

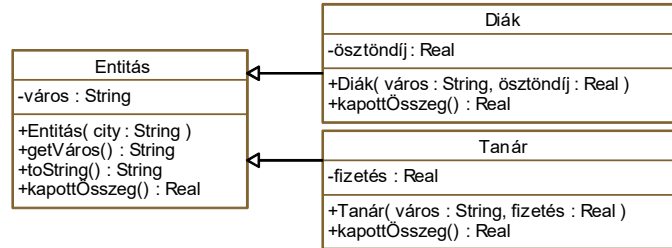


## Záróvizsga 2016. szeptember Informatika Szak

### 1. tétel

Írjunk egy programot a C++, C#, Java, vagy Python programnyelvek egyikében, mely:

- a) **Definiálja** az alábbi ULM-diagram alapján az *Entitás*, *Diák* és *Tanár* osztályokat:



- A *város* mező nem lehet null és nem lehet üres; az *ösztöndíj* és a *fizetés* mezők szigorúan pozitívek; a megkötéseket a konstruktorokban ellenőrizzük.
  - A *kapottÖsszeg()* metódus az entitások által kapott összeget téríti vissza (az *ösztöndíjat* a *Diák*, a *fizetést* a *Tanár* számára);
  - A *toString()* metódus egy karaktersort térít vissza, mely a *város* és a *kapott összeg* összefűzése.
- b) **Definiál egy függvényt**, mely létrehoz és visszatérít egy négy elemet tartalmazó *Entitás-listát*, mely két *Tanárt* és két *Diákot* tartalmaz,
- c) **Definiál egy függvényt**, melynek bemenő paramétere egy *Entitás-lista*, melyet a függvény átrendez a következőképpen: előbb a *Tanárok* jelennek meg a *kapott összeg* szerinti *csökkenő* sorrendben, ezután a *Diákok* a *kapott összeg* szerint *növekvő* sorrendben.
- d) **Definiál egy függvényt**, melynek bemenő paramétere egy *Entitás-lista* és egy *város*; visszatéríti a bemenő paraméterként adott *város entitásainak teljes jövedelmét*.
- e) **Definiálja a főprogramot**, mely meghívja egymás után a (b) és (c) pontok függvényeit, majd beolvas egy városnevet és kiírja a (d) függvény által visszatérített eredményt.
- f) A program során használt típusok műveleteit specifikáljuk.

### Megjegyzés:

- NE használjunk rendezett konténereket vagy predefiniált rendező függvényeket.
- NE definiáljunk más metódusokat.

Az adattípusok számára használhatunk létező könyvtárakat (C++, C#, Java, Python).

### 2. tétel

- a) Tervezzünk **relációs adatbázis sémát**, melynek táblái **3NF**-ben vannak és a női teniszeversenyek következő információit tartalmazzák:

- **teniszezönök**: kód, név, ország, helyezés a világranglistán;
- **tornák**: torna kód, név, város, torna-típus kód, torna-típus név (például: Grand Slam, Premier Mandatory, Premier 5, stb.), pálya-típus kód, pálya-típus név (például: füves, salakos, kemény, stb.), illetve a **résztevők listája**, melynek elemei egyenként tartalmazzák a teniszezőnt, a részvétel évét, a nyert meccsek számát a tornán (ha egy teniszező nem vett részt egy évben a tornán, nem jelenik meg). Egy példa a tornára a Roland Garros (az egyszerűség kedvéért a kódok nincsenek feltüntetve): (Roland Garros, Paris, Grand Slam, salak, ( (Garbine Muguruza, 2016, 7), (Serena Williams, 2016, 6), (Simona Halep, 2016, 3), (Simona Halep, 2015, 0), ... )).

A funkcionális függőségeket megadva, indokoljuk meg, hogy a táblák 3NF-ben vannak.

- b) Az a) pont adatbázisára vonatkozóan, SQL parancsot vagy relációs algebrát használva, adjuk meg:
- Azokat a Grand Slam típusú tornákat (név, év), ahol 7 meccset nyertek olyan teniszezők, akik nincsenek a világranglista első 10 helyezettje között.
  - Azokat a teniszezőket (név, ország), akik 2016-ban nyertek legalább egy meccset salakos pályán, de nem vettek részt füves pályás tornán ugyanebben az évben.
  - Azokat a teniszezőket (név, ország, nyert meccsek száma), akiknek összességében a legtöbb nyert meccsük van.

### 3. tétel

3.1 Feltéve, hogy az alábbi, C-ben írt kódrészlet minden parancsa hiba nélkül hajtódik végre, válaszoljunk az alábbi kérdésekre:

<pre> 1  int main() { 2      int pfd[2], i, n; 3      pipe(pfd); 4      for(i=0; i&lt;3; i++) { 5          if(fork() == 0) { 6              write(pfd[1], &amp;i, sizeof(int)); 7              close(pfd[0]); close(pfd[1]); 8              exit(0); 9          } 10         else { 11             // lasd a c) es d) alpontokat 12         } 13     } 14     for(i=0; i&lt;3; i++) { 15         wait(0); 16         read(pfd[0], &amp;n, sizeof(int)); 17         printf("%d\n", n); 18     } 19     close(pfd[0]); close(pfd[1]); 20     return 0; 21 }</pre>	<p>a) Mi lesz a kimenet, ha a megadott kódot futtatjuk?</p> <p>b) Hány folyamat jön létre, az eredeti folyamatot is beleértve, amennyiben hiányzik a 8. sor? Adjuk meg a szülő-gyerek viszonyt az így létrejött folyamatok között.</p> <p>c) Hány folyamat jön létre, az eredeti folyamatot is beleértve, amennyiben a 8. sorban levő utasítást áthelyezzük a 11. sorba (a kezdeti kódból kiindulva)? Adjuk meg a szülő-gyerek viszonyt az így létrejött folyamatok között.</p> <p>d) Mi lesz a kimenet, ha a 16. és 17. sort áthelyezzük az <code>else</code> ágba, az eredeti kód 11. sorától kezdődően? Indokoljuk meg a választ.</p>
---	--

3.2 Válaszoljunk az alábbi kérdésekre a megadott UNIX shell szkripttel kapcsolatban, feltételezve, hogy a **DIR** katalógus alkatalógusok és szöveges állományok hierarchiáját tartalmazza:

<pre> 1  for f in `find DIR -type f`; do 2      if grep -q "^[^0-9]" \$f; then 3          echo \$f &gt;&gt; 1.txt 4      fi 5      if ! grep -q "[a-z]\$" \$f; then 6          echo \$f &gt; 2.txt 7      fi 8  done</pre>	<p>a) Mit fog tartalmazni az 1.txt állomány a szkript futtatását követően?</p> <p>b) Mit fog tartalmazni a 2.txt állomány a szkript futtatását követően?</p> <p>c) Magyarázzuk meg részletesen a második, illetve az ötödik sorban szereplő reguláris kifejezéseket.</p>
--	--

Megjegyzés: Minden tétel kötelező. Egyes tételeket egytől tízig osztályozunk két javító által.

Munkaidő három óra.