

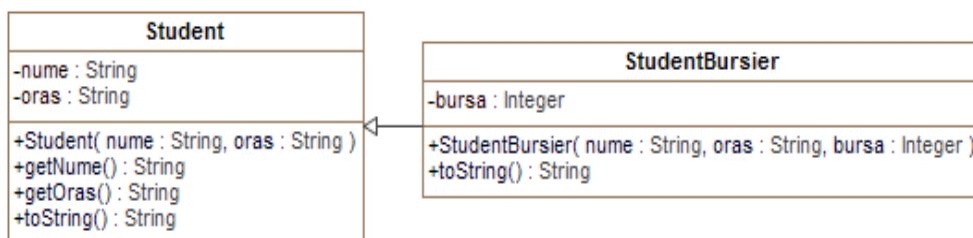


Examen de licență iunie 2016 Specializarea Informatică Română

Subiectul 1

Scrieți un program într-unul din limbajele de programare Python, C++, Java, C# care:

(a) **Definește clasele** Student și StudentBursier pe baza următoarei diagrame UML:



- Nume și oraș trebuie să fie nenule și nevide, iar bursa trebuie să fie strict pozitivă. Constructorii trebuie să impună constrângerile.
 - Metoda toString() din clasa Student returnează un șir de caractere conținând numele studentului concatenat cu orașul de proveniență, iar metoda toString() din StudentBursier adaugă bursa la șirul de caractere conținând numele și orașul.
- (b) **Definește o funcție** care inserează un obiect de tip Student într-o listă de studenți ordonată alfabetic după nume, astfel încât lista să rămână ordonată.
- (c) **Definește o funcție** care tipărește o listă de studenți.
- (d) **Definește o funcție** care primește ca parametru o listă de studenți și tipărește, pentru fiecare oraș, lista studenților bursieri (ordonată după nume) din acel oraș. Fiecare oraș va fi tipărit o singură dată, în orice ordine.
- (e) **Funcția principală** a programului apelează funcția indicată la punctul (b) pentru a construi o listă cu patru studenți, din două orașe “Arad” și “Oradea”, trei dintre ei fiind bursieri. Apoi, folosind funcția de la punctul (c), se tipărește lista studenților. La final, se apelează funcția de la punctul (d).
- (f) Pentru tipurile de date utilizate în program, scrieți specificațiile operațiilor folosite.

Notă

- Nu se vor folosi containere sortate și operații de sortare.
- Nu se vor defini alte metode decât cele specificate în enunț.

Pentru tipurile de date puteți folosi biblioteci existente (Python, C++, Java, C#).

Subiectul 2

a. Creați o bază de date relațională, având toate tabelele în 3NF, ce va reține următoarele informații pentru un sistem de rezervări online:

turiști: nume, email, cod oraș, nume oraș, cod țară, nume țară;

unități de cazare: denumire, cod oraș, nume oraș, cod țară, nume țară, cod tip cazare, nume tip cazare (exemple de valori: pensiune, hostel, hotel, etc.), număr stele, scor, preț / noapte și o listă de rezervări, pentru fiecare rezervare cunoscându-se turistul, data de început și numărul de nopți.

Arătați că baza de date creată e în 3NF, identificând dependențele funcționale.

b. Pentru baza de date de la punctul a, să se rezolve, folosind algebra relațională sau SQL, următoarele interogări:

- Turiștii (nume și email) care au rezervat cel puțin o pensiune cu scor mai mare ca 9, dar nu au rezervat nici un hotel de 3 stele cu scor mai mic ca 9.
- Numărul total de rezervări ale tuturor turiștilor din Cluj într-un hotel de 5 stele din Paris în anul 2015.
- Unitatea de cazare (denumire, nume oraș, nume tip cazare și număr stele) în care au fost cele mai multe rezervări în ultimii 5 ani.

Subiectul 3

3.1 Se dă fișierul **grep.c** care conține fragmentul de cod de mai jos și care se compilează în directorul personal al utilizatorului sub numele **grep**. Răspundeți la următoarele întrebări, considerând că toate instrucțiunile se execută cu succes.

<pre>1 int main(int c, char** v) { 2 int p[2], n; 3 char s[10] = "ceva"; 4 pipe(p); 5 n = fork(); 6 if(n == 0) { 7 close(p[0]); 8 printf("înainte\n"); 9 if(c > 2) 10 execlp("grep", "grep", v[1], v[2], NULL); 11 strcpy(s, "după"); 12 write(p[1], s, 6); 13 close(p[1]); 14 exit(0); 15 } 16 close(p[1]); 17 read(p[0], s, 6); 18 close(p[0]); 19 printf("%s\n", s); 20 return 0; 21 }</pre>	<p>a) Enumerați și explicați valorile posibile ale variabilei n.</p> <p>b) Ce vor afișa pe ecran următoarele rulări, considerând că directorul personal al utilizatorului nu se află în variabila de mediu PATH.</p> <p>b.1) grep grep grep.c b.2) ./grep grep grep.c b.3) ./grep grep</p>
---	---

3.2 Se dă fișierul **abc.sh** conținând scriptul Shell UNIX de mai jos. Răspundeți la următoarele întrebări.

<pre>1 n=0 2 for i in `cat \$1`; do 3 c=`echo \$i cut -c1` 4 if echo \$i grep -q "^[0-9][0-9]*\$"; then 5 echo \$i >> \$1.nr 6 elif echo \$c grep -q "[A-Za-z]"; then 7 echo \$i >> \$c 8 else 9 n=`expr \$n + 1` 10 fi 11 done 12 echo \$n</pre>	<p>a) Explicați expresia regulară de la linia 4</p> <p>b) Ce se întâmplă dacă scriptul este rulat fără argumente?</p> <p>c) Ce va tipări pe ecran rularea ./abc.sh f3 și ce fișiere (nume și conținut) va crea, dacă f3 conține "abc 74 2-8 aa 3a =c b2" și rularea se face într-un director conținând doar fișierele abc.sh și f3?</p> <p>d) Dați un exemplu de fișier f3 astfel încât rularea de la punctul precedent să creeze 4 fișiere noi astfel încât numele niciunuia să nu aibă prefixul f3.</p>
--	---

Notă: Toate subiectele sunt obligatorii. Fiecare subiect se notează între 1 și 10 de către ambii corectori.
Timpe de lucru: 3 ore.

BAREM
INFORMATICĂ

Subiect 1 (Algoritmă și Programare):

Oficiu – 1p

Definirea clasei Student– **0.75p** din care

atribute – 0.25

constructor – 0.25

metode - 0.25

Definirea clasei StudentBursier– **1.25p** din care

relația de moștenire – 0.25

constructor – 0.5

metoda toString() – 0.5

Funcția de la punctul b) – **2p** din care

signatura corectă - 0.1p

algoritm de inserare în interiorul listei - 1.8p

– parcurgere listă și determinarea poziției de inserare – 1.6p

– adăugare element pe poziția determinată anterior – 0.2

returnare rezultat - 0.1p

Funcția de la punctul c) - **0.5p** din care

signatura corectă - 0.1p

parcurgere listă și afișare – 0.4p

Funcția de la punctul d) – **2.5p** din care

determinare lista bursieri din orașe – 2p

tipărire perechi – 0.5p

Funcția de la punctul e) – **0.5p**

Specificațiile operațiilor tipurilor de dată folosite– **1.5p**

Subiect 2 (Baze de date)

1 punct oficiu

Problema a:

1 punct pentru dependențe funcționale

2 punct pentru tabelele în 3NF;

1 punct pentru justificare.

Problema b:

1 puncte pentru b1

1.5 puncte pentru b2

2.5 puncte pentru b3

Subiect 3 (Sisteme de operare):

Oficiu: 1p

3.1

a) 1p - Valorile posibile

1p - Explicații

b.1) 1p - Afișează linia 10

b.2) 1p - Afișează “înainte”, linia 10 și “ceva”

b.3) 1p - Afișează “înainte” și “după”

3.2

a) 1p - Secvență una sau mai multe cifre

b) 0.5p - Comanda “cat” așteaptă input de la intrarea standard

c) 0.5p - Afișează 3 pe ecran

0.5p - Fișierul f3.nr conținând 74

0.5p - Fișierele dicționar a și b, conținând "abc aa" respectiv "b2"

d) 1p - Orice secvență de cuvinte dintre care 4 încep cu litere diferite