii predefinite pentru: citirea, afişarea, concatenarea, cãutarea ,extragerea , inserarea şi eliminarea unor şiruri, pentru conversii (şir « valoare numericã)  - Tipul înregistrare. | |
| * să definească variabilele de tip pointer; * să explice modul de utilizare a operatorilor de adresare și dereferențiere și interacțiunea lor; * să aplice funcțiile de alocare dinamică a memoriei și eliberarea memoriei alocate dinamic; * să deducă formulele de accesare a elementelor tablourilor bidimensionale alocate dinamic. | 1. **Pointeri în limbajul C. Alocarea dinamică a memoriei.**   Variabile de tip pointer. Utilizarea operatorii de adresare şi dereferenţiere. Aritmetica pointerilor. Alocarea dinamică a adreselor. Eliberarea memoriei alocate dinamic. | |
| *Limbajul de programare C++* | | |
| * să definească apelul unor subprograme proprii cu înțelegerea mecanismelor de transfer prin intermediul parametrilor; * să identifice proprietățile unor structurii de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora; | **1. Subprograme definite de utilizator**  Funcții, declarare şi apel. Parametri formali şi parametri efectivi. Parametri transmişi prin valoare, parametri transmişi prin referinţã. Variabile globale şi variabile locale, domeniu de vizibilitate.Pproiectarea modularã a rezolvãrii unei probleme. | |
| * să difineasca noţiunea de clasă şi obiect. * să argumenteze accesul public şi privat într-o clasă. * să aplice clasele şi obectele la elaborarea programelor. | 2. Clase şi obecte  Date şi funcţii member ale clasei. Domeniul clasei. Operatorul rezoluţiei. Structura bibliotecelor. | |
| * să difinească noţiunea de constructori şi distructori. * să aplice încapsularea datelor la modelarea problemelor. * să explice alocarea dinamică a obectelor | 3. Încapsularea datelor  Constructori şi destructori. Constructori. Destructori. Alocarea dinamică a obectelor. | |
| *Grafica pe calculator* | | |
| * să implementeze pe calculator programe transformări 2D; * să aplice transformările geometrice bidimensionale in programare; * să elaboreze programe cu mişcări a unor figuri plane de-a lungul unei traectorii. | 1. Transformarea de vizualizare 2D   Construirea curbelor plane definite în formă parametrică, în coordonate carteziene, în coordonate polare şi în formă implicită f(x,y)=0. Transformări geometrice bidimensionale. Translaţia, scalarea, rotaţia faţă de un punct. Compunerea transformărilor. Simularea mişcării unei figuri plane de-a lungul unei traiectorii plane. | |
| * să implementeze pe calculator programe complexe de grafică tridimensională; * să implementeze pe calculator programe de animaţii bidimensionale şi tridimensionale; * să scrie proceduri şi funcţii pentru implementarea pe calculator a programelor de vizualizare a scenelor tridimensionale. | 1. Transformarea de vizualizare 3D   Translaţia, scalarea, rotaţia în jurul unei axe. Compunerea transformărilor. Vizualizarea scenelor 3D. Proiectarea paralelă şi perspectivă. Simularea mişcării unei figuri tridimensionale în spaţiu. Vizualizarea suprafeţelor definite în formă parametrică şi în coordonate carteziene. Algoritmi de determinare a vizibilităţii obiectelor la afişarea scenelor tridimensionale. Redarea luminii la afişarea scenelor tridimensionale.  Modele de culori. Redarea umbrelor la afişarea scenelor tridimensionale. | |
| *Visual C++* | | |
| * Să definească noțiunile de casete de dialog * Să distingă tipurile de casete modale * Capacitatea de a elabora programe în Visual C++ * Să definească funcțiile din Win32API   Exemple   * Să elaboreze o aplicație pentru Win32API care conține 3 butoane,1pictograma și mesaj | Biblioteca MFC  Casete de dialog modale şi  Nemodale,Butoane, etichete  Programarea Win32API  Funcția de afișare a mesajului  Funcția principală  Funcția procedură fereastră  Funcția de colorare a textului  Funcția de aliniere a textului pe fereastra Win32API | |
| *Programarea orientată pe obiecte (Java)* | | |
| * să definească noțiunile de clase, constructori, obiecte; * să definească variabilele statice, de instanță și locale; * să creeze metode supradefinite și supraîncărcate; * să analizeze utilizarea moștenirii în comparare cu utilizarea interfețelor în java. Exemple; * să creeze o interfață cu metodele proprii și clasele ce o va implementa; * să creeze o clasă abstractă și a claselor moștenitoare * să elaboreze clasele șabloane (generice). | 1. Clase și interfețe în limbajul Java   Noțiuni de clase, constructori, obiecte, metode statice și obișnuite, variabilele statice, de instanță ș locale. Supraîncărcarea și supradefinirea metodelor. Moștenirea. Utilizarea interfețelor. Utilizarea șabloanelor (claselor generici) | |
| * să definească noțiunea colecții; * să descrie utilizarea excepțiilor; * să elaboreze o listă (List) și să utilizeze metode specifice; * să elaboreze Map și să utilizeze metode specifice; * să utilizeze Set; * să descrie diferența dintre colecțiile de tip Tree și de tip Hash; * să elaboreze clase Comparator cu definirea modurilor de ordonarea obiectelor. | 1. Colecțiile și excepțiile în limbajul Java   Noțiuni de excepții, colecțiile. Utilizarea lor. Ordonarea obiectelor colecții. | |
| *Limbaje de programare Web* | | |
| * Să definească noțiunile de: formular, cadre. * Să argumenteze rolul utilizării formularelor la elaborarea unei pagini web; * Să elaboreze macheta unui site cu 4 cadre (2 verticale și 2 orizontale). | 1. Limbajul de marcare HTML   Noțiuni introductive  Formatarea textului  Inserarea Imaginilor  Inserarea legăturilor în documente HTML  Liste  Tabele  Ferestre în HTML  Formulare în HTML | |
| * Să definească noțiunea de CSS; * Să determine sintaxa unei reguli CSS; * Să argumenteze rolul regulilor în CSS. | 1. Limbajul de marcare CSS   Stiluri CSS  Clase CSS  Meniuri create cu CSS | |
| * Să definească noțiunile de: variabilă, constantă; * Să clasifice tipurile de date în PHP; * Să explice diferența dintre metoda GET și POST. | 1. Limbajul de programare Web PHP   Elemente de bază ale limbajului PHP  Instrucțiuni condiționate  Instrucțiuni repetitive  Funcții  Tipuri compuse  Lucru cu directoare și fișiere  Utilizarea datelor introduse în formatul XHTML  Generarea dinamică a conținutului | |
| * Să definească noțiunile de: browser Web, server Web. * Să cunoască mecanismul de transmitere a informațiilor intre client si server, folosind protocolul HTTP * Să înțeleagă aspectele-cheie ale dezvoltării unei aplicații web cu PHP | 1. Integrare PHP și MySQL   Funcționarea procesorului PHP aflat pe un server Web  Operații efectuate în cadrul unei tranzacții de informații date între browser și severul web, cu ajutorul unui server MySQL  Trimiterea datelor la o adresă de e-mail  Formular pentru înregistrarea utilizatorilor | |
| * Să fie familiarizați cu conceptul scripturilor client-Side; * Să înțeleagă conceptele generale ale limbajului JavaScript, conceptele şi protocoalele din spatele AJAX; * Să integreze cod JavaScript într-o aplicație web. | 1. Elemente JAVASCRIPT și AJAX   JAVASCRIPT  AJAX | |
| **III. Structuri de date şi algoritmi** | | |
| ***Baze de date și algoritmi*** | | |
| * să definească cazurile de utilizare a funcțiilor de grup și restricționarea rezultatelor în cadrul grupelor; * să determine modalitatea de utilizare a imbricării funcțiilor de grup. | 1. Agregarea datelor utilizând funcțiile de grup   Tipul funcțiilor de grup. Sintaxa funcțiilor de grup. Funcțiile de grup şi valorile nule. Clauza GROUP BY. Clauza GROUP BY cu mai multe coloane. Restricționarea rezultatelor grupelor. Clauza HAVING. Imbricarea funcțiilor de grup. | |
| * să definească modalitățile de legături dintre mai multe tabele. | 1. Afișarea datelor din mai multe tabele   Tipuri de legături. Clauza USING. Clauza ON. Produsul cartezian. | |
| - să definească argumentai de intrare a funcțiilor singulare şi generale;  - să determine regulile de utilizare a funcțiilor imbricate;  - să determine cazurile de utilizare a expresiei CASE și funcției DECODE. | 1. Funcții singulare   Tipurile funcții singulare. Funcții caracteriale. Funcții numerice. Lucrul cu datele. Funcții de tip date. Funcții de convertare. Elementele formatului de date. Funcții imbricate. Funcții generale. Expresia CASE. Funcția DECODE. | |
| *Structuri de date și metode de programare* | | |
| * să descrie construirea algoritmilor corespunzãtori unor prelucrãri elementare şi reprezentarea lor prin intermediul schemelor logice, programelor pseudocod. * să determine tipurile abstracte de date. * să argumenteze eficienţa structurilor de date. | **1. Structuri de date si tipuri de date abstracte**   * Structuri de date fundamentale * Clasificãri ale structurilor de date * Tipuri abstracte de date * Eficienta structurilor de date | |
| * să definească structurile şi funcţiile recursive * să utulizeze tipurile generice şi pointerii generici * să implimenteze operaţii cu structuri de date | **2. Structuri de date în limbajul C**   * Implementarea operatiilor cu structuri de date. * Utilizarea de tipuri generice. * Utilizarea de pointeri generic. * Structuri si functii recursive. | |
| * să identifice vectorii ordonaţi, alocaţi dinamic.multidimensionali şi de biţi * să stabilească legăturile între vectori * să creeze aplicaţii cu componente conexe | **3. Vectori**   * Vectori. * Vectori ordonati. * Vectori alocati dinamic. * Aplicatie: Componente conexe. * Vectori multidimensionali. * Vectori de biti. | |
| * să descrie listele cu legături * să stabileasca legaturile dintre liste * să compară colecţiile de liste * să propună probleme cu liste de legături * să optimizeze listele de legături | **4. Liste cu legături**   * Liste înlãntuite. * Colectii de liste. * Liste înlãntuite ordonate. * Variante de liste înlãntuite. * Liste dublu-înlãntuite. * Comparatie între vectori si liste. * Combinatii de liste si vectori. * Tipul abstract listã (secventã). * Liste Skip. * Liste neliniare. | |
| * să descrie tipurile abstracte * să aplice tipul abstract”Mulţime” * să aplice acoperirea optimă cu mulţimi * să implimenteze dicţionare prin tabel de dispersie   să dezvolte compresia LZW | **5. Mulţimi şi dicţionare**   * Tipul abstract “Multime”. * Aplicatie: Acoperire optimã cu multimi. * Tipul “Colectie de multimi disjuncte”. * Tipul abstract “Dictionar”. * Implementare dictionar prin tabel de dispersie. * Aplicatie: Compresia LZW. | |
| * să identifice listele stivă şi listele coadă * să utilizeze tipul „Coadă cu priorităţi” * să realizeze evaluarea expresiei * să utilizeze vectorii heap * să utilizeze elimenarea recursivităţii folosind o stivă | **6. Stive şi cozi**   * Liste stivã. * Aplicatie: Evaluare expresii. * Eliminarea recursivitãtii folosind o stivã. * Liste coadã. * Tipul “Coadã cu prioritãti”. * Vectori heap. | |
| * să difineasca arborii * să recunoască structurile arboriscente * să utilizeze arborii binari, Huffman şi multicãi. * să propună alte structuri de arbori * să descrie arborii de căutare * să compare arborii binari de căutare şi echilibraţi * să compare arborii Splay si Treap, RB şi AA * să aplice arborii de căutare * să recomande arborii de căutare pentru diferite probleme | **7. Arbori**   * Structuri arborescente. * Arbori binari neordonati. * Traversarea arborilor binari. * Arbori binari pentru expresii. * Arbori Huffman. * Arbori multicãi. * Alte structuri de arbore. * Arbori binari de cãutare. * Arbori binari echilibrati. * Arbori Splay si Treap. * Arbori AVL. * Arbori RB si AA. * Arbori 2-3. | |
| * să difenească structurile de graf * să identifice metode de explorare a grafurilor * să stabilească legăturile grafurilor ca structuri de date * să utilizeze structurile de graf in anumite probleme * să argumenteze arborii de acoperire de cost minim * să formuleze grafurile virtuale | **8. Structuri de graf**   * Grafuri ca structuri de date. * Reprezentarea grafurilor prin alte structuri. * Metode de explorare a grafurilor. * Sortare topologicã. * Aplicatii ale explorãrii în adâncime. * Drumuri minime în grafuri. * Arbori de acoperire de cost minim. * Grafuri virtuale. | |
| * să definească structurile de date externe * să interpreteze specificul datelor pe suport extern * să utilizeze indexarea datelor * să formuleze arborii B | **9. Structuri de date externe**   * Specificul datelor pe suport extern. * Sortare externã. * Indexarea datelor. * Arbori B. | |
| * să difinească structurile de date în C++ * să interpreteze clasele şi obiectele în C++ * să se servească de clase şablon în C++ * să utilizeze clase container din biblioteca STL * să utilizeze clasele STL în aplicaţii * să dezvolte noi clase container | **10. Structuri de date in limbajul c++**   * Avantajele utilizãrii limbajului C++. * Clase si obiecte în C++. * Clase şablon (“template”) în C++. * Clase container din biblioteca STL. * Utilizarea claselor STL în aplicatii. * Definirea de noi clase container. | |
| *Proiectarea bazelor de date* | | |
| * să stabilească modalităţile de efectuare a tuturor tipurilor de proiectare; * să identifice redundanța datelor; * să clasifice tipurile de anomalii; * să efectuieze controlul integrităţii legăturilor; * să determine domeniile de aplicabilitate ale limbajelor de interogare; * să stabilească rolul restricţiilor în aplanarea anomaliilor; | 1. **Problemele proiectării.** Proiectarea conceptuală, logică, fizică. Tipuri de redundanță de date. Integritatea datelor. Anomaliile. | |
| * să argumenteze utilizarea operațiilor algebrice; * să clasifice operatorii algebrici relaționali; * să stabilească modalităţi şi principii de utilizare pentru micșorarea redundanței; * să distingă metodele de eliminare e repetărilor; * să identifice rolul operatorilor în procesul de proiectare a bazei de date; | 1. **Algebră relațională**. Operatorii din teoria mulțimilor (intersecția, reuniunea, diferența, produsul cartezian). Operatori relaționali (proiecția, selecția, joncțiunea, pseudojoncțiunea) | |
| * să definească normalizarea; * să identifice normele normale; * să efectuieze trecerea de la o formă normală la alta de rang mai superior; * să identifice principiul descompunerii; * să aplice proprietatea joncțiunii fără pierdere. | 1. **Metoda formelor normale.**  Principiul normalizării. Formele normale ( FN1, FN2, FN3, FNBC, FN4, FN5). Supercheie. Descompunerea. Proprietatea joncțiunii fără pierdere. | |
| *Prelucrarea datelor fizice și metode numerice* | | |
| * să identifice probleme din fizică ale căror rezolvare se reduc la rezolvarea unei ecuaţii neliniare; * să clasifice metodele de rezolvare a ecuaţiilor neliniare în dependenţă de problemele abordate; * să demonstreze relaţiile care duc la rezolvarea ecuaţiilor neliniare pentru metoda Newton; * să analizeze eroarea metodei Newton. | 1. **Rezolvarea numerică a ecuaţiei neliniare si sistemelor de ecuatii neliniare prin metoda Newton.**   Relațiile metodei Newton pentru rezolvarea ecuației neliniare. Relațiile metodei Newton pentru rezolvarea sistemelor de ecuatii neliniare. | |
| * să identifice probleme din fizică ale căror rezolvare se reduce la rezolvarea ecuaţiilor diferenţiale de tip problemă Cauchy; * să deducă relaţiile metodelor Euler, Euler modificată pentru rezolvarea ecuației diferentiale ordinare; * să analizeze eroarea metodelor Euler, Euler modificată. | 1. **Rezolvarea numerică a problemei pentru ecuaţiile diferenţiale ordinare**   Relațiile metodei Euler pentru rezolvarea ecuației diferentiale ordinare de ordinul1. Relațiile metodei Euler modificată pentru rezolvarea ecuației diferentiale ordinare de ordinul1. Estimarea erorii metodelor. | |
| * să identifice probleme din fizică ale căror rezolvare se reduce la interpolare/extrapolare numerică. * să identifice aplicabilitatea polinomului de interpolarea Lagrange pentru setul concret de date; * să deducă relaţiile pentru polinomul Lagrange; * să identifice aplicabilitatea polinomului de interpolarea Newton de speța întîi și speța a doua pentru setul concret de date; * să deducă relaţiile pentru polinomul Newton de speța întîi și speța a doua; * să determine polinomul Newton de speța întîi și speța a doua pentru setul dat; * să deducă diferențele finite de orice ordin; | 1. **Interpolarea numerică**. stabilirea condițiilor de interpolare. Interpolarea datelor prin polinomul Lagrange. Construirea polinomului Lagrange. Interpolarea datelor cu polinomul Newton de speța întîi și speța a doua. Diferențele finite. Estimarea gradului polinomului. | |
| * să identifice probleme din fizică ale căror rezolvare se reduce la aproximare numerică. * să identifice tipul funcției de aproximare numerică. * să deducă relaţiile pentru metoda pătratelor minime * să aplice metoda pătratelor minime la prelucrarea datelor experimentale/observații concrete; | 1. **Metoda pătratelor minime**. Aproximarea datelor experimentale prin metoda pătratelor minime. Estimarea condiției de abatere minimă. Construirea polinomului de interpolare cu grad diferit. | |
| * să identifice probleme din fizică ale căror rezolvare se reduce la calculul numeric al integralelor; * să deducă relaţiile pentru formula de cuadratură a dreptunghiurilor; * să deducă relaţiile pentru formula de cuadratură Simpson ; * să analizeze eroarea formulelor de cuadratură a dreptunghiurilor si Simpson; | 1. **Calculul numeric al integralelor definite.**   Formula de cuadratură a dreptunghiurilor. Formula de cuadratură Simpson. Eroarea formulelor de cuadratură a dreptunghiurilor și Simpson. | |
| * să identifice probleme din fizică ale căror rezolvare se reduce la minimizarea funcţiilor; * să deducă relaţiile pentru minimizarea a functiilor multivariabile prin metoda gradientului; * sa formuleaza problema minimizarii neconditionata a funcţiilor multivariabile; * să deducă relaţiile pentru minimizarea a functiilor multivariabile prin metoda coborarii cea mai rapida | 1. **Minimizarea neconditionata a funcţiilor multivariabile. Metoda gradientului.**   Minimizarea functiilor multivariabile in de-a lungul gradientului. Minimizarea functiilor multivariabile in de-a lungul gradientului: metoda coborarii cea mai rapida. | |
| ***SQL*** | | |
| * să determine cazurile de utilizare a comenzilor de manipulare cu datele tabelelor; * să determine modul de utilizare a comenzilor de fixare şi ștergere a datelor. | 1. **Limbajul de Manipulare cu datele (DML)**   Comanda INSERT. Comanda UPDATE. Comanda DELETE. Comanda TRUNCATE. Tranzacțiile bazei de date. | |
| * să determine obiectele unei baze de date; * să definească şi să utilizeze obiectele bazei de date. | 1. **Alte obiecte ale bazei de date**   Obiectele bazei de date. Reprezentări. Secvenţe. Indexii. Sinonime. | |
| * să definească subcereri cu mai multe coloane, subcereri scalare şi corelate; * să determine ariile de utilizare a operatorilor EXISTS, NOT EXISTS. | 1. **Subcereri complexe**   Subcereri cu mai multe coloane. Subcereri scalare şi corelate. Operatorii: EXISTS, NOT EXISTS. Clauza WITH. | |
| ***Limbaje procedurale pentru BD*** | | |
| * să definească variabilele, identificatori, delimitatori; * să determine reguli pentru declararea și inițializarea variabilelor PL/SQL; * să identifice tipurile de date. | **1. Declararea variabilelor PL/SQL**  Noțiuni de variabile, identificatori şi delimitatori. Noțiuni de tipuri de date. Crearea variabilelor şi tipurilor de date scalare. Definirea variabilelor BIND şi de substituție. Analiza diferenței dintre tipurile de date scalare şi atributul %TYPE, variabile BOOLEAN, variabile BIND, variabile de substituţie. | |
| * să definească funcții SQL în PL/SQL; * să determine structura blocurilor imbricate; * să identifice metodele de convertare a tipurilor de date. | 1. **Crearea structurilor executabile**   Sintaxa blocului PL/SQL și reguli de utilizare. Comentarea codului de programă. Sintaxa structurii blocurilor imbricate. Analiza metodelor de convertare a tipurilor de date. Crearea operatorilor în PL/SQL. | |
| * să definească instrucțiuni și expresii PL/SQL; * să determine tipurile de instrucțiuni IF; * să identifice valorile NULL în instrucțiunile IF. | 1. **Construcții de control**   Tipurile de instrucțiuni IF. Expresia CASE. Valorile NULL în instrucțiunile IF. Manipularea valorile nule. Instrucțiunea LOOP. Bucla WHILE. | |
| ***BD Web orientate în Oracle*** | | |
| - să definească proceduri;  - să determine structura procedurilor;  - să identifice parametrii de intrare/ieșire a procedurilor. | **1. Proceduri**  Sintaxa de creare a procedurilor. Crearea, ștergerea și modificarea procedurilor. Executarea procedurilor | |
| * să definească funcții; * să determine structura funcțiilor; * să identifice parametrii de intrare/ieșire a funcțiilor. | **2. Funcții**  Sintaxa de creare a funcțiilor. Crearea, modificarea și ștergerea funcțiilor. Executarea funcțiilor. Definirea parametrilor funcțiilor. | |
| * să definească specificația și corpul pachetului; * să determine componentele unui pachet; * să identifice componentele pachetului în dicționarul bazei de date. | **3.Crearea pachetelor**  Componentele pachetului PL/SQL. Sintaxa de creare a specificației și corpului pachetelor. Developarea pachetelor PL/SQL. Sintaxa de ștergere a pachetelor. Vizualizarea pachetelor în dicționarul de date. Avantajele utilizării pachetelor PL/SQL. | |
| IV. | | |
| *Principiile testării Software* | | |
| * să definească testarea; * să descrie modelurile de Ciclu de viața al produsului software (cu avantajele și dezavantajele); * să descrie rolurile în echipe de elaborare produselor software (obligațiunile); * să definească nivelurile de testare (unit, integration, system, user acceptance); * să definească tipurile de testare (funțională și non-funcțională); * să descrie ciclul de viața defectelor (cu toate atributele necesare); * să elaboreze test case-urile bazându-se pe cerințele funcționale. | Noțiuni de testare, QA.  Descrierea detaliată modelellor Cascadă, V-model, Spirală și Agile.  Structura echipei, Project manager, Business analyst, System Architect, QA lead, QA engineer  Nivelurile de testare cu detalii (testarea modulelor aparte – unit testing, 3 tupuri de integration testing, system și user acceptance testing)  Analiza tipurilor de testare – funcțională (inclusiv regresivă) și non-funcțională (performance, load, stress, usability, portability)  Atributele și statusuri de viața ai defectului.  Elaborarea test case-urilor | |
| *Testarea software automatizată* | | |
| * să definească testarea automatizată, serviciile web, Page Object Model, HTTP; * să descrie modurile de prelucreare a datelor în formatele XML și JSON (analiza de sintaza, construirea arborilor); * să aplice tehnice de serializare și deserializare datelor în/din formate binar/XML/JSON; * să aplice metode din biblioteca Selenium pentru automatizarea testării interfețelor de utilizator; * să aplice tehnice de construirea frameworkurilor cu bibliotecile TestNG, mecanismurilor de logare și Maven; * să elaboreze test case-urile pentru testarea serviciilor REST (cu JSON) | Componente și structura protocolului HTTP. Noțiunile de POM, web serviciile.  Prelucrarea datelor in XML și JSON.  Serializarea și deserializarea datelor și rolurile lor în transmitearea datelor.  Elaborarea frameworkului pentru testarea web serviciilor și/sau interfeței de utilizator.  Aplicarea metodelor din biblioteca Selenium. | |
| *Sisteme de inteligență artificială* | | |
| 1. Să definească noțiunea de sistem expert; 2. Să descrie structura unui sistem expert; 3. Să determine funcționalitatea sistemului expert. | **Sistem expert**  Sisteme expert – generalităţi  Structura generală a unui Sistem Expert  Funcţionalitatea unui Sistem expert | |
| 1. Să definească noțiunile de: Sistem inteligent de suport decizional (SISD), decizie, proces decizional, sisteme suport pentru decizii (SSD); 2. Să clasifice tipurile de decizie, SSD; 3. Să aplice tehnicile de optimizare a deciziilor la rezolvarea problemelor. | **Sisteme inteligente de suport decizional**  Definiții, concepte  Arhitectura sistemelor suport pentru decizii  Metode de dezvoltare a sistemelor suport pentru decizii | |
| 1. Să definească noțiunile de: logică fuzzy, mulțimi fuzzy, sisteme cu logică fuzzy, decizie fuzzy; 2. Să clasifice tipurile de mulțimi fuzzy; 3. Să aplice logica fuzzy la rezolvarea problemelor. | **Sisteme inteligente cu logică fuzzy**  Definiții, concepte  Mulțimi fuzzy  Sisteme inteligente cu logică fuzzy  Rezolvarea problemelor | |

1. **Bibliografia recomandată**
2. Ș. Raicu, E. Roșca, Ș. Burciu. Teoria informației și coduri. Editura AGIR, București, 2012. 171p.
3. D Ionescu. Codificare şi coduri, Ed. Tehnică, Bucureşti, 1981.
4. A.T. Murgan. Principiile teoriei informaţiei în ingineria informaţiei şi a comunicaţiilor Editura Academiei Române, Bucureşti, 1998.
5. E Roşca. Teoria informaţiei şi coduri. Culegere de probleme, Ed. Fair Partners, Bucureşti, 2000.
6. A.Spătaru. Teoria transmisiunii informaţiei, Editura Didactică şi Pedagogică, București 1983.
7. C. Târcolea, C.Câşlaru, E. Roşca. Statistică matematică, teoria informaţiei şi coduri, Ed. Fair Partners, Bucureşti, 2003.
8. Horea Ors. Arhitectura sistemelor de calcul, suport de curs. Universitatea din Oradea - Facultatea Științe. 147 p.
9. Groza B. Introducere în criptografie. Timișoara 2012.
10. MIHAI I. Securitatea informațiilor, Editura Sitech, 2012
11. MIHAI I. Securitatea sistemului informatic, Editura Dunărea de Jos, 2007
12. Patriciu V., Pietrosanu M., Bica I., Priescu J. Semnături electronice și securitate informatică, Editura All, 2006.
13. M.Beldiga, Gh. Căpățînă. Sisteme suport pentru decizii-îndrumar de laborator. Chișinău, 2016
14. D. Rotaru. Inteligență artificială - note de curs. Bacău 2007
15. Dzitac, Inteligența artificială, Editura Universității Aurel Vlaicu, 2008.
16. C.Ileană, Agenți inteligenți, Babel, 2002
17. **Herbert Shildt**. *C++ Manual complet*. Editura Teora, București 1997
18. **Andrew S. Tanenbaum.** *Rețele de calculatoare. Ediția 3-a.* Editura Agora,București 1998
19. *Intel Architecture Sofware Developer's Manual Vol 1, 2A, 2B, 3,* Intel Corporation, Santa Clara, California, 2003.
20. *INTEL Microprocessors, Vol. 1, 2,* Intel Corporation, Santa Clara, California, 1991;
21. **D. Somnea, T.Vladuţ,** *Programarea în Assembler*, Editura tehnică, Bucureşti, 1992;
22. **Spânulescu, S.I.Spânulescu,** *Prelucrarea, înregistrarea, transmiterea şi afişarea datelor*, Universitatea din Bucureşti, 1984;
23. **Spânulescu, S.I.Spânulescu**, *Circuite integrate digitale şi sisteme cu microprocesoare*, Editura Victor, Bucureşti, 1996;
24. **Л.И.Турчак**. *Основы численных методов*. М.: Наука. 1987.
25. **Н.С.Бахвалов, Н.П.Жидков, Г.М.Кобельков**. *Численные методы*., М.: Наука. 1987.
26. **Bodescu Gh**. *Metode numerice în algebră*. Bucureşti, 1979.
27. **Н.Н.Калиткин**. *Численные методы*. М.: Наука. 1978.
28. **F. Ionescu,** *Grafica in realitatea virtuala*, Ed. Tehnica 2000
29. **R. Baciu, D. Volovici,** „Sisteme de prelucrare grafica”, Ed. Albastra 1999
30. **Bogdan Dumitrescu** *Carte Algoritmi de Calcul Paralel,* București 2001
31. **M. Popescu** - *Baze de date*, Editura Renaissance, Bucureşti, 2010 2.
32. **M. Popescu** - *Baze de date relaţionale*, Editura Academiei Tehnice Militare, Bucureşti, 2001
33. **M. Tilca, R. Boriga** - *Baze de date*, Ed. Univ. Titu Maiorescu, 2007
34. **M. Fotache** - *Proiectarea bazelor de date. Normalizare şi postnormalizare. Implementări SQL şi Oracle*, Ed. Polirom, 2005
35. **Ş. Ardeleanu** - *Transact SQL*, Ed. Niculescu, 2004