**TEME PROIECTE "ELEMENTE AVANSATE DE PROGRAMARE"**

**1. (1-2 studenti)** Să se realizeze o aplicație Java care să asigure gestiunea unei biblioteci universitare cu cărţi în diferite domenii (literatură, informatică, fizică, matematică, mecanică etc.). Aplicația trebuie să permită efectuarea următoarele operații:

* adăugarea/actualizarea/ștergerea unei cărți;
* adăugarea/actualizarea/ștergerea unui cititor;
* căutarea unei cărți după titlu sau autor;
* căutarea unui cititor după CNP sau nume;
* împrumutarea/returnarea unei cărți unui cititor;
* afișarea unei liste care să conțină cărțile nereturnate la timp.

Aplicația va avea o interfață grafică și va stoca informațiile despre cărți și cititori într-o bază de date.

**2**. **(1-2 studenti; pentru realizarea unui daemon ce va efectua anumite operatii in mod automat se poate lucra in echipa pana la 4 studenti)** Să se realizeze o aplicație Java care să asigure gestiunea unor adrese de mail dintr-o firmă. Firma este structurată pe departamente, iar în fiecare departament adresele vor conţine următoarele câmpuri: nume, prenume, adr\_mail, parola, domeniu, funcţie, tip\_cont, activa, ultima\_accesare. Aplicația trebuie să permită efectuarea următoarelor operații:

* Încărcare din fişier a adreselor de mail ale firmei, din toate departamentele.
* Adăugare adresă mail pentru un nou angajat.
* Inactivare adrese nefolosite de mai mult de 3 luni.
* Ştergere adrese mail inactive.
* Sortare adrese mail după nume şi prenume angajat de la un anumit departament.
* Modificare parolă la o anumită adresă de mail.
* Adăugare departament nou.
* Salvare adrese de mail în fişier, pe departamente.

Aplicația va avea o interfață grafică și va stoca informațiile despre cărți și cititori într-o bază de date.

**3. (1-3 studenti)** Să se realizeze o aplicație Java care să permită generarea unor fișiere text cu numere aleatoare. Aplicația trebuie să aibă o interfață grafică și să permită efectuarea următoarele operații:

* selectarea unui generator de numere pseudo-aleatoare dintr-o listă care conține cel puțin 3 astfel de generatoare;
* inițializarea parametrilor generatorului selectat;
* stabilirea intervalului de valori pentru numerele pseudo-aleatoare generate, precum și numărul lor;
* afișarea numerelor aleatoare în interfața grafică folosind un control de tip text;
* salvarea numerelor aleatoare generate într-un fișier text.

**4. (1-3 studenti)** Să se realizeze o aplicație Java care să permită generarea unor fișiere text cu conținut aleator. Aplicația trebuie să aibă o interfață grafică și să permită efectuarea următoarele operații:

* selectarea metodei de generare:
  + cuvinte selectate aleator dintr-un anumit fișier text, caz în care utilizatorul trebuie să poată selecta un anumit fișier text, care va fi afișat în interfața grafică folosind un control de tip text;
  + caractere aleatoare, caz în care utilizatorul trebuie să poată stabili mulțimea caracterelor pe care dorește să le folosească, folosind un șir de caractere.
* stabilirea numărului de cuvinte/caractere pe care trebuie să le conțină fișierul;
* opțional, utilizatorul trebui să poată stabili numărul de linii din fișier, lungimea minimă/maximă a unui cuvânt și lungimea minimă/maximă a unui cuvânt;
* afișarea textului aleatoriu generat în interfața grafică folosind un control de tip text;
* salvarea textului aleator generat într-un fișier text.

**5. (1-4 studenti)** Să se realizeze o aplicație Java care să permită criptarea unui fișier text folosind un cifru Vernam. Aplicația trebuie să aibă o interfață grafică și să permită efectuarea următoarele operații:

* selectarea unui generator de numere pseudo-aleatoare, care va fi folosit pentru generarea cheii de criptare, dintr-o listă care conține cel puțin 3 astfel de generatoare;
* inițializarea parametrilor generatorului selectat și salvarea lor într-un fișier text (aceștia reprezintă, de fapt, cheia secretă);
* selectare fișierului text care trebuie criptat și afișarea acestuia în interfața grafică folosind un control de tip text;
* criptarea fișierului selectat folosind un cifru Vernam;
* afișarea textului criptat în interfața grafică folosind un control de tip text;
* salvarea textului criptat într-un fișier text.

**6. (1-3 studenti)** Să se realizeze o aplicație Java care să folosească biblioteca choco (http://choco-solver.org/) pentru a genera permutări, aranjamente sau combinări. Aplicația trebuie să aibă o interfață grafică și să permită efectuarea următoarele operații:

* selectarea unui tip de generare dintre cele 3 enumerate mai sus;
* inițializarea valorilor necesare pentru tipul respectiv de generare;
* afișarea rezultatului generării selectate în interfața grafică, folosind un control de tip text;
* salvarea rezultatului generării selectate într-un fișier text.

**7.** **(1-2 studenti)** Să se realizeze o aplicație Java care să asigure gestiunea unui magazin de componente PC (procesoare, plăci de bază, memorii, hdd-uri etc.). Fiecare componentă conţine următoarele câmpuri: cod\_produs, preţ, marca, an\_fabricaţie, garanţie, producător, ţară\_provenienţă, stoc (0 – dacă nu se află în stoc, >0 – câte sunt în stoc).

Aplicația trebuie să permită efectuarea următoarele operații:

* Încărcare din fişier a componentelor din magazin.
* Adăugare produs nou la o anumită componentă introdusă de la tastatură.
* Vânzare produs (stoc = stoc - 1).
* Ştergere anumit produs după cod\_produs.
* Încărcare stoc (refacerea stocului la un anumit produs, căutat în prealabil după cod).
* Să se calculeze totalul în lei pentru o anumită categorie (suma produselor stoc\*preţ de la o anumită componentă).
* Afişare preţ pentru anumite produse căutate după marcă.

Aplicația va avea o interfață grafică și va stoca informațiile despre cărți și cititori într-o bază de date.

**8.** **(1-3 studenti\*) Game of life** („Jocul vieții”) presupune simularea evoluției unor celule într-un spațiu bidimensional. Jocul se desfășoară într-un grid de dimensiuni **N** și **M**, având o configurație inițială de celule de la care pornește simularea. Evoluția are loc parcursul a **K** etape, starea la pasul **p** depinzând exclusiv de starea la pasul **p - 1**.

Multiplicarea celulelor urmează câteva reguli:

* orice celulă cu mai puțin de două celule vecine moare de singurătate
* orice celulă cu mai mult de trei celule vecine moare din cauza supraaglomerării
* orice celulă cu două sau trei celule vecine supraviețuiește
* o celulă nouă este creată dacă are exact trei celule vecine

Spațiul simulării poate fi reprezentat fie ca un **plan**, fie ca un **toroid**. În cazul reprezentării **planare**, celulele marginale au 5, respectiv 3 vecini. În cazul reprezentării **toroidale**, toate celulele au 8 vecini, harta putând fi parcursă circular.

Sa se realizeze o aplicatie Java cu interfata grafica ce va primi configuratia initiala a gridului si va scrie atat intr-un fisier text cat si intr-un controller de tip text configuratia dupa K etape (oferite tot ca input).

**9.** **(1-3 studenti; daca aplicatia va avea o componenta de joc online, atunci se poate extinde echipa spre maxim 5 studenti, in functie de modulele adaugate)** Sa se realizeze o aplicatie Java cu interfata grafica ce va simula jocul de sah intre:

\* AI vs AI

\* Human vs AI

\* Human vs Human

Datele statistice precum si profilele jucatorilor vor fi retinute intr-o baza de date.

**10.** **(1-2 studenti)** Sa se realizeze o aplicatie Java cu interfata grafica pentru sustinerea de teste grila. Intrebarile si variantele de raspuns pot contine atat text cat si fisiere foto/video/audio si sunt incarcate dintr-o baza de date. Utilizatorii pot sustine teste doar daca se inregistreaza in aplicatie, datele acestora fiind stocate in baza de date, avand ulterior posibilitatea editarii profilului personal si a unor informatii din cadrul acestuia, incarcarea unui avatar etc.

**11. (1-2 studenti\*)** Sa se realizeze o aplicatie Java cu interfata grafica pentru jocul de Puzzle. Fereastra de joc va contine un grid NxN cu numere de la 1...(NxN-1) dispuse aleator si o celula goala, singura operatie permisa fiind interschimbarea dintre celula goala si una vecina (doua celule sunt vecine daca au o muchie comuna). Operatia poate fi realizata cu ajutorul mouse-ului sau a sagetilor de pe tastatura. Totodata meniul principal va oferi posibilitatea incarcarii de pe disc a unei poze de dimensiune MxM ce va fi taiata automat de catre aplicatie iar partile componente (mai putin cea din coltul din dreapta jos) vor fi amestecate in celulele gridului.

**12. (1-4 studenti\*)** Simulator editor grafic. Se va implementa un simulator de editor grafic care va permite următoarele operaţii:

- trasare linie dreapta (poziţie început, poziţie sfârşit);

- trasare triunghi (având cele 3 colturi);

- trasare cerc (având centrul si raza);

- trasare patrulater (având cele 4 colturi);

- trasare poligon (având n colturi);

- afişarea imaginii

- adaugarea de text

- desenarea figure pline

- „drag and drop”, „resize” si „rotate” pentru fiecare figura desenata. Se va considera disponibil un panel de 800x600 pixeli.

Aplicaţia va trebui sa permită serializarea figurile geometrice desenate si salvarea acestora pe disc sub forma de .jpg (sau orice alt format) si/sau pdf.

**13. (1-4 studenti\*)** Editor de diagrame pentru baze de date. Se va realiza o aplicatie Java care va permite crearea in mod vizual de diagrame pentru o baza de date, cu entitati si relatii, nume de tabele si atribute. Aplicatia va permite salvarea diagramei pe disc precum si generarea unui fisier .sql ce va contine codul de creare a tabelelor (inclusiv constrangeri).

**14. (1 student)** Joc Minesweeper. Aplicatia va permite salvarea jocului curent si reluarea ulterioara a acestuia si va avea mai multe nivele de dificultate. Se impune folosirea unei baze de date.

**15. (1 student)** Joc Solitaire. Aplicatia va permite salvarea jocului curent si reluarea ulterioara a acestuia si va avea mai multe nivele de dificultate. Se impune folosirea unei baze de date.

**16. (1 student)** Joc Spanzuratoarea. Aplicatia va permite salvarea jocului curent si reluarea ulterioara a acestuia si va avea mai multe nivele de dificultate. Se impune folosirea unei baze de date.

**17. (1 student)** Joc Blackjack. Aplicatia va permite salvarea jocului curent si reluarea ulterioara a acestuia si va avea mai multe nivele de dificultate. Se impune folosirea unei baze de date.

**18. (1 student)** Joc Connect4. Aplicatia va permite salvarea jocului curent si reluarea ulterioara a acestuia si va avea mai multe nivele de dificultate. Se impune folosirea unei baze de date.

**19. (1-2 studenti\*)** Joc Sudoku. Aplicatia va permite salvarea jocului curent si reluarea ulterioara a acestuia si va avea mai multe nivele de dificultate. Se impune folosirea unei baze de date.

**20. (1-5 studenti; orice alt modul poate adauga studenti in plus)** Joc Ping-Pong. Aplicatia va permite atat jocul AI vs Human cat si Human vs Human. Ca mod de joc vor exista optiunile:

- Single Game

- Knock-out Tournament (single Human player vs AI’s)

- Knock-out Tournament (only Human players)

- Diferite nivele de dificultate

Se impune folosirea unei baze de date. Utilizarea jocului online este optionala.

**21. (1-3 studenti)** Aplicatie de chat (1 server cu n clienti) cu o camera publica si camere private. Se vor utiliza metode input atat text cat si emoji, se pot incarca imagini si profilele utilizatorilor se pot personaliza. Se vor implementa roluri suplimentare pentru administratori.

**22. (1-4 studenti)** Aplicatie pentru gestionarea taskurilor din realizarea unui proiect. Se vor realiza functii pentru urmatoarele operatii:

- crearea unui proiect nou de catre project manager

- inrolarea membrilor echipei

- deschiderea de taskuri si atribuirea lor catre membrii echipei

- contorizarea timpului si alerte pentru apropierea si depasirea deadline-urilor

- posibilitatea updatarii statusurilor taskurilor

- posibilitatea adaugarii de notite si incarcarii de fisiere

- posibilitatea evaluarii membrilor echipei de catre project manager

Aplicatia va comunica in mod constant cu o baza de date.