UNIVERSITATEA BABEŞ-BOLYAI

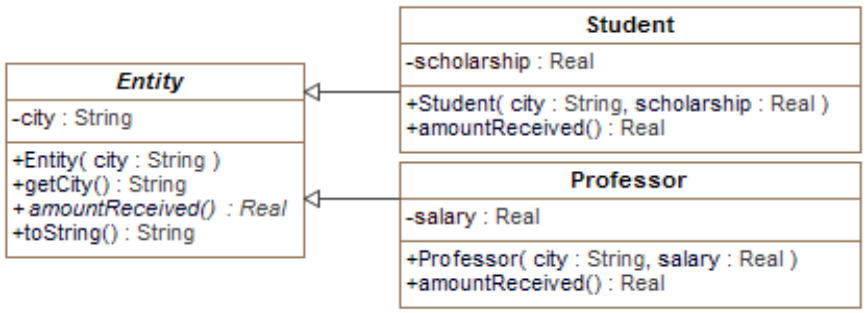
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Bachelor Degree Exam, September 2016
Computer Science - English

Subject 1

Write a program in one of the programming languages Python, C++, Java, C# that:

(a) **Defines the classes** *Entity*, *Student*, and *Professor* according to the following UML class diagram:



- The *city* must be not null and non-empty, and *scholarship* and *salary* must be strictly greater than zero. The constructors must enforce the constraints.
- The *amountReceived*() method returns the amount received by the entity (*scholarship* for *Student*, and *salary* for *Professor*).
- The *toString()* method from the *Entity* class returns a string that contains the *city* concatenated with the *amount received* by that entity.

(b) **Defines a function** that creates and returns a list of four *Entities*, two *Students* and two *Professors*.

(c) **Defines a function** that has as parameter a list of *Entities* and rearranges it such that the *Professors* are at the beginning of the list, ordered descendingly by the *amount received*, followed by the *Students*, ordered ascendingly by the *amount received*.

(d) **Defines a function** that has as parameters a list of *Entities* and a *city*, and returns the total *amount received* by the *Entities* belonging to that city.

(e) **The *main* function** of the program calls the functions indicated at (b), (c), then reads the name of a *city* and prints the result returned by the function at (d).

(f) For the data structures used by your program, write the specifications of the used operations.

Remarks

- **Do not use sorted containers and predefined sorting operations.**
- **Do not define other methods than those specified in the subject.**
  You can use existing libraries for data structures (Python, C++, Java, C#).

Subject 2

a. Create a relational database with all the tables in 3NF. The database will store the following information for the women's professional tennis circuit:

tennis players: player id, name, current rank in the world, country;

tournaments: tournament id, name, city, tournament type id, tournament type name (values: *Grand Slam*, *Premier Mandatory*, *Premier 5*, etc), court surface id, court surface name (values: *clay*, *grass*, *hard*), and a list with the number of matches won in a year by every player that participated in the tournament (if a player didn't participate in a certain year, she won't appear in

the list).

Examples of information one can store for the *Roland Garros* tournament (for simplicity, the ids are ignored): (Roland Garros, Paris, Grand Slam, clay, ((Garbine Muguruza, 2016, 7), (Serena Williams, 2016, 6), (Simona Halep, 2016, 3), (Simona Halep, 2015, 0), …)).

Prove the created database is in 3NF, by identifying the functional dependencies.

b. For the database created for **a**, solve, using the relational algebra **or** SQL, the following queries:

i) The *Grand Slam* tournaments (tournament name, year) in which players outside of the world top 10 won 7 matches in a year.

ii) The players (name and country) who, in 2016, won at least one match in a clay tournament, but didn't participate in any grass tournament.

iii) The players (name, country, total number of victories) with the greatest number of victories.

Subject 3

3.1 Answer the following questions considering that all instructions are executed successfully.

```
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             // see questions c) and 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) What will display the execution of the code as it is?

b) How many processes will be created, including the initial process, if line 8 is missing? Specify the parent/child relationship between these processes.

c) How many processes will be created, including the initial process, if line 8 is moved to line 11 (in the given code)? Specify the parent/child relationship between these processes.

d) What will display the execution of the code, if lines 16 and 17 are moved inside the else branch, starting with line 11 of the initial code? Justify your answer.

3.2 Answer the following questions about the UNIX Shell script below, considering that directory **DIR** contains a hierarchy of subdirectories and text files.

```
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;
6  then
7          echo $f > 2.txt
8      fi
   done
```

a) What will file 1.txt contain after the script is executed?

b) What will file 2.txt contain after the script is executed?

c) Explain in detail the regular expressions on lines 2 and 5.

Remarks: All subjects are compulsory. Each subject will be graded with a mark between 1 and 10 by both evaluators.

Time limit: 3 hours

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 fişierelor care conţin 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