



Examen de licență iunie 2013 Specializarea Informatică Română

Subiectul 1

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

- Definește o clasă **Produs** având un atribut privat **denumire** de tip șir de caractere, o metodă publică de accesare a atributului **denumire**, un constructor public pentru inițializarea denumirii și o metodă publică de tipărire care afișează atributul **denumire** la ieșirea standard.
- Definește o clasă **ProdusPerisabil** derivată din **Produs** având un atribut privat **valabilitate** de tip întreg, un constructor public pentru inițializarea denumirii și valabilității și de asemenea o metodă publică de tipărire pe ieșirea standard care va afișa atributul **denumire** din clasa de bază reutilizând tipărirea din clasa de bază și atributul **valabilitate**.
- Construiește în program o listă conținând în ordine *alfabetică* după **denumire** următoarele: un obiect de tip **Produs** având **denumire** egală cu "Bec"; un obiect de tip **ProdusPerisabil** având **denumire** egală cu "Brânză" și **valabilitate** egală cu 60; un obiect de tip **Produs** având **denumire** egală cu "Cană". După construirea listei, se va citi de la intrarea standard denumirea unui produs și se va căuta *secvențial* în listă produsul având denumirea citită (căutarea va ține cont de ordonarea listei).
- Pentru tipul de dată **listă** utilizat în program, scrieți specificațiile operațiilor folosite.

Se pot folosi biblioteci existente pentru structuri de date (C++, Java, C#). În cazul în care nu se folosesc biblioteci existente, nu se cere implementare pentru operațiile listei.

Subiectul 2

- Să se evidențieze dependențele funcționale pentru următoarele informații dintr-o bibliotecă electronică:
 - documente**: cota, tip document (articol în revistă, carte, raport de cercetare, document web, etc.), lista autorilor (un document poate avea 0, 1, 2, ... autori), titlu, an apariție;
 - cititori**: cod cititor, nume, lista documentelor consultate;

Se cere o bază de date relațională cu tabele în 3NF pentru informațiile de mai sus. Justificați că tabelele obținute sunt în 3NF.

- Pentru baza de date de la punctul a, să se rezolve, folosind algebra relațională sau Select-SQL, următoarele interogări:
 - Codul și numele cititorilor care au consultat cel puțin un "raport de cercetare" și nu au consultat nici o "carte".
 - Codul și numele cititorilor care au consultat documente care au cel puțin 5 autori.

Subiectul 3

a. Fragmentele de cod de mai jos se rulează concurrent pe același calculator. Considerați că:

- fișierele FIFO a2b și b2a au fost create și sunt vide
- toate instrucțiunile se execută fără eroare
- operațiile pe FIFO nu sunt afectate de O_NDELAY

Răspundeți la următoarele întrebări:

1. Care sunt toate variantele posibile de afișare în consola programului P1?
2. Ce se întâmplă dacă se inversează ordinea instrucțiunilor `open` din programul P2?
3. Câte procese creează procesul P1 (excluzând procesul părinte) ?
4. Câte procese creează procesul P1 (excluzând procesul părinte) dacă lipsește instrucțiunea `exit(0)`?

P1	P2
<pre>int main() { int i, n=1; int a2b = open("a2b", O_WRONLY); int b2a = open("b2a", O_RDONLY); for(i=0; i<3; i++) { if(fork() == 0) { n += i; write(a2b, &n, sizeof(int)); read(b2a, &n, sizeof(int)); printf("%d: %d\n", getpid(), n); exit(0); } } wait(0); wait(0); wait(0); close(a2b); close(b2a); return 0; }</pre>	<pre>int main() { int i, n; int a2b = open("a2b", O_RDONLY); int b2a = open("b2a", O_WRONLY); for(i=0; i<3; i++) { read(a2b, &n, sizeof(int)); n++; write(b2a, &n, sizeof(int)); } close(a2b); close(b2a); return 0; }</pre>

b. Considerați fragmentul de script UNIX shell de mai jos.

1. Explicați funcționalitatea liniei 2.
2. Explicați funcționalitatea liniei 3.
3. Ce se afișează la fiecare dintre rulările de mai jos?

Linie	Scriptul a.sh	Rulări
1	SUM=0	R1: ./a.sh 1 2 3 4
2	for A in \$*; do	R2: ./a.sh a b1 c d2 5
3	N=`echo \$A grep "[0-9]\$" `	R3: ./a.sh 1 20 300
4	if ["\$N" != ""]; then	R4: ./a.sh
5	SUM=`expr \$SUM + \$N`	
6	fi	
7	done	
8	echo \$SUM	

Notă: Toate subiectele sunt obligatorii. Fiecare subiect se notează cu notă întregă între 1 și 10 de către ambii corectori.

Timp de lucru: 2 ore.

BAREM

Subiect 1 (Algoritmă și Programare):

Oficiu – 1p

Definirea clasei **Produs** – 1p

atribut – 0.25

constructor – 0.25

accesor – 0.25

tipărire - 0.25

Definirea clasei **ProdusPerisabil** – 1.5p

relația de moștenire – 0.25

constructor – 0.5

atribut – 0.25

tipărire – 0.5

Program – construirea listei – 1.5p

declarare listă – 0.75p

adăugare elemente – 0.75p

Program – căutarea produsului cerut – 3p

citire denumire – 0.5p

comparare șiruri de caractere – 0.5p

parcurgere listă – 1p

optimizarea parcurgerii pentru liste ordonate – 1p

Specificațiile operațiilor folosite din Tipul de dată **Listă** – 2p

Subiect 2 (Baze de date):

Oficiu **1 punct**

Cerința a:

pentru dependențe **1 punct**

pentru tabele în 3NF **2 puncte**

pentru justificare **1 punct**

Cerința b:

pentru b1 **2 puncte**

pentru b2 **3 puncte.**

Subiect 3 (Sisteme de Operare):

Oficiu **1 punct**

Cerinta a - **5 puncte:**

- a.1 - **2 puncte** (1p pentru o varianta corecta, cate 0.2p pentru restul de 5)
- a.2 - **1 punct** (0.5p pentru deadlock, 0.5p explicatia)
- a.3 - **1 punct**
- a.4 - **1 punct**

Cerinta b - **4 puncte:**

- b.1 - **1 punct**
- b.2 - **1 punct**
- b.3 - **2 puncte** (0.5p/rulare)