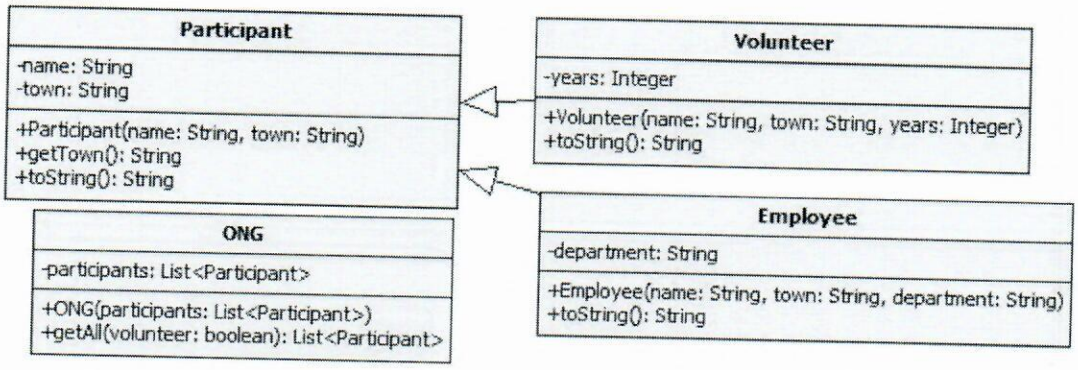
UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Proba scrisă a examenului de licenţă, 4 septembrie 2018
Informatică Română
VARIANTA 2

SUBIECTUL 1. Algoritmică şi programare

Scrieţi un program într-unul din limbajele de programare Python, C++, Java, C#, cu următoarele cerinţe:

a) **Defineşte** clasele **Participant**, **Volunteer**, **Employee** şi **ONG** pe baza următoarei diagrame UML:



- *name* (numele participantului), *town* (oraşul participantului) şi *department* (departamentul la care lucrează angajatul) trebuie să fie nenule şi nevide, iar *years* (anii de voluntariat) trebuie să fie o valoare strict pozitivă. Constructorii trebuie să impună constrângerile.
- Metoda **toString()** din clasa **Participant** returnează *name* concatenat cu *town*. Metoda **toString()** din clasa **Volunteer** returnează textul "Volunteer <*years*> years" concatenat cu **toString()** returnat de clasa de bază, iar metoda **toString()** din clasa **Employee** returnează textul "Employee <*department*>" concatenat cu **toString()** returnat de clasa de bază.
- Clasa **ONG** conţine o listă *participants* de obiecte de tip **Participant**. Metoda **getAll**(*volunteer*:boolean) din **ONG** returnează lista participanţilor voluntari (de tip **Volunteer**) sau angajaţi (de tip **Employee**) în funcţie de valoarea parametrului *volunteer*.

b) **Defineşte o funcţie** care, având ca parametru un obiect de tip **ONG,** sortează şi returnează o listă cu voluntarii (de tip **Volunteer**) din acel **ONG,** în ordine alfabetică a oraşului de proveniență a acestora (*town*). Sortarea listei se va face folosind sortarea prin interclasare (*Merge Sort*).

c) **Defineşte o funcţie** care, având ca parametru o listă de obiecte de tip **ONG,** returnează oraşul având număr maxim de angajaţi (de tip **Employee**). În cazul în care sunt mai multe astfel de oraşe, se va returna unul dintre acestea.

d) **Defineşte o funcţie** care, având ca parametru o listă de obiecte de tip **ONG**, returnează numărul total al angajaţilor (de tip **Employee**) din lista de **ONG**-uri.

e) **Funcţia principală** a programului creează o listă cu două obiecte de tip **ONG** (alegeţi voi valori pentru proprietăţile lor): primul **ONG** conţine două obiecte de tip **Volunteer** şi unul de tip **Employee**, al doilea conţine un obiect de tip **Volunteer** şi unul de tip **Employee**. Apelaţi funcţia de la b) pentru al doilea ONG din listă, apoi apelaţi funcţiile de la c) şi d) pentru lista de **ONG**-uri şi afişaţi rezultatele obţinute în urma apelurilor.

f) Pentru tipul de date **Listă** utilizat în program, scrieţi specificaţiile operaţiilor folosite.

Notă

- **Se va indica limbajul de programare folosit.**
- **Pentru clasele din diagrama UML, nu se vor defini alte metode decât cele specificate.**
- **Nu se vor folosi containere sortate şi operaţii de sortare predefinite.**

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

SUBIECTUL 2. Baze de date

Fie o bază de date care stochează date despre toate zborurile de pasageri din ultimii 50 de ani. Baza de date are următoarea structură:

- tabelul ***Companii*** cu câmpurile **CodC**, **Denumire**, **Vechime**, **Tara**, **Continent**;
- tabelul ***Aeroporturi*** cu câmpurile **CodA**, **Nume**, **Oras**, **Tara**, **Continent**;

- tabelul ***Zboruri*** cu câmpurile **CodCompanie**, **CodAeroportPlecare**, **CodAeroportSosire**, **Data**, **Ora**, **TipAvion**, **NumarLocuri**.

1. Determinați cheile primare și cheile externe pentru fiecare dintre tabelele de mai sus.
2. Determinați cel puțin 3 dependențe funcționale care se referă la câmpuri care nu reprezintă coduri.
3. Considerați următoarele modificări de structură și precizați care dintre ele sunt esențiale pentru ca baza de date să fie în **3NF** (a 3-a formă normală). În cazul unui răspuns afirmativ, justificați-vă opțiunea.
   a. Crearea unui tabel separat pentru stocarea țărilor.
   b. Utilizarea unui câmp **DataInfiintarii** în locul câmpului **Vechime**.
   c. Crearea unui tabel separat pentru stocarea tipurilor de avioane.
   d. Adăugarea constrângerii de integritate ***CodAeroportPlecare*** > ***CodAeroportSosire***.
4. Scrieți, pe structura inițială, o interogare SQL echivalentă cu următoarea interogare:

$$\Pi_{\text{Denumire}} (\sigma_{\text{Continent = 'Asia'}} (\textbf{\textit{Companii}} \otimes_{\text{CodC = CodCompanie}} \sigma_{\text{NumarLocuri >200}} (\textbf{\textit{Zboruri}})))$$

5. Scrieți o interogare SQL care returnează numărul total de zboruri din și înspre fiecare aeroport (***Nume, NrZboruri***).

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.

```
1  int main(){
2    int pfd[2], i;
3    char buffer, c;
4    pipe(pfd);
5    for(i=0;i<3;i++){
6      if(fork()==0){
7          while(read(pfd[0],&buffer, 1)>0){
8            printf("%c\n", buffer);
9          }
10         close(pfd[0]);
11         close(pfd[1]);
12         exit(0);
13     }
14   }
15   close(pfd[0]);
16   for(i=0;i<10;i++){
17     c = 'a' + i;
18     write(pfd[1],&c,1);
19   }
20   close(pfd[1]);
21   while(wait(NULL)>0){}
22   exit(0);
23 }
```

a) Desenați ierarhia proceselor create, incluzând și procesul părinte.

b) Ce afișează execuția programului?

c) Explicați de ce procesele fiu nu se termină.

d) Între ce linii ați muta linia 11 astfel încât procesele fiu să se termine?

e) Explicați linia 21.

3.2 Scriptul Shell UNIX de mai jos, este salvat într-un fișier numit `a.sh`. Răspundeți la următoarele întrebări considerând execuția comenzii `cat test | ./a.sh`

```
#!/bin/bash
read a
x=0
while [ "$a" != "" ]; do
  if echo "$a" | grep -q "^[0-9].*$"
  then
    x=`expr $a + $x`
  fi
  read a
done
echo $x
```

a) Ce se va afișa, dacă fișierul `test` conține numere de la 0 la 5 pe linii consecutive?

b) Ce se va afișa, dacă fișierul `test` conține doar o linie cu numere de la 0 la 5, separate prin spațiu?

c) Ce se va afișa, dacă în fișierul `test` de la punctul (a) se adaugă o linie cu secvența `abc`?

d) Ce se va afișa, dacă în fișierul `test` de la punctul (a) se adaugă o linie cu secvența `6bc`?

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
VARIANTA 2

Subiect 1 (Algoritmică și Programare):

Oficiu – 1p
Definirea clasei Participant– 0.3p din care
- atribute – 0.1
- constructor – 0.1
- metode - 0.1

Definirea claselor Volunteer și Employee – 1.2p din care
- relația de moștenire – 0.25
- constructor – 0.25
- metode – 0.7

Definirea clasei ONG– 0.8p din care
- atribut – 0.1
- constructor – 0.1
- metoda getAll - 0.6

Funcția de la punctul b) – 2.2p din care
- signatura corectă - 0.1p
- sortare listă voluntari (alfabetic după oraș) folosind *Merge Sort* - 2p
- returnare listă rezultat – 0.1

Funcția de la punctul c) - 2p din care
- signatura corectă - 0.1p
- determinare oraș cu nr. maxim de participanți – 1.8p
- returnare rezultat – 0.1

Funcția de la punctul d) – 1p din care
- signatura corectă - 0.1p
- determinare număr total angajați din lista de ONG-uri– 0.8p
- returnare rezultat – 0.1p

Funcția principală e) – 0.5p
f) Specificațiile operațiilor folosite pentru tipul de dată Listă– 1p
(*la* f) *punctajul se va acorda proporțional cu numărul de cerințe din enunț rezolvate corect*)

Subiect 2 (Baze de date)

1. 0.5p (chei primare) + 0.5p (chei externe) = 1p
2. 1p
3. a, c

   2 x (0.5p răspuns + 0.5p justificare) = 2p
4. rezolvarea completă a interogării = 2p
5. rezolvarea completă a interogării = 3p

1p of

Subiect 3 (Sisteme de operare):

3.1.a - diagrama cu un părinte și mai mulți fii - 0.5p
- 3 procese fiu - 0.5p

3.1.b - a b c d e f g h i j - 0.5p
- în ordine nedeterminabilă - 0.5p

3.1.c - fiii nu închid capetele de scriere în pipe și read e blocat la pipe gol - 1p
3.1.d - între liniile 6 și 7 - 1p
3.1.e - așteaptă terminarea tuturor fiilor - 1p

3.2.a - 15 (suma de la 0 la 5) - 1p
3.2.b - expr dă eroare (0 + 0 1 2 3 4 5) - 1p
3.2.c - 15 (suma de la 0 la 5), linia abc e eliminată de if - 1p
3.2.d - expr dă eroare (6abc începe cu cifră dar nu e număr) - 1p