UNIVERSITATEA BABEŞ-BOLYAI

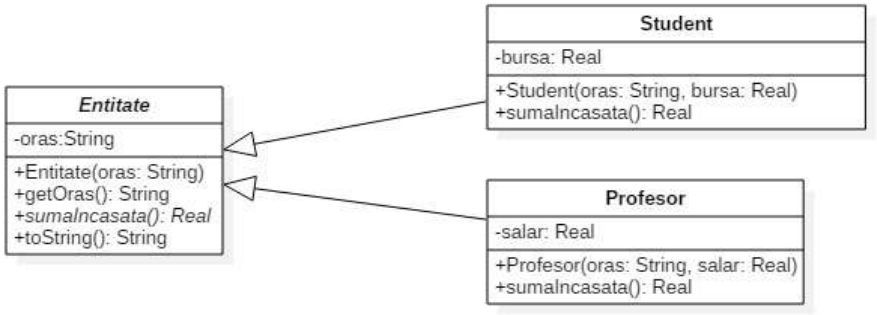
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Examen de licență septembrie 2016
Specializarea Informatică Română

Subiectul 1

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

(a) **Defineşte clasele** *Entitate*, *Student* şi *Profesor* pe baza următoarei diagrame UML.



- *Oraş* trebuie să fie nenul şi nevid, iar *bursa* şi *salar* trebuie să fie strict pozitive. Constructorii trebuie să impună constrângerile.
- Metoda *sumaIncasata()* returnează suma pe care o încasează fiecare entitate (valoarea *bursei* pentru *Student*, respectiv *salarul* pentru *Profesor*).
- Metoda *toString()* din clasa *Entitate* returnează un şir de caractere conținând *oraş* concatenat cu *suma incasată*.

(b) **Defineşte o funcție** care creează şi returnează o listă formată din patru *Entități*: doi *Studenți* şi doi *Profesori*.

(c) **Defineşte o funcție** care primeşte ca parametru o listă de *Entități* şi o rearanjează astfel încât la începutul listei să fie *Profesorii*, în ordine descrescătoare a *sumei încasate*, după care vor urma *Studenții* în ordine crescătoare a *sumei încasate*.

(d) **Defineşte o funcție** care primeşte ca parametri o listă de *Entități* şi un *oraş* şi returnează *suma totală încasată* de *Entitățile* din acel oraş.

(e) **Funcția principală** a programului apelează funcțiile indicate la punctele (b), (c), apoi citeşte numele unui oraş şi afişează rezultatul returnat de funcția de la (d).

(f) Pentru tipurile de date utilizate în program, scrieți specificațiile operațiilor folosite.

Notă

- **Nu se vor folosi containere sortate şi metode 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

a. Creați o bază de date relațională, cu toate tabelele în 3NF, ce va reține următoarele informații pentru circuitul profesionist de tenis feminin:

jucătoare de tenis: cod jucătoare, nume, poziția curentă în topul mondial, țară;

turnee: cod turneu, nume, localitate, cod tip turneu, denumire tip turneu (valori: *Grand Slam*, *PremierMandatory*, *Premier 5* etc), cod suprafață de joc, denumire suprafață de joc (valori: *zgură*, *iarbă*, *suprafață dură*) şi o listă cu numărul de partide câştigate într-un an de fiecare jucătoare participantă la acel turneu (dacă o jucătoare nu a participat într-un anumit an, ea nu va apărea în listă).

Exemplu de informații care se pot reține pentru turneul *Roland Garros* (pentru simplitate am ignorat codurile): (Roland Garros, Paris, Grand Slam, zgură, ((Garbine Muguruza, 2016, 7), (Serena Williams, 2016, 6), (Simona Halep, 2016, 3), (Simona Halep, 2015, 0), …))

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

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

i) Turneele *Grand Slam* (denumire turneu, an) la care jucătoare din afara topului 10 mondial au câștigat 7 partide într-un an.

ii) Jucătoarele (nume și țară) care în 2016 au câștigat cel puțin o partidă la un turneu desfășurat pe zgură, dar nu au participat la niciun turneu pe iarbă.

iii) Jucătoarele (nume, țară, număr total de victorii) care au cel mai mare număr de victorii.

Subiectul 3

3.1 Considerând că toate instrucțiunile din fragmentul de cod de mai jos se execută cu succes, răspundeți la următoarele întrebări:

```
1  int main() {
2      int pfd[2], i, n;
3      pipe(pfd);
4      for(i=0; i<3; i++) {
5          if(fork() == 0) {
6              write(pfd[1], &i, sizeof(int));
7              close(pfd[0]); close(pfd[1]);
8              exit(0);
9          }
10         else {
11             // a se vedea punctele c) si d)
12         }
13     }
14     for(i=0; i<3; i++) {
15         wait(0);
16         read(pfd[0], &n, sizeof(int));
17         printf("%d\n", n);
18     }
19     close(pfd[0]); close(pfd[1]);
20     return 0;
21 }
```

a) Ce va tipări rularea codului așa cum este?

b) Câte procese se creează, incluzând procesul inițial, dacă lipsește linia 8? Specificați relația părinte fiu dintre aceste procese.

c) Câte procese se creează, incluzând procesul inițial, dacă mutăm instrucțiunea de pe linia 8 pe linia 11 (pornind de la codul dat)? Specificați relația părinte fiu dintre aceste procese.

d) Ce va tipări rularea codului, dacă liniile 16 și 17 se mută în interiorul ramurii else, începând cu linia 11 a codului inițial? Justificați răspunsul.

3.2 Considerând că directorul **DIR** conține o ierarhie de subdirectoare și fișiere text răspundeți la următoarele întrebări despre scriptul Shell UNIX de mai jos.

```
1  for f in `find DIR -type f`; do
2      if grep -q "^[^0-9]" $f; then
3          echo $f >> 1.txt
4      fi
5      if ! grep -q "[a-z]$" $f; then
6          echo $f > 2.txt
7      fi
8  done
```

a) Ce va conține fișierul 1.txt după rularea scriptului?

b) Ce va conține fișierul 2.txt după rularea scriptului?

c) Explicați în detaliu expresiile regulare de pe liniile 2 și 5.

Notă: Toate subiectele sunt obligatorii. Fiecare subiect se notează între 1 şi 10 de către ambii corectori.

Timp de lucru: 3 ore.

BAREM
INFORMATICĂ

Subiect 1 (Algoritmică şi Programare):

Oficiu – **1**p

Definirea clasei *Entitate*– **0.7**p din care
- atribut – 0.1
- constructor – 0.2
- metode - 0.4

Definirea clasei *Student*– **0.7**p din care
- relația de moștenire – 0.1
- constructor – 0.5
- metoda *sumaIncasata*() – 0.1

Definirea clasei *Profesor*– **0.7**p din care
- relația de moștenire – 0.1
- constructor – 0.5
- metoda *sumaIncasata*() – 0.1

Funcția de la punctul b) – **0.3**p din care
- signatura corectă - 0.1p
- creare obiecte – 0.1p
- adaugare în listă – 0.1p

Funcția de la punctul c) – **2.5**p din care
- signatura corectă - 0.1p
- rearanjarea listei – 2.4p

Funcția de la punctul d) – **2.2**p din care
- signatura – 0.1p
- determinare sumă încasată – 2p
- returnare rezultat – 0.1p

Funcția de la punctul e) – **0.4p**

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

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 puncte** pentru b1
- **1.5 puncte** pentru b2
- **2.5 puncte** pentru b3

Subiect 3 (Sisteme de operare):

Oficiu: 1p

3.1
- a) 0, 1, 2 pe linii separate în orice ordine 1p
- b) 8 procese, arbore cu 8 procese 2p
- c) 4 procese, arbore cu 4 procese 1p
- d) 0, 1, 2 pe linii separate întodeauna în această ordine 1p

3.2
- a) Căile (relative la DIR) ale tuturor fișierelor care conțin linii care nu încep cu cifră 1p
- b) Calea (relativă la DIR) a ultimului fișier găsit de comanda find care nu conține linii terminându-se cu literă mică 1p

c) Linia 2: început de linie, interval de cifre negat 1p
Linia 5: interval de litere mici, sfârșit de linie 1p