**ATENTIE! Prima conditie pentru nota 5 este ca aplicatia trebuie obligatoriu sa compileze si sa porneasca!!!**

**Aplicatie ponturi pariuri sportive**

Se va implementa o aplicatie in care pentru diverse echipe sportive se vor memora vectori de forma (victorii, egaluri, infrangeri). La input-ul unui meci intre doua echipe, se va oferi un pont referitor la rezultatele posibile (victorie acasa, egal, victorie deplasare, etc.).

**Joc “Rock-paper-scissors”**

Implementati un joc Rock-paper-scissors astfel incat 2 jucatori sa poata juca simultan folosind aceeasi tastatura. Jocul va afisa un counter 3-2-1, din secunda in secunda. Dupa ce se afiseaza 1, jucatorii vor trebui sa aleaga una dintre cele 3 arme folosind butoane ale tastaturii. Jucatorul care alege primul va declansa un timer de 100ms, timp in care al doilea jucator va trebui sa aleaga si el o arma, altfel va pierde runda. Castiga primul jucator care invinge in 5 runde.

**Aplicatie meteo**

Construiti o aplicatie cu interfata grafica in care puteti vedea informatii despre starea vremii in Constanta, Bucuresti, Iasi, Cluj si Timisoara. Informatiile trebuie sa poata fi actualizate automat o data la 3 minute sau instant prin apasarea unui buton. Folositi API-ul openweathermap.

**Joc “Create your own adventure”**

Implementati un joc de tip “create your own adventure”. Povestea trebuie sa aiba cel putin 15 etape. Jucatorul trebuie sa poata salva progresul in orice etapa. Jucatorul trebuie sa poata incarca un joc salvat.

**Aplicatie “Learn Java”**

Creati o aplicatie tip e-learning, cu interfata grafica. In partea stanga se va afisa lectia, iar in partea dreapta studentul va putea scrie codul programului. Programul va verifica daca studentul a scris corect si va executa codul studentului. Aplicatia va avea doua lectii: “Hello, World!” si “Hello, Mike!”. Completarea primei lectii va afisa a doua lectie.

**Joc “Flappy bird”**

Implementati un joc de tip “Flappy bird” cu interfata grafica. Se poate juca prin click pe mouse sau apasand tasta Space de pe tastatura. Se vor salva cele mai bune 10 scoruri.

**Simulare Liga I**

Consideram un campionat de fotbal cu 8 echipe. Echipele trebuie sa poata fi introduse de la tastatura. Campionatul se joaca in sistem tur-retur, fiecare cu fiecare. Se va afisa cate un meci si se va introduce scorul pentru acesta. Clasamentul trebuie sa fie vizibil permanent pe ecran. Dupa adaugarea unui scor se va actualiza automat clasamentul. Clasamentul trebuie sa contina locul, numele echipei, numarul de meciuri, numarul de victorii, numarul de egaluri, numarul de infrangeri, numarul de goluri inscrise, numarul de goluri primite si numarul de puncte. Clasamentul va fi ordonat de la cea mai buna la cea mai slaba echipa.

**Magazie fabrica**

Scrieti o implementare a problemei producator-consumator care sa simuleze magazia unei fabrici. Implementati si o paradigma observator care sa anunte intrarea in magazie a unor noi piese catre departamentele interesate.

**Joc “Texas Hold ‘Em”**

Se va defini setul de carti de joc.

Trebuie sa se poata introduce numele a 2 jucatori, fiecare incepand jocul cu 100 de fise si cu 2 carti extrase aleator din pachet. Se vor afisa cartile si fisele.

In runda de flop se vor extrage aleator 3 carti din setul ramas si se vor afisa. Va urma o runda de pariuri. Dupa runda de pariuri, daca este cazul va urma runda de turn, in care se mai extrage aleator o carte din pachet, urmata de o noua runda de pariuri. Daca este cazul, ultima este runda de river, in care se extrage aleator inca o carte din pachet.

Fiecare runda de pariuri este pornita de primul jucator, acesta putand alege un multiplu de 10 fise pe care sa-l parieze. Al doilea jucator poate egala pariul sau poate ceda mana, fisele pariate transferandu-se la primul jucator.

Dupa fiecare river se va decide mana castigatoare, fisele pariate transferandu-se corespunzator. Jucatorul care ramane primul fara fise pierde jocul.

**Comunicare Java – Alt limbaj**

Scrieti un program Java care primeste de la tastatura numere intregi pozitive. Daca numerele sunt mai mici de 1.000.000, programul Java calculeaza divizorii acestora si ii afiseaza in consola. Altfel, programul memoreaza numerele intr-o coada. Un fir de executie trimite, o data la 10 secunde, unul dintre numerele din coada catre un program scris intr-un alt limbaj de programare la alegere. Programul respectiv calculeaza divizorii numarului si ii intoarce la programul Java, care ii afiseaza in consola, cu mesajul “Au venit divizorii numarului X”, X fiind numarul mai mare de 1.000.000. Programul Java nu asteapta raspunsul pentru numerele mari, ci va accepta in continuare intrari.

**Joc “Blackjack”**

Se va defini setul de carti de joc.

Jucatorul isi va introduce numele si i se vor atribui 100 de fise din care va trebui sa parieze pentru a incepe jocul. Se va extrage aleator o carte din set si se va afisa.

Dupa fiecare carte aleasa din set, jucatorul va avea optiunea de a-si incheia runda sau de a mai alege o carte, valoarea acesteia adaugandu-se la total. Daca totalul depaseste 21, i se va afisa mesajul “Ai pierdut”, iar suma pariata se va scadea din cele 100.

Dupa ce jucatorul incheie runda (cu o suma de cel mult 21), este randul dealer-ului (calculatorul). Se va extrage aleator cate o carte din set. Daca totalul dealer-ului depaseste totalul jucatorului, fara a depasi 21, se va afisa mesajul “Ai pierdut”, iar suma pariata se va scadea din cele 100. Daca totalul dealer-ului depaseste 21, se va afisa mesajul “Ai castigat” iar suma pariata se va aduna la cele 100.

Jocul se incheie fie cand jucatorul ajunge la 0 fise, fie cand ajunge la 200 de fise.

**Aplicatie calculare calorii**

Construiti un calculator de calorii. Salvati caloriile calculate pentru fiecare zi / ora. Creati un calendar cu interfata grafica, in care utilizatorul va putea sa vada istoricul caloriilor. Lansati o alarma sonora daca consuma prea multe calorii.

**Aplicatie convertor valutar**

Folosind JavaFX construiti un convertor valutar, cu urmatoarele valute: Euro, Dolar american, RON, Fanc elvetian, Yuan chinezesc, Dolar UAE, Dolar australian, Peso mexican, Yen japonez, Leu moldovenesc. Cursul valutar se va updata cu o valoare random o data pe minut. Folositi RON ca referinta. Utilizatorul va putea alege intre ce valute vrea sa converteasca si va introduce o suma. I se va afisa valoarea convertita

Faceti un meniu in care sa grupati in functie de continentul de origine al valutei. Utilizatorul va alege un grup si va vedea cursurile din acel grup raportate la RON.

**Joc “Snake”**

Implementati un joc “Snake” cu interfata grafica. Pentru ghidarea sarpelui se vor folosi tastele sageti. Mancarea va aparea random pe grila. Sarpele moare daca isi musca propria coada.

**Joc “Tetris”**

Implementati un joc “Tetris” cu interfata grafica. Trebuie definite formele. In momentul in care se formeaza o linie pe grila, aceasta trebuie sa dispara si toate piesele de deasupra ei sa coboare un rand.

**Joc “Sah”**

Se va afisa grila 8x8, colorata corespunzator, si cele 32 de piese marcate corespunzator. Incepand cu culoarea alba, fiecare jucator va putea muta o piesa alegand-o si apoi alegand campul tinta. Trebuie sa poata fi capturate piesele, acestea disparand de pe grila. Trebuie sa se afiseze un mesaj de eroare la mutari ilegale (ex: pion in lateral).

**Simulare senzori autoturism**

Definiti senzorii autoturismului. Lansati functionarea fiecarui senzor pe un thread, valorile fiind generate aleator. Definiti alerte daca valorile nu se regasesc in niste intervale de referinta. Afisati grafic martori ai senzorilor si aprindeti-i daca este o situatie de alerta.

**Server web**

Se va construi un server web care va putea interactiona cu o baza de date cu tabelul Studenti. Studentii vor avea nume, prenume si data nasterii. Se vor putea insera sau aduce date folosind apeluri POST si GET.

**Heat-map tastatura**

Se va construi o aplicatie care va rula in background. Aceasta va inregistra numarul de apasari pe taste. Se va reprezenta grafic o tastatura si se vor colora tastele pe aceasta in functie de numarul de apasari. Se vor putea configura limitele pentru schimbarea culorilor.

**Joc “Table”**

Implementati jocul de table pentru 2 jucatori, cu interfata grafica.

**Joc “Parcheaza masina”**

Folosind JavaFX, construiti o grila de 21x21 campuri. In centru este o zona de 5x5 campuri reprezentand parcarea. Parcarea este inconjurata de un zid cu latimea de un camp, exceptand o deschidere de 3 campuri in partea de jos.

In afara parcarii va aparea o masina care va trebui condusa folosind tastele sageti catre intrarea in parcare (unul dintre cele 3 campuri). Daca masina ajunge pe unul dintre cele 3 campuri, se ocupa un loc in parcare si apare urmatoarea masina. Daca masina loveste zidul se afiseaza mesajul “Accident”.

**Joc “Scrabble”**

Implementati un joc de Scrabble turn-based. Trebuie definita corect grila de joc, piesele cu litere disponibile si punctajele lor. Se poate configura numarul de jucatori.

**Joc “Whack-a-mole”**

Folosind JavaFX, construiti un joc Whack-a-mole. Se va construi o grila de 5x5 campuri pe care apar aleatoriu cartite care trebuie lovite. Cartitele stau vizibile un timp dat. Daca se da click pe cartita cat timp este vizibila, se obtine un punct. Jocul dureaza un timp dat.

**Joc “Checkers”**

Implementati jocul checkers pentru 2 jucatori, cu interfata grafica.

**Impodobirea pomului de Craciun**

Avem un pom de Craciun cu 10 globuri, repartizate piramidal astfel:

- 1 glob pe primul rand (1a)

- 2 globuri pe al doilea rand (2a si 2b)

- 3 globuri pe al treilea rand (3a, 3b si 3c)

- 4 globuri pe al patrulea rand (4a, 4b, 4c si 4d)

Se va alege un glob de pornire, care va avea culoarea rosie. Folosind culorile rosu, galben si albastru, trebuie sa se coloreze toate globurile astfel incat doua globuri vecine sa nu aiba aceeasi culoare. Vecinatatile sunt (1a, 2a), (1a, 2b), (2a, 2b), (2a, 3a), (2a, 3b), (2b, 3b), (2b, 3c), (3a, 3b), (3a, 4a), (3a, 4b), (3b, 3c), (3b, 4b), (3b, 4c), (3c, 4c), (3c, 4d), (4a, 4b), (4b, 4c), (4c, 4d).

Se vor reprezenta grafic globurile colorate.

**Finger dance revolution**

Se va introduce numele jucatorului. I se va afisa aleator, o data la o secunda, cate o tasta (litera sau cifra) pe care trebuie sa o apese. Daca apasa corect, va primi un punct. Daca apasa gresit sau nu apasa in respectiva secunda, i se va salva punctajul intr-un fisier text.

Se va defini un dictionar cu 10 cuvinte de 7 litere. Daca jucatorul apasa corect 20 de taste, se va alege aleator un cuvant din dictionar. Jucatorul va avea la dispozitie 3 secunde pentru a-l introduce de la tastatura. Daca reuseste, se va afisa mesajul “Ai castigat”.

**Joc “Battleboats”**

Implementati un joc de Battleboats pentru 1 sau 2 jucatori. Pentru jocul cu un singur jucator, implementati un robot care plaseaza navele si ataca aleator. Grila de joc are 10x10 campuri. Navele disponibile sunt barcuta (1 camp), submarinul (3 campuri coliniare, dar nu diagonale), crucisatorul (4 campuri in T) si portavionul (5 campuri in X, pe diagonala). La plasarea navelor, trebuie sa existe spatiu de cel putin un camp intre ele.

**Tragere la sorti UCL**

Se vor introduce 32 de echipe impartite in 4 urne valorice (cate 8 in fiecare urna). Pentru fiecare echipa se va specifica tara de origine. Se vor imparti echipele in 8 grupe a cate 4 echipe. In fiecare grupa va fi o echipa din fiecare urna valorica. In fiecare grupa trebuie sa fie maxim 2 echipe din aceeasi tara.

In faza grupelor, fiecare echipa va juca cu celelalte din grupa in sistem tur-retur. Afisati toate meciurile ce se vor desfasura in faza grupelor.

**Loto**

Se pot inregistra jucatori. Pentru fiecare jucator se pot adauga bilete. Biletele vor contine 6 numere intre 1 si 49 si o suma jucata. Se va afisa o eroare daca un jucator incearca sa adauge acelasi numar pe acelasi bilet de mai multe ori.

Se va putea porni o extragere, in care se vor alege aleator 6 numere intre 1 si 49. Se va verifica daca exista un bilet ce contine toate cele 6 numere si, daca exista, se va afisa mesajul “X a castigat Y”, X fiind numele jucatorului castigator, iar Y fiind totalul sumelor jucate pe toate biletele. Altfel, se va afisa “Nimeni nu a castigat la aceasta extragere”, iar suma se va reporta pentru urmatoarea.

**Transpiler Java in limba romana**

Scrieti un program capabil sa execute cod Java scris in limba romana (ex: clasa publica Pisica{}).

**Automate de parcare**

Se considera o parcare cu 10 locuri. Intrarea unei masini in parcare se marcheaza prin introducerea de la tastatura a numarului de inmatriculare. Daca se incearca intrarea a mai mult de 10 masini se va afisa mesajul “Parcarea este plina”.

Pentru fiecare masina se va masura timpul petrecut in parcare pentru a calcula suma de plata: 1 pentru maxim 10 secunde petrecute in parcare, 2 pentru maxim 20 de secunde, 3 pentru maxim 30 de secunde, respectiv 5 pentru mai mult de 30 de secunde.

Se vor putea scoate masini din parcare, de asemenea prin introducerea de la tastatura a numarului de inmatriculare. Se va afisa mesajul “X are de plata Y”, X fiind numarul de inmatriculare, iar Y fiind suma de plata. Daca se incearca scoaterea unei masini care nu este in parcare se va afisa mesajul “Masina nu a fost gasita”.

**Urmarire avioane**

Definiti 20 tari din Europa ca poligoane de coordonate geospatiale (nu trebuie sa fie perfect, doar sa semene aproximativ forma). Putem alege un punct de plecare (coordonate geospatiale) si un punct destinatie. Considerand ca atat in punctul de plecare, cat si in punctul destinatie exista aeroporturi, si ca zborurile au loc in linie dreapta, specificati pentru fiecare traseu pe deasupra cator tari se zboara.

**Formular sign-up**

Folosind JavaFX construiti un formular in care studentii se pot inregistra intr-un site web. Formularul trebuie sa contina un camp pentru adresa de e-mail si un camp tip parola. Validati parola sa contina minim 10 caractere, doar litere latine si cifre, si minim 3 cifre. Validati formatul adresei de e-mail. Daca informatiile introduse sunt valide, salvati-le intr-un fisier text.

**Simulator mecanica**

Implementati un formular in care se vor introduce forta, unghiul cu care forta actioneaza asupra corpului, masa corpului, coeficientul de frecare, unghiul pe care planul pe care se deplaseaza corpul il face cu planul orizontal, timpul cat forta actioneaza asupra corpului. Considerand corpul in repaus in momentul inceperii actiunii fortei, afisati viteza pe care o va avea corpul in momentul in care forta isi inceteaza actiunea. Reprezentati grafic sistemul descris.