Programozás I. Nagy zh

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 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).

A feladat megoldása során használd a `minta.zip` fájlban lévő kiindulási projektet! A projekt egy egyszerű tehergépjárművet (kamiont) valósít meg, amivel a vezetői tudnak majd fuvarozni a rendelkezésre álló kilométerkorlátjuk és a tehergépjármű üzemanyagból fakadó kilométerkorlátjának megfelelően.

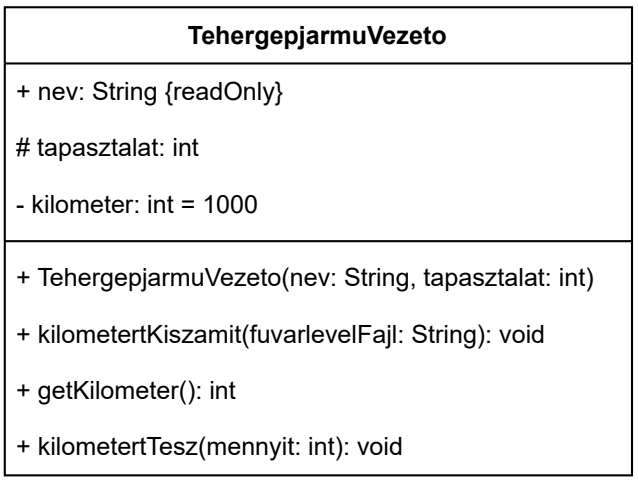
1. feladat: Tehergépjárművek és tehergépjármű-vezetők (15 pont)

1.1. feladat: Tehergépjármű (3 pont)

Módosítsd a `Tehergegpjarmu` osztályt az alábbiak szerint!

- A *fuvarok* kollekció tartalmazza a tehergépjárműnek a különböző fuvarokon megtett kilométereit. A probléma az, hogy ha két fuvarban is ugyanannyi kilométere van a tehergépjárműnek, akkor a kettő közül csak az egyik kerül bele a kollekcióba. Módosítsd az adattagot (a többi kód módosítása nélkül), hogy azt is el tudja tárolni, ha több fuvarban is ugyanannyi kilométert tett meg a tehergépjármű. **(1 pont)**
- A *kilometertTesz()* metódus bizonyos esetekben egy `IllegalArgumentException` típusú kivételt dob. Cseréld ki a dobott kivételt egy *saját kivétel osztályra* úgy, hogy a dobott kivétel hibaüzenete ne változzon! A *kilometertTesz()* metódus fejlécén ne változtass! **(2 pont)**

1.2. feladat: Tehergépjármű-vezetők (8 pont)



Készítsd el a `TehergepjarmuVezeto` osztályt a mellékelt UML diagram alapján! **(1 pont)**

Készítsd el az osztály konstruktorát! A konstruktor várja a tehergépjármű-vezető nevét és a tehergépjármű vezetéssel eltöltött éveinek számát, és azok alapján beállítja az adattagokat. Kezdetben a tehergépjármű-vezető 1000 km-t tud vezetni. **(1 pont)**

A *getKilometer()* metódus visszaadja, hogy a tehergépjármű-vezető mennyi kilométert vezethet még. **(0 pont)**

A *kilometertTesz()* metódus paraméterben várja, hogy a tehergépjármű-vezető mennyi kilométert szeretne haladni, tehát a sofőr még megtehető kilométere (azaz `kilometer` adattagja) ennyivel csökkenjen. Amennyiben a sofőrnél nincs elég kilométere, akkor dobj egy ugyanolyan típusú kivételt, ugyanazzal a hibaüzenettel, mint a `Tehergapjarmu` osztály *kilometertTesz()* metódusában. **(2 pont)**

A *kilometertKiszamit()* metódus paraméterben egy fájlnak a nevét várja. A metódusnak ebből a fájlból kell olvasnia. A fájl minden sora egy pihenést (ahol töltődik a kilométer) vagy haladást (ahol fogy a kilométer) ír le. A fájlban nem tudjuk, hogy hány sor van. A metódus feladata, hogy a fájl tartalma alapján módosítsa a tehergépjármű-vezető megtehető kilométereinek számát.

A fájl minden sora úgy néz ki, hogy először látható, hogy pihenésről vagy haladásról van szó, utána pedig a kilométerek mennyisége olvasható. Az adatok pontosvesszővel vannak egymástól elválasztva.

Példa fájl:
PIHENES;200
HALADAS;1000
PIHENES;500
HALADAS;700

Ha a sofőrnek kezdetben 1000 kilométere volt, akkor a végén 0 kilométere lesz (1000+200-1000+500-700). **(4 pont)**

1.3. feladat: Közös rész (4 pont)

Bár a tehergépjármű és a tehergépjármű-vezető egymástól elég távol áll, mégis van bennük valami közös. Minden olyan kódrészletet, amit tudsz, szervezz ki egy interface-be, amelyet ezután mind a két osztályod implementál.

2. feladat: Fuvarkozvetito (15 pont)

Készítsd el a `Fuvarkozvetito` osztályt!

Az osztály egy nagyon egyszerű fuvarközvetítő működését írja le, amely lehet egyszemélyes közvetítő vagy többszemélyes közvetítő. A fuvarközvetítő különböző fuvarokat kínál megadott kilométeren (ez persze idővel változhat (útlezárás/új út)). A fuvarközvetítő mindegyik fuvarból véges mennyiséget tud kínálni, de természetesen időnként adott fuvar ismét megvalósítható. A tehergépjármű-vezetők betérhetnek a fuvarközvetítőhöz, és a rendelkezésre álló fuvarok közül vállalhatnak egyet, amennyiben a fuvar megfelel a tehergépjármű-vezető által megadott kilométerkorlátnak.

A megoldást tetszőleges módon elkészítheted, ehhez tetszőleges új adattagokat, metódusokat, akár osztályokat is létrehozhatsz. Azonban fontos, hogy az osztály ne tartalmazzon publikus adattagot, illetve az elvárt 5+1 metóduson kívül ne legyen benne másik publikus metódus. A megoldás elkezdése előtt érdemes végigolvasni az összes feladatot, a feladat pontos megértése céljából!

Az osztály rendelkezzen egy 1 paraméteres konstruktorral, amely egy logikai értéket vár. Ez az érték igaz, ha egyszemélyes fuvarközvetítőről van szó, hamis ha többszemélyes fuvarközvetítőről.

Először is szükségünk van egy adott fuvar távolságának és rendelkezésre álló mennyiségének lekérdezésére. Készíts egy metódust, amely paraméterben egy fuvar viszonylatát várja (pl. "Bonyhád-Fácánkert-Szeged"), és egy 2 elemű tömbben adja vissza az adott fuvar adatait. A viszonylat egy szöveg, ami a fuvarra nézve mindig egyedi, tehát nem lesz duplikációja. A tömb első eleme a fuvar távolságát, míg a második eleme a rendelkezésre álló mennyiségét tartalmazza. Amennyiben a keresett fuvart a közvetítő nem kínálja, akkor a tömb mindkét eleme *null* legyen! Ha a közvetítő bármikor is kínált egy fuvart, akkor a visszaadott értékek akkor se legyenek null-ok, ha éppen nincs a fuvar kínálatban (lásd: későbbi metódusok)! **(0 pont)**

FONTOS! A megoldások tesztelése a fenti lekérdező metódus segítségével történik, ami azt jelenti, hogy ennek helyes megvalósítása nélkül az összes többi feladat 0 pontot ér. Feltöltés előtt ezt a metódust mindenképpen készítsd el, különben az osztályra a Bíró nem fog pontot adni!

Az osztálynak az alábbi 4 funkciót kell megvalósítania:

A közvetítőnek tudnia kell fogadni új közvetítendő fuvarokat. Az erre használható metódus paraméterben várja a fuvar viszonylatát, illetve a fuvar távolságát, valamint a rendelkezésre álló mennyiséget, azaz, hogy hány tehergépjármű-vezetőnek kínálhatja a fuvart.
Az új kínálandó fuvarok az eddigi fuvarhoz hozzáadásra kerülnek (pl. ha volt Baja-Szegedből már eddig 10, de most felajánlottak még 5-öt, akkor most már 15 Baja-Szeged fuvart kínál a közvetítő.).
Amennyiben egy új viszonylatú fuvarról van szó, akkor a közvetítő felveszi a kínálatába az új fuvart.
Ha egy közvetítő már korábban bármikor kínált egy fuvart, akkor annak távolsága álljon át az újra (mert ez a friss infó.) **(4 pont)**

A közvetítőnek reagálnia kell a térképészeti változásokra (útzár, új út). Az erre használható metódus várja a fuvar viszonylatát, illetve a fuvar új távolságát. Ha egyszemélyes fuvarozóról van szó, akkor a metódus módósítsa a kilométert az új kilométer és a régi átlagára, hisz egyszemélyként nem tud alapos térképet adni a tehergépjármű-vezetőknek. Ha többszemélyes fuvarozóról van szó, akkor a metódus módósítsa a kilométert az új kilométerre. Ha olyan fuvar távolsága módosul, ami eddig nem szerepelt a fuvarkínálatban, kerüljön felvételre 0 mennyiséggel, hisz jó, ha erről is információk állnak rendelkezésre. **(2 pont)**

A közvetítő weboldalán szeretné listázni a fuvarjainak viszonylatát. Ehhez kétféle rendezést kell biztosítani: ABC szerinti, illetve távolság szerinti (növekvő) sorrendet. Készíts egy metódust, amely paraméterben várja, hogy mi alapján szeretnénk rendezni (true: távolság alapján; false: ABC szerint). A metódus egy megfelelő méretű String[] tömbben adja vissza a fuvarok viszonylatát, a kért módon rendezve! **(3 pont)**

A fuvarközvetítő legfontosabb funkciója azért mégis csak az, hogy a tehergépjármű-vezetők találjanak fuvart nála. Készíts egy metódust, amely a fuvarvállalás lebonyolításáért felelős. A metódus megkapja a tehergépjármű-vezető személyét, illetve egy kollekciót az elvállalni kívánt fuvarokról. A fuvarokat egy Map adja meg, amelyben a kulcsok a fuvarok viszonylatai, az értékei pedig az elvállalni kívánt mennyiség. A metódus bonyolítsa le a vállalást az alábbiak szerint:

- Az aláírás előtt a tehergépjármű-vezető kiszámolja, hogy van-e elég kilométere a listája alapján. Ha nincs elég kilométere, akkor inkább semmit nem vállal el, hazamegy és pihen. (ha egy fuvarból nem kínál elegendő mennyiségűt a közvetítő, akkor nyilván csak azzal számolunk, ami van).
- Ha van elég kilométere, akkor a listáján szereplő fuvarokból a megadott mennyiségű fuvart vállal el. Ilyenkor (ahogy várnánk), a közvetítő kínálatában lévő fuvar csökken a megadott mennyiséggel, illetve a sofőr levonja a kilométereiből. Előfordulhat, hogy egyes fuvarokból egyáltalán nincs a kínálatban, vagy nincs annyi, amennyit a tehergépjármű-vezető szeretne. Előbbi esetben az adott fuvart a vezető kihagyja, utóbbi esetben annyit vállal el, amennyi rendelkezésre áll.
- A metódus adja vissza, hogy a tehergépjármű-vezető el tudta-e vállalni, amelyet a listájára felírt. **(6 pont)**

Jó munkát!