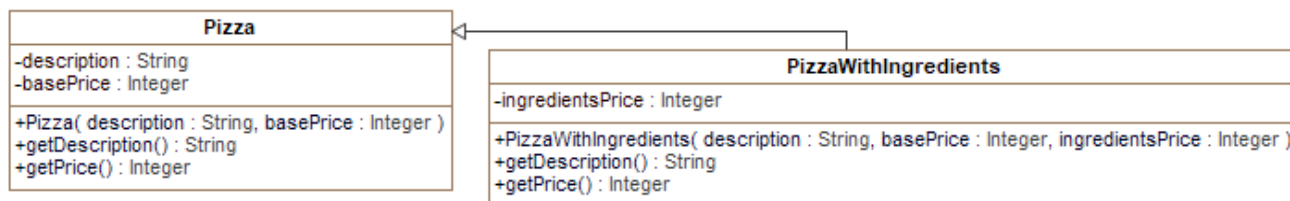


Proba scrisă a examenului de licență, 5 septembrie 2017
Informatică Română
VARIANTA 1

SUBIECTUL 1. Algoritmă și programare

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

a) **Definește** clasele **Pizza** și **PizzaWithIngredients** pe baza următoarei diagrame UML:



- *Description* (descriere) trebuie să fie nenul și nevid, iar *basePrice* (pret baza) și *ingredientsPrice* (pret ingrediente) trebuie să fie strict pozitive. Constructorii trebuie să impună constrângerile.
 - Metoda **getDescription()** din clasa **Pizza** returnează *description*, iar metoda **getPrice()** returnează *basePrice*. Metoda **getDescription()** din clasa **PizzaWithIngredients** adaugă textul “+ ingredients” la *description* din clasa de bază, iar metoda **getPrice()** din clasa **PizzaWithIngredients** returnează prețul din clasa de bază plus prețul ingredientelor.
- b) **Definește o funcție** care având ca parametri două liste conținând obiecte de tip **Pizza** ordonate descrescător după preț, returnează o nouă listă conținând toate obiectele din cele două liste parametru, ordonată descrescător după preț. Complexitatea *timp* a funcției trebuie să fie *liniară* .
- c) **Definește o funcție** care sortează, în ordine descrescătoare după preț, o listă conținând obiecte de tip **Pizza**. Sortarea se va face astfel: elementele listei sunt parcurse secvențial (de la poziția a doua până la ultima); se va insera elementul de pe poziția *i* în sublista (deja sortată) conținând elementele de la prima poziție la poziția *i-1* .
- d) **Definește o funcție** care tipărește o listă cu obiecte de tip **Pizza** și suma prețurilor lor.
- e) **Funcția principală** a programului creează două liste cu obiecte de tip **Pizza**. În prima listă se adaugă trei obiecte (alegeți voi descrierea și prețul): două **Pizza** și o **PizzaWithIngredients**. În a doua listă se adaugă trei obiecte (alegeți voi descrierea și prețul): o **Pizza** și două **PizzaWithIngredients**. Sortați cele două liste, folosind funcția de la c), apoi formați din cele două liste o listă *L* folosind funcția de la b). La final, tipăriți lista *L* folosind funcția de la punctul d).
- f) Pentru tipul de date **Listă** utilizat în program, scrieți specificațiile operațiilor folosite.

Notă

- Se va indica limbajul de programare folosit.
- Nu se vor folosi containere sortate și operații de sortare predefinite.
- 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. Baze de date

a. Creați o bază de date relațională, având toate tabelele în a treia formă normală, care va reține următoarele informații pentru Examenul de Licență 2017:

- **comisii**: denumire comisie, specializare (unde o specializare are un cod și o denumire, exemple de denumire: *Informatică engleză, Informatică română, Matematică maghiară, Informatică germană* etc*);
- **studenți**: nume, titlul lucrării de licență, cod profesor coordonator, nume profesor coordonator, o listă de cuvinte cheie asociate licenței (unde un cuvânt cheie are un cod și denumire, exemple de denumiri: *big data, data mining* etc) și evaluarea (comisia la care studentul susține lucrarea și notă finală acordată de comisie).

Justificați că baza de date creată este în a treia formă normală, identificând dependențele funcționale.

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

- Lista comisiilor (denumire și specializare) care au evaluat cel puțin o lucrare cu tematica *data mining* și cel puțin una cu tematica *cloud computing*.
- Numărul studenților coordonați de profesorul *Ion Popescu* și care au primit notă de trecere (≥ 6) din partea comisiilor de la *Informatică engleză*.
- Lista profesorilor (cod și nume) cu cel mai mare număr de studenți cu note peste 8.

* Pot exista mai multe comisii pe aceeași specializare, de exemplu, pentru specializarea Informatică engleză pot exista comisiile *Comisia IE 1*, *Comisia IE 2* și *Comisia IE 3*.

SUBIECTUL 3. Sisteme de operare

3.1 Răspundeți la următoarele întrebări, considerând că toate instrucțiunile din fragmentul de cod de mai jos se execută cu succes.

<pre> 1 int main () { 2 int p1[2], p2[2], x, y, z, m1, m2; 3 pipe(p1); pipe(p2); 4 x=fork(); 5 if (x!=0) y=fork(); 6 if(x==0){ 7 z=getpid(); 8 close(p1[0]); close(p2[1]); 9 write(p1[1], &z, sizeof(int)); close(p1[1]); 10 read(p2[0], &m1, sizeof(int)); close(p2[0]); 11 printf("x=%d: %d\n", x, m1); 12 exit(0); 13 } 14 if(y==0){ 15 z=getpid(); 16 close(p1[1]); close(p2[0]); 17 write(p2[1], &z, sizeof(int)); close(p2[1]); 18 read(p1[0], &m2, sizeof(int)); close(p1[0]); 19 printf("y=%d: %d\n", y, m2); 20 exit(0); 21 } 22 printf("x=%d, y=%d\n", x, y); 23 close(p1[0]); close(p1[1]); close(p2[0]); close(p2[1]); 24 wait(NULL); wait(NULL); 25 return 0; 26 }</pre>	<ol style="list-style-type: none"> Desenați ierarhia proceselor create, incluzând și procesul părinte. Care dintre procese va executa linia 11? Ce reprezintă valoarea variabilei <i>m1</i> afișată? Care dintre procese va executa linia 19? Ce reprezintă valoarea variabilei <i>m2</i> afișată? Care dintre procese va executa linia 22? Ce reprezintă valorile variabilelor <i>x</i> și <i>y</i> afișate? Există vreo relație între valoarea variabilei <i>m1</i> din linia 11 și valoarea variabilei <i>x</i> din linia 22? Justificați răspunsul.
--	--

3.2 Răspundeți la următoarele întrebări, considerând că scriptul Shell UNIX *a.sh* de mai jos, se află într-un director care mai conține doar fișierele specificate dedesubt.

<pre> 1 for f in *.txt; do 2 if file \$f grep -q "text"; then 3 if grep -q "^[^\$1]" \$f; then 4 echo "\$f" 5 fi 6 elif echo \$f grep -q "^[^\$1]"; then 7 echo \$f 8 fi 9 done</pre>	<ol style="list-style-type: none"> Ce valori va lua variabila <i>f</i>? Ce va afișa rularea comenzii <i>./a.sh a</i> Explicați expresiile regulate din liniile 3 și 6. Dați un nume și conținut de fișier text, cu trei rânduri, pentru care linia 4 se execută la rularea comenzii <i>./a.sh b</i>
<ul style="list-style-type: none"> <i>abc.txt</i>, fișier text cu conținutul <div style="border: 1px solid black; padding: 2px; display: inline-block;"> abc abb </div> <i>ubb.txt</i>, fișier text cu conținutul <div style="border: 1px solid black; padding: 2px; display: inline-block;"> aaa bbb </div> <i>ab.txt</i>, fișier ZIP, redenumit cu extensia <i>txt</i> 	

NOTĂ.

- Toate subiectele sunt obligatorii. La toate subiectele se cer rezolvări cu soluții complete.
- Nota minimă ce asigură promovarea este 5,00.
- Timpul efectiv de lucru este de 3 ore.

BAREM INFORMATICĂ

Subiect 1 (Algoritmica și Programare):

Oficiu – 1p

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

atribute – 0.25

constructor – 0.25

metode - 0.25

Definirea clasei **PizzaCuUnIngredient**– 1.05p din care

relația de moștenire – 0.25

constructor – 0.4

metode – 0.4

Funcția de la punctul b) – 2.2p din care

signatura corectă - 0.1p

algoritmul de combinare - 2p

– parcurgere simultană a listelor și adăugarea elementului maxim în lista rezultat – 1p

– adăugarea în rezultat a elementelor rămase în prima listă – 0.5

– adăugarea în rezultat a elementelor rămase în a doua listă – 0.5

returnare rezultat - 0.1p

Funcția de la punctul c) – 2p din care

signatura corectă - 0.1p

parcurgere listă – 0.2p

inserarea elementului i în sublista (sortată) de pe pozițiile $0/1...i-1$ – 1.7p

Funcția de la punctul d) – 1p din care

signatura corectă - 0.1p

parcurgere listă – 0.4p

accesare și tipărire element – 0.25p

calcul și tipărire preț total – 0.25p

Funcția principală e) – 0.5p

Specificațiile operațiilor folosite pentru tipul de dată **Listă**– 1.5p

Subiect 2 (Baze de date)

1 punct oficiu

Problema a:

2 puncte pentru tabelele în 3NF

2 puncte pentru justificare:

1 punct definiții

1 punct explicații

Problema b:

1.5 puncte pentru i

1 punct pentru ii

2.5 puncte pentru iii

Subiect 3 (Sisteme de operare):

Oficiu – 1p

3.1 – 5p din care

a) Diagrama ierarhiei - 1p

b) Primul proces fiu – 0.5p

PID-ul celui de-al doilea proces fiu – 0.5p

c) Al doilea proces fiu – 0.5p

PID-ul primului proces fiu – 0.5p

d) Procesul părinte – 0.5p

x este PID-ul primului proces fiu – 0.5p

y este PID-ul celui de-al doilea proces fiu – 0.5p

e) Sunt PID-urile celor două procese fiu (frați) – 0.5p

3.2 – 4p din care

a) Numele fișierelor cu extensia `txt` – 0.75p

Valorile specifice: `abc.txt`, `ubb.txt`, `ab.txt` – 0.25p

- b) Pe linii separate: `ab.txt` și `*ubb.txt*` – 1p
- c) Linia 3: orice linie care nu începe cu vreunul din caracterele primului argument – 0.5p
Linia 6: orice linie care începe primul argument – 0.5p
- d) Fișier text cu extensia `txt` – 0.5p
Conținut cu trei linii dintre care cel puțin una nu începe cu `b` – 0.5p