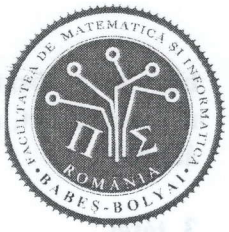
BABEŞ-BOLYAI TUDOMÁNYEGYETEM

MATEMATIKA ÉS INFORMATIKA KAR



Záróvizsga – 2014 július
Magyar informatika szak

1. tétel

Írjunk egy programot (Java, C++, C# vagy Python nyelven), mely:

a. Definiál egy *Diák* osztályt, mely tartalmaz egy *Név* karaktersor típusú privát tagot (attribútumot), egy publikus *konstruktort*, mely a nevet inicializálja, illetve egy publikus *toString()* metódust, mely a diák nevét téríti vissza.

b. Definiál egy *ÖsztöndíjasDiák* osztályt, melyet a *Diák* osztályból származtatunk, tartalmaz egy *Ösztöndíj* egész típusú privát tagot, egy publikus *konstruktort,* mely a nevet és az ösztöndíj értékét inicializálja, valamint egy túlterhelt (overloaded, felüldefiniált) publikus *toString()* metódust, mely az ős metódus visszatérített értékéhez hozzáfűzi az ösztöndíj értékének sztring-jét.

c. Kijelent egy függvényt, mely felépít egy *asszociatív tömb*öt (szótár, dictionary), mely az alábbi értékeket tartalmazza: egy *Diák* típusú objektumot „Ionescu" névvel; egy *ÖsztöndíjasDiák* típusú objektumot „Popescu" névvel és 200 értékű ösztöndíjjal. A kulcs legyen a név, az érték pedig az objektum.

d. Definiál egy kétparaméteres függvényt, melynek egyik bemenete a **c.** pontban alkotott asszociatív tömb, a második pedig egy karakter. A függvény ellenőrzi, hogy létezik-e az asszociatív listában érték, melynek kulcsa a specifikált karakterrel kezdődik.

e. Írjuk meg a főprogramot, mely (1) a **c.**-beli asszociatív tömböt építi; (2) beolvas egy karaktersort a standard bemenetről; (3) kiírja az összes objektumot, melynek kulcsa megegyezik a beolvasott karaktersorral. Amennyiben a keresés sikertelen, a „nem találtam" üzenet jelenik meg; (4) legvégül a program – a (d.) pontban írt függvény segítségével – ellenőrzi, hogy van-e az asszociatív tömbben a karaktersor első karakterével kezdődő elem, és keresés eredményét megjeleníti.

f. Specifikáljuk az asszociációs tömb feldolgozására használt operátorokat és műveleteket.

A megoldáshoz használhatunk létező programkönyvtárakat (Python, C++, Java , vagy C# programnyelven). Amennyiben nem használunk létező programokat, specifikáljuk ebben az esetben is a használt metódusokat.

2. tétel

a. Írjuk fel a funkcionális függőségeket egy egyetemi kar következő infomációira:

- **ösztöndíjak diákoknak**: ösztöndíj-azonosító (egyedi), megnevezés, leírás, az ösztöndíjat ajánló cég/szervezet (név, weboldal), felajánlott helyek száma, ösztöndíj hónapjainak a száma, egy havi ösztöndíj összege, pályázó diákok listája;
- **diákok**: személyi szám (egyedi), név, csoport, évfolyam, szak, jegy (amivel minden ösztöndíjra pályázik), megpályázott ösztöndíjak listája (a kívánt sorrendben)

Tervezzünk **relációs adatbázis sémát** a fenti információknak, melynek táblái **3NF**-ban vannak, indokoljuk, hogy a végső táblák 3NF-ban vannak.

b. Relációs algebrát **vagy** SELECT-SQL parancsot használva, az **a.** pont adatbázisára vonatkozóan adjuk meg:

b1. Azokat a diákokat (név, szak, jegy), akik pályáznak "BM" azonosítójú ösztöndíjra és nem pályáznak „BT" azonosítóval rendelkezőre.

b2. Azon ösztöndíjak esetén, melyekre a legtöbben feliratkoztak, adjuk meg az ösztöndíj megnevezését, ajánló cég nevét, a feliratkozott diákok számát és az ösztöndíjra pályázók jegyeinek átlagát.

3. tétel

a) Adott a következő Linux C program. A végrehajtható program neve **p.**

```
1  #include <unistd.h>
2  #include <stdio.h>
3  int p[2];
4  pid_t mypid;
5  main (int argc, char *argv[]){
6    pipe (p);
7    if (fork () == 0){
8         close (p[0]);
9         if (argv[1][0] == 'c'){
10             execlp (argv[2], argv[2], 0);
11             printf ("Exec finished\n");
12        }
13        mypid = getpid ();
14        printf ("p=%d: pp=%d\n", mypid, getppid ());
15        write (p[1], &mypid, sizeof (int));
16        exit (0);
17   }
18   close (p[1]);
19   read (p[0], &mypid, sizeof (int));
20   printf("Pid parinte %d: pid fiu:%d\n",getpid(), mypid);
21 }
```

```
/home/scs/exam/p s ls

/home/scs/exam/p c pwd
```

a.1 Mit ír a képernyőre a 3. oszlopban adott két futtatás? Magyarázzuk meg az eredményt.
a.2 Mit ír a képernyőre a két futtatás, ha kihagyjuk a 16. sort (exit(0)). Magyarázzuk meg az eredményt.

b) Adott a következő shell program:

```
echo $#: $*
p=`echo $1|grep ^[^0-9]*[0-9]$`
if [ "$p" != ""  ]
then
    shift
    ./s.sh  $*
fi
```

```
./s.sh 1 2 3
./s.sh 1 22 3
./s.sh a1 2
./s.sh 1 2a 3
```

b1. Magyarázzuk meg a 2. sor működését.
b2. Mit ír a képernyőre a 2. oszlopban adott négy futtatás? Magyarázzuk meg az eredményt.

Megjegyzés: Az összes tétel kötelező.
Minden tételt 1-től 10-ig osztályoz mindkét javító.
Munkaidő: 3 óra.