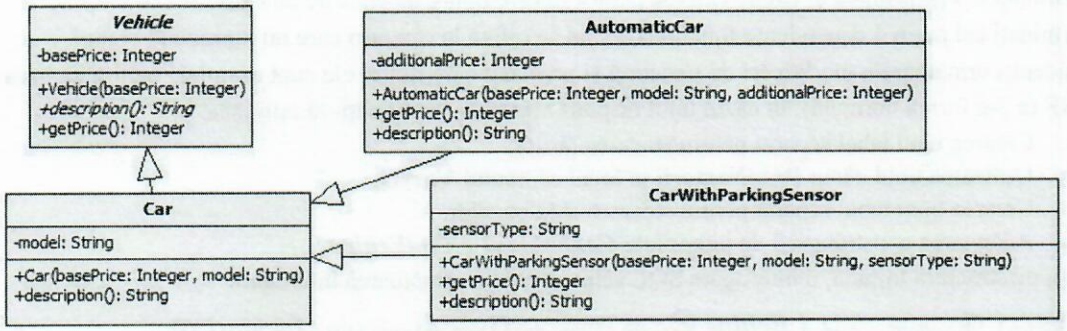
UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Proba scrisă a examenului de licenţă, 2 iulie 2018
Informatică Română
VARIANTA 1

SUBIECTUL 1. Algoritmică şi programare

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

a) **Defineşte** clasele ***Vehicle*** (abstractă), **Car**, **AutomaticCar** şi **CarWithParkingSensor** pe baza următoarei diagrame UML:



- *basePrice* (preţ de bază) şi *additionalPrice* (preţ adiţional) trebuie să fie o valoare strict pozitivă, iar *sensorType* (tip senzor) şi *model* (modelul maşinii) trebuie să fie nenule şi nevide. Constructorii trebuie să impună constrângerile.
- Metoda **getPrice()** din clasa ***Vehicle*** returnează preţul de bază al vehiculului, metoda **getPrice()** din clasa **AutomaticCar** returnează preţul adiţional însumat cu preţul de bază al maşinii, iar metoda **getPrice()** din clasa **CarWithParkingSensor** returnează preţul de bază al maşinii însumat cu 2500.
- Metoda **description()** din clasa **Car** returnează modelul maşinii, metoda **description()** din clasa **AutomaticCar** returnează textul "Automatic car " concatenat cu modelul maşinii, iar metoda **description()** din clasa **CarWithParkingSensor** returnează textul "Car with parking sensor " concatenat cu tipul senzorului, " " şi modelul maşinii.

b) **Defineşte o funcţie** care, având ca parametru o listă L de obiecte de tip ***Vehicle***, returnează o listă cu perechi de forma <*model*, *nrCars*>, conţinând pentru fiecare *model* de maşină din lista L numărul de masini *nrCars* cu acel model. În lista returnată ca rezultat, fiecare model de maşină din lista L va apărea o singură dată.

c) **Defineşte o funcţie** care, având ca parametru o listă de obiecte de tip ***Vehicle***, rearanjează lista astfel încât toate maşinile având preţul cuprins în intervalul [1000, 2000] să apară în listă înaintea celor având preţul mai mic decât 1000 sau mai mare decât 4000, păstrând ordinea obiectelor din lista iniţială. Nu se vor folosi liste sau alte structuri auxiliare.

d) **Defineşte o funcţie** care, având ca parametru o listă de obiecte de tip ***Vehicle***, afişează descrierea maşinilor din listă.

e) **Funcţia principală** a programului creează o listă cu următoarele maşini (alegeţi valori pentru proprietăţile lor neprecizate): un Audi, un Audi cu schimbător automat, o Toyota, un Mercedes cu schimbător automat şi un Opel cu senzor de parcare. Pentru lista construită anterior, apelaţi funcţia de la b) şi afişaţi lista returnată. Apelaţi funcţia de la c) şi apoi afişaţi lista rezultată folosind funcţia de la punctul d).

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

Notă

- **Se va indica limbajul de programare folosit.**
- **Nu se vor folosi containere sortate şi operaţii de sortare predefinite.**
- **Nu se vor defini alte metode decât cele specificate în enunţ.**

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

SUBIECTUL 2. Baze de date

Fie o bază de date care stochează rezultatele obținute de echipele naționale de fotbal în toate meciurile disputate (o echipă poate juca un singur meci la o anumită dată). Baza de date are următoarea structură:

- tabelul ***Echipe*** cu câmpurile **CodE**, **Tara**, **Continent**;
- tabelul ***Jucatori*** cu câmpurile **CodJ**, **CodEchipa**, **Nume**, **Varsta**, **NumarMeciuri**;
- tabelul ***Meciuri*** cu câmpurile **CodEchipa1**, **GoluriEchipa1**, **CodEchipa2**, **GoluriEchipa2**, **Data**, **Stadion**, **NumarLocuri**, **Oras**, **Tara**.

1. Determinați cheile primare și cheile externe pentru fiecare dintre tabelele de mai sus.
2. Determinați cel puțin 4 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 **DataNasterii** în locul câmpului **Varsta.**
   c. Crearea unui tabel separat pentru stocarea stadioanelor.
   d. Adăugarea constrângerii de integritate ***CodEchipa1*** > ***CodEchipa2.***
4. Scrieți, pe structura inițială, o interogare SQL echivalentă cu următoarea interogare

$$\Pi_{\text{Nume}} (\sigma_{\text{Continent = 'Asia'}} (\textbf{\textit{Echipe}} \otimes_{\text{CodE = CodEchipa}} \sigma_{\text{NumarMeciuri >100}} (\textbf{\textit{Jucatori}})))$$

5. Scrieți o interogare SQL care returnează numărul total de meciuri jucate de fiecare echipă națională (***Tara, NrMeciuri***).

SUBIECTUL 3. Sisteme de operare

3.1 Fragmentul de cod de mai jos este compilat cu succes în executabilul `pr`. Argumentul `w` al funcției `f` specifică FIFO-ul în care se va scrie: 0 pentru `a` și 1 pentru `b`. Răspundeți la următoarele întrebări, considerând că: toate instrucțiunile se execută cu succes, toate FIFO-urile necesare sunt șterse și create din nou înaintea fiecărei execuții și `pr` este singurul program care accesează aceste FIFO-uri.

```
void f(char* a, char* b, int w, char* s) {
  int f[2], r=1-w; char c;
  if(fork() == 0) {
    f[0] = open(a, w==0 ? O_WRONLY : O_RDONLY);
    f[1] = open(b, w==1 ? O_WRONLY : O_RDONLY);
    write(f[w], s, 1);
    read(f[r], &c, 1);
    printf("%c\n", c);
    close(f[0]); close(f[1]);
    exit(0);
  }
}
int main(int n, char** a) {
  int i;
  for(i=1; i<n; i+=4) {
    f(a[i], a[i+1], a[i+2][0]-'0', a[i+3]);
  }
  for(i=1; i<n; i+=4) {wait(0);}
  return 0;
}
```

a) Desenați diagrama ierarhiei de procese pentru o execuție în care `pr` primește `4*K` argumente în linia de comandă.

b) Ce va tipări următoarea execuție? Justificați răspunsul.

```
./pr p q 1 x
```

c) Ce va tipări următoarea execuție? Justificați răspunsul.

```
./pr p q 1 x p q 0 y
```

d) Desenați o diagramă care ilustrează procesele fiu și operațiile lor de read/write asupra FIFO-urilor, în execuția de la (c).

e) Ce va tipări următoarea execuție? Justificați răspunsul.

```
./pr p q 1 x q p 1 y
```

3.2 Comanda `sed "s/A/B/"` înlocuiește prima apariție de pe fiecare linie a expresiei regulare `A` cu stringul `B`, substituind orice referință `\N` din `B` cu conținutul celei de-a `N`-a expresii între paranteze din `A`. Care sunt rezultatele afișate de scriptul Shell UNIX de mai jos, când e rulat într-un director conținând fișiere sursă și header C/C++? Explicați în detaliu linia 3: comenzi, argumente și pipe.

```
1 for F in *.c *.cpp *.h; do
2   if [ -f $F ]; then
3     grep "#include.*<" $F | sed "s/^.*<\(.*\)>.*$/\1/"
4   fi
5 done | sort
```

NOTĂ.

- Toate subiectele sunt obligatorii. La toate subiectele se cer rezolvări cu soluții complete.
- Nota minimă ce asigură promovarea este 5.00

BAREM INFORMATICĂ
VARIANTA 1

Subiect 1 (Algoritmică și Programare):

Oficiu – 1p
Definirea clasei abstracte Vehicle – 0.3p din care
- atribut – 0.1
- metode - 0.2

Definirea clasei Car – 0.3p din care
- relația de moștenire – 0.1
- atribut – 0.1
- metode - 0.1

Definirea clasei AutomaticCar– 1p din care
- relația de moștenire – 0.15
- atribut – 0.15
- constructor – 0.35
- metode – 0.35

Definirea clasei CarWithParkingSensor – 1p din care
- relația de moștenire – 0.15
- atribut – 0.15
- constructor – 0.35
- metode – 0.35

Funcția de la punctul b) – 2.25p din care
- signatura corectă - 0.1p
- construire listă perechi conținând modele distincte- 2p
- returnare listă rezultat – 0.15

Funcția de la punctul c) – 2.25p din care
- signatura corectă - 0.1p
- rearanjare listă – 2.15p

Funcția de la punctul d) – 0.4p din care
- signatura corectă - 0.1p
- afișare descrierei mașini din listă – 0.3p

Funcția principală e) – 0.5p
f) Specificațiile operațiilor folosite pentru tipul de dată Listă– 1p

Subiect 2 (Baze de date)

1. 0.5p (chei primare) + 0.5p (chei externe) = 1p
2. 0.25p x 4 = 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 mulți fii - 0.5p
- K procese fiu - 0.5p

3.1.b - nimic - 0.5p
- se blochează open la deschiderea FIFO-ului - 0.5p

3.1.c - x si y - 0.5p
- în ordine nedeterminabilă - 0.5p

3.1.d - diagrama circulară cu două procese și două FIFO-uri - 1p

3.1.e - nimic - 0.5p
- se blochează open-urile și creează deadlock - 0.5p

3.2 Rezultate - lista sortată a fișierelor header sistem incluse în surse - 1p

3.2 Linia 3 - grep: extrage liniile care includ fișiere header sistem - 0.5p
- grep: detaliere expresie regulară - 0.5p
- pipe: transmite output-ul lui grep ca input lui sed - 0.5p
- sed: pastrează pe linie doar numele fișierului header - 1p
- sed: detaliere comandă și expresie regulară – 0.5p