

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 ilyen 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 a példa bemenetre az elvárt kimenet az, 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).

Időjárásjelentés

Egy idegen civilizáció, a Maranditiliák célba vették a Földet, és leszálltak a hatalmas űrhajójukkal. Szerencsére ők egy békés civilizáció, akiknek céljuk az emberek segítése. Elhozták nekünk a legmodernebb időjárás előrejelző készüléküket, amely segítségével akár évekre előremenőleg pontosan meg lehet határozni az időjárást. De ez tényleg így van? Vagy csak szélhámosok, és igazából le akarják igázni a Földet a gyíkemberek segítségével? Az UFO 3 betű, mint ahogyan a BAJ is. Véletlen lenne? Nem hiszem!!!



A feladat megoldása során használd a `minta.zip` fájlban lévő kiindulási projektet, amelyben megtalálható két osztály. Az osztályok adattagjait, illetve a meglévő metódusait ne változtasd meg! A program tesztelhető lokálisan is a mellékelt `main` függvény segítségével.

Időjárásjelentés (9 pont)



Összehasonlíthatóság (5 pont)

Oldd meg, hogy egy `IdojarasJelentes` objektum összehasonlítható legyen egy másik `IdojarasJelentes` objektummal. Az összehasonlítás a *hitelesseg* alapján (növekvő sorrendben) történjen. Amennyiben a két objektum hitelessége megegyezik, akkor az összehasonlítás a *varos* alapján történjen (ABC sorrendben).

Ehhez implementáld a megfelelő interface-t az osztályban és valósítsd meg a megfelelő metódust!

hozzáadás (2 pont)

Valósítsd meg a *hozzaad* metódust, amely egy szöveget (nap neve) és egy számot (hőmérséklet) vár. A metódus amely célja, hogy egy adott nap hőmérsékletét hozzáadja az *előrejelzések*hez. Amennyiben még az előrejelzésekben nincs benne a paraméterben kapott nap, akkor tegyük be a kapott napot, a hozzá tartozó hőmérséklettel. Amennyiben már benne volt, akkor ne történjen semmi.

melegNapokSzama (2 pont)

Valósítsd meg a *melegNapokSzama* metódust, amely nem vár paramétert. A metódus célja, hogy visszaadja, hogy az előrejelzések alapján hány meleg nap várható. Azok számítanak meleg napnak, ahol a hőmérséklet magasabb, mint 30 fok.

IduwujarasKereso osztály (11 pont)



rendezés (6 pont)

Készítsd el a `rendez` metódust, amely a paraméterében időjárásjelentések tömbjét várja és egy (tetszőleges) kollekciót ad vissza.

A metódus által visszaadott kollekcióban az időjárásjelentések a fent megadott sorrendben legyenek (tehát először hitelesség szerint növekvő sorrendben, egyenlőség esetében város szerint ABC sorrendben).

Előfordulhat, hogy a paraméter null vagy üres tömb, ilyenkor a metódus egy (tetszőleges) üres kollekcióval térjen vissza.

irrelevánsak (5 pont)

Készítsd el az `irrelevansakTorlese` metódust, amely paraméterben egy időjárásjelentésekből álló listát, illetve egy egész számot vár, ami azt jelöli, hogy mennyitől tekintünk hitelesnek egy időjárásjelentést.

A metódus feladata, hogy törölje ki a listából azokat az időjárásjelentéseket amelyekre az alábbi két feltétel közül legalább az egyik igaz:

- a hitelessége kisebb a paraméterben kapott értéknél
- a város nem *Szeged*

A metódus adja vissza, hogy hány elemet töröltünk a listából.

Jó munkát!