Programozás I. Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2023. tavasz

Általános követelmények, tudnivalók

- A feladat elkészítésére 90 perc áll a rendelkezésre. Ez szigorú határidő, a Bíró előre megadott időben zár.
- A feladatokat számítógép előtt kell megoldani, tetszőleges fejlesztői környezetben, tetszőleges operációs rendszer segítségével.
- Az elkészült programot **20** alkalommal lehet benyújtani, a megadott határidőig.
- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz ont.
  - Aki Windowst használ, annak a gép elindítása után érdemes egyből a fejlesztőkörnyezetet elindítani, és létrehozni egy új projektet, és csak utána a böngészőt, mivel az elején egy néhány percig indexel, addig pont el lehet olvasni a feladatot.
- Bármely segédanyag használata **tilos** (a fejlesztőkörnyezetek nyújtotta segítségen kívül), aki mégis ilyet tesz, vagy próbálkozik vele, annak a dolgozata nem értékelhető és a kurzus nem teljesített. Ha valakinek a padtársa segít, akkor mérlegelés nélkül mindkettő hallgató dolgozata sikertelen, a kurzus nem teljesített.
- A feladat megoldása során minden megadott előírást pontosan követni kell! Tehát, ha a feladat leírása szerint egy adattag neve a "elsoFoku", akkor az alábbi elnevezések nem megfelelőek: "elsőFokú", "elsofoku", "elso_foku", "elsőFoq". Ugyanez igaz a metódusok, osztályok elnevezésére is!
- A metódusok esetében a visszatérési típus, a név, módosítók és a paraméterek típusai (és azok sorrendje) kerülnek ellenőrzésre, azonban a paraméterek nevei tetszőlegesek lehetnek.
- Az órán tanult konvenciókat követni kell (getter/setter elnevezés, toString, indentálás, stb). Abban az esetben is, ha ezt a feladat külön nem emeli ki, az ellenőrzés során erre is építünk.
- A nem forduló kódok nem kerülnek kiértékelésre, ezt utólagosan a gyakorlatvezető sem bírálhatja felül. (Hiszen mindenki rendelkezésére áll a saját környezete, ahol fordítani, futtatni tudja a forráskódot, így feltöltés előtt ezt mindenképpen érdemes megnézni!)
- Az adattagok és konstruktorok hiányában garantáltan 0 pontos lesz a kiértékelés, mert ezek minden teszt alapját képezik.
- Ha végtelen ciklus van a programban, akkor ezt a Bíró ki fogja dobni 3 másodperc után (ha többször is meghívásra kerül ilyen metódus, akkor ez többszöri 3 másodperc, összesen akár 2 perc is lehet). Ilyenkor NE kattints még egyszer a *Feltöltés* gombra, mert akkor

kifagyhat a Bíró, csak a böngésző újraindításával lehet megoldani a problémát (emellett elveszik 1 feltöltési lehetőség is).

- Kérdés/probléma esetén a gyakorlatvezetők tudnak segítséget nyújtani.
- **A feladat megoldása során a default csomagba dolgozz, majd a kész forrásfájlokat tömörítve, zip formátumban töltsd fel, azonban a zip fájlt tetszőlegesen elnevezheted!**
  - Zip készítése: Windowson és Linuxon is lehet a GUI-ban jobb klikkes módszerrel tömörített állományt létrehozni (Windowsban pl. a 7-Zip nevű ingyenes program használatával).
  - Linux terminálon belül például a "zip feladat.zip *.java" paranccsal is elkészíthető a megfelelő állomány.
- A feladatokban az alábbi dolgok az alapértelmezettek (**kivéve**, ha a feladat szövege mást mond)
  - az osztályok láthatósága publikus
  - az egész érték 32 bites
  - a lebegőpontos számok dupla pontosságúak
  - az olyan metódusok void visszatéréssel rendelkeznek, amelyeknél nincs specifikálva visszatérési típus.
  - a metódusok mindenki számára láthatóak
  - az adattagok csak az adott osztályban legyenek elérhetőek
- A *riport.txt* és a fordítási log fájlok megtekinthetőek az alábbi módon:
  1. Az *Eredmények megtekintése* felületen a vizsgálandó próba új lapon való megnyitása
  2. A kapott url formátuma: `https://biro.inf.u-szeged.hu/Hallg/IB204L-1/1/hXXXXXX/4/riport.txt`
  3. Az url-ből visszatörölve a 4-esig (`riport.txt` törlése) megkaphatók a 4-es próbálkozás adatai.
- Szövegek összehasonlításánál az egyezés a pontos egyezést jelenti, azaz ha kis-nagy betűben térnek el, akkor már nem tekinthetők egyenlőnek (pl. a "piros" != "Piros")
- A leírásokban bemutat példákban a stringek köré rakott idézőjelek nem részei az elvárt kimenetnek, azok csak a string határait jelölik. Például ha az szerepel, hogy a példa bemenetre az elvárt kimenet az, hogy `"3 alma"`, akkor az elvárt kimenet idézőjelek nélkül az `3 alma`, de a szóköz szükséges!
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).

Kosárlabda

Játékos (4 pont)

Készítsük el a `Jatekos` osztályt az ábrán látható adattagokkal és metódusokkal.

Csapat (7 pont)

Készítsük el a `Csapat` osztályt az ábrán látható adattagokkal és metódusokkal.

A `beolvas` metódus megnyitja a paraméterben kapott csv fájlt, melynek sorai egy-egy játékos jellemzőit tartalmazzák: *játékos neve;pozíciója;kora;magassága;súlya*. A metódus létrehozza a játékosokat, majd hozzáadja a `jatekosok` attribútum megfelelő kulcsához tartozó értékhez. A map kulcsai a lehetséges pozícók: *Starting Pitcher, First Baseman, Shortstop, Third Baseman, Designated Hitter, Catcher, Second Baseman, Relief Pitcher, Outfielder*

A `hianyzoPoziciok` metódus leellenőrzi, hogy egy csapat rendelkezik-e játékossal az összes pozícióra. A függvény a hiányzó pozíciók listájával tér vissza. Ha például egy csapatnak nincs játékosa a *Catcher* pozícióra, akkor egy egyelemű lista a visszatérés.

A `atlag` metódus végigmegy a játékosokon és kiszámolja a paraméterben kért átlagot. A `melyikAtlag` értéke *magassag* vagy *suly* lehet, minden egyéb érték esetén 0-val tér vissza.

Meccs (5 pont)

Készítsük el a `Meccs` osztályt az ábrán látható adattagokkal és metódusokkal.

A `gyoztes` metódus meghatározza a győztes csapatot, majd visszatér vele. A győztes csapat az, amelyikben átlagosan magasabb játékosok vannak, ha ugyanakkora a két csapat átlagmagassága, akkor a kisebb átlagsúlyú játékosokból álló csapat a nyertes, ha ez is megegyezik, akkor az *elsoCsapat*.

Liga (11 pont)

Készítsük el a `Liga` osztályt az ábrán látható adattagokkal és metódusokkal.

Az egyetlen konstruktora egy csv fájlokat tartalmazó könyvtárat vár paraméterül. A konstuktorban végigiterál a csv kiterjesztésű fájlokon és minden fájlhoz létrehoz egy csapatot. A csapat neve egyezzen meg a csv fájl nevével a kiterjesztés nélkül.

A `hianyosCsapatok` metódus gyűjtse egy listába azokat a csapatokat, melyeknek nincs minden pozícióra játékosuk.

A `csapatokKizarasa` metódus törölje a `csapatok` listából a hiányos csapatokat, ezek nem vehetnek részt a meccseken.

A `jatek` metódusban hozzuk létre a meccseket. A meccsek létrehozása a csapatok párosításával történik az első csapat játszik a második csapattal, a harmadik a negyedikkel, ... Minden meccs esetén a vesztes kikerül a csapatok közül, a győztes csapat pedig a lista végére kerül. A függvény a győztes csapattal térjen vissza.

A `eredmenyTabla` metódusban egy kimeneti .csv kerül létrehozásra a következő fejléccel: *ElsoCsapat;MasodikCsapat;GyoztesCsapat*, a fejléc után a játék során lejátszott összes meccs kiírásra kerül.

Jó munkát!

1. ábra. A feladathoz tartozó osztálydiagram