

Programozás I. Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2024. 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 pont.
 - 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 ZH nem teljesített. Ha valakinek a padtársa segít, akkor mérlegelés nélkül mindkettő hallgató dolgozata sikertelen, a ZH 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.
- A Java elnevezési 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 bemutatott példákban a stringek köré rakott idézőjelek nem részei az elvárt kimenetnek, azok csak a string határait jelölik. Például ha az szerepel, hogy "3 alma", akkor az elvárt kimenet idézőjelek nélkül a 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).

A letölthető csomagban két osztályt találsz. Feladatod ennek a két osztálynak a megfelelő bővítése oly módon, hogy a feladat kéréseit meg tudd valósítani. Az osztályokat bővítheted ugyan, de a meglevő adattagokat, konstruktorokat, metódusokat NEM változtathatod meg, mert ez a kiértékelő működését módosíthatja.

Hallgato osztály (9 pont)

Minden hallgató rendelkezik `keresztNev`-vel, `vezetekNev`-vel, `neptun` kóddal, illetve minden hallgatónak adottak az eddig elvégzett kurzusai érdemjegyekkel, a `teljesitesek` adattag által. A `Hallgato` osztály előre rögzített konstruktora ezeket állítja be.

A `zh` feladata a következő osztály megadott metódusainak implementálása, amely során igény szerint kiegészítheted, illetve adott esetben ki is kell egészítened a *Hallgato* osztályt. Figyelj rá, hogy mindeközben az osztály előre adott elemeit **ne** változtasd meg!

A megoldáshoz kell készítened egész biztosan egy olyan metódust, amely egy hallgató legjobb osztályzatát adja vissza. Legyen ez a metódusod paraméter nélküli és egy egész értéket adjon vissza, a metódus neve tetszőleges. **VIGYÁZZ!** A `Hallgato` osztálynak ez legyen az egyetlen ilyen metódusa (paraméter nélküli, egészet visszaadó), különben nem kapsz pontot több tesztre sem, amely használná ezt a metódust!

Zh4 osztály (11 pont)

Készítsd el a `rendez` metódust, amely a paraméterében hallgatók tömbjét kapja.

A függvény visszatérési értéke az a kollekció, amely a legjobb osztályzatuk alapján teszi bejáráhatóvá a hallgatókat, először a legjobb hallgatóktól indulva. Amennyiben többen is ugyanazzal a legjobb osztályzattal rendelkeznek, akkor a `neptun` kód szerinti ábécé sorrendben vegyük a hallgatókat! A `neptun` kód egyedi, minden hallgató esetében különböző. Amennyiben a paraméter *null*, esetleg *üres tömb*, a metódus egy új, üres kollekcióval térjen vissza.

Bár a megoldás nem feltétlen indokolná, de tedd a `Hallgato` osztályt összehasonlíthatóvá, azaz bármely esetben legyen lehetőség két hallgató összehasonlítására a fenti meghatározás alapján.

Készítsd el a `teljesitesNelkuliekTorlese` metódust, ami egy hallgatókból álló listát vár paraméterben, és nem ad vissza semmit. A feladatod a listából azon hallgatók törlése, akiknek nincs egy teljesítésük sem.

A teljes `zh` **20** pontot ér, a részpontszámokról a bíró tesztesetei adnak tájékoztatást!

Jó munkát!