

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 "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!

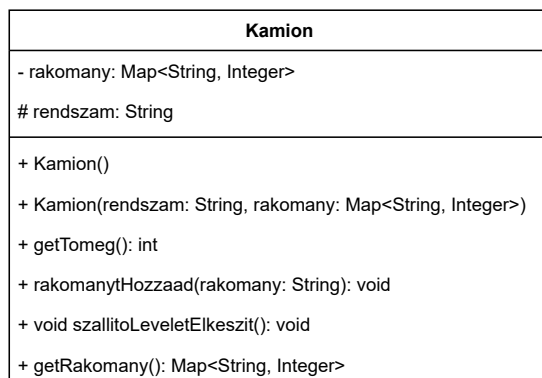
1. feladat: Kamionok és Vonatok (15 pont)

1.1. feladat: Vonat (4 pont)

Módosítsd a `Vonat` osztályt az alábbiak szerint!

- Az *utasok* kollekció tartalmazza a vonat által szállított utasokat, jelenleg letárolja minden egyes utashoz a tömegét is, viszont arra nincs szükségünk. Módosítsd az adattagot (csak a szükséges kód módosításával), hogy az utasokról csak a nevüket tárolja el. Figyelj arra, hogy egyszerre több ember is rendelkezhet ugyanazzal a névvel! **(1 pont)**
- A *rakomanyHozzaad(String utas)* metódus bizonyos esetekben egy `RuntimeException` típusú kivételt dob. Cseréld ki a dobott kivételt egy *saját kivétel osztályra* úgy, hogy a dobott kivétel hibaüzenete ne változzon! A *rakomanyHozzaad()* metódus fejlécén ne változtass! **(3 pont)**

1.2. feladat: Kamion (7 pont)



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

Készítsd el az osztály 2 konstruktorát! Az alapértelmezett konstruktor állítsa a rendszámot "AAA-000" szövegre, a rakományt pedig inicializálja egy üres, tetszőleges Map objektummal. A paraméteres konstruktor várja a tulajdonságok alap értékeit paraméterként és inicializálja velük az objektumot. **(1 pont)**

A *getTomeg()* metódus visszaadja, hogy a kamion mekkora tömeget szállít jelenleg (a kamion tömege a rakományok tömegének összessége), amennyiben nem szállít semmit, térjen vissza *-1* értékkel. **(1 pont)**

A *rakomanyHozzaad()* metódus paraméterben várja, hogy milyen terméket szeretnének felrakodni (lehet, hogy már van a trélerben ilyen termék, lehet, hogy nincs). Minden rakomány súlya 1 egység. Értelmszerűen, amennyiben már van egy adott termékből a rakományok között, akkor egy új rakomány felrakodásával megnő az adott típusú rakomány tömege pontosan

1-gyel. Amennyiben a kamion terhelése meghaladná a 100 000-es értéket, akkor dobj egy ugyanolyan típusú kivételt, ugyanazzal a hibaüzenettel, mint a `Vonat` osztály `rakomanytHozzaad()` metódusában. **(2 pont)**

A `szallitoLeveletElkeszit()` metódus kiírja fájlba, hogy milyen termékeket szállít a kamion. A fájl neve legyen `szallitoLevel_<RENDSZAM>.txt`, ahol a `<RENDSZAM>` a kamion rendszáma, a fájl tartalma pedig a kamion által szállított termékek nevei és mennyiségét, kettősponttal elválasztva. Amennyiben a kamion nem szállít semmit, akkor a fájl tartalma legyen `"Ures a rakomany!"`.

A fájl első sora legyen az, hogy *Rakomany*: az utána következő minden sora úgy néz ki, hogy először a termék neve, majd kettőspont, majd a termék mennyisége kettősponttal elválasztva. Minden termék után legyen újsor karakter.

Példa fájl:

Rakomany:

Alma:200

Körte:1000

Szilva:500

Vinyak:700

Ha egy terméknek a mennyisége nulla, akkor azt ne írd bele a fájlba! **(3 pont)**

1.3. feladat: Közös rész (4 pont)

A vonat és kamion különböző kaliberű járművek, viszont van bennük valami közös. Minden olyan kódrészletet, amit tudsz, szervezz ki egy interface-be, amelyet ezután mind a két osztályod implementál.

2. feladat: Vámos (15 pont)

Készítsd el a Vámos osztályt!

Az osztály egy vámügyintéző működését írja le. Munkájából kifolyólag tudja, hogy egy *cég* milyen kamionokkal szállítja az áruját. A vámügyintézőnek a következő feladatai vannak:

A megoldást tetszőleges módon elkészítheted, ehhez tetszőleges új adattagokat, metódusokat, akár osztályokat is létrehozatsz. 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 alapértelmezett konstruktorral, ami inicializálja az adattagokat. A nap elején nem tudja, hogy kik fognak keresztül haladni, azt majd a később kapja meg.

Az osztálynak az alábbi 5 funkciót kell megvalósítania:

Készíts el egy olyan metódust, ami paraméterben megkapja egy cégnek a nevét és visszatér egy olyan kollekcióval, amely tartalmazza a cég nevét és az általa használt kamionokat. Amennyiben olyan cég nevét kapja meg, amely nem szerepel a rendszerben, akkor üres kollekcióval térjen vissza. **(1 pont)**

A vámosnak rugalmasnak kell lennie, a nap folyamán folyamatosan érkezhetnek be utasítások, amik megmondják, hogy várhatóan milyen cégek fognak áthaladni a határon. Készíts egy metódust, amely paraméterben megkapja a cég nevét és a cég által alkalmazott kamionokat. A metódusnak a cég neve alapján el kell tárolnia a cég által alkalmazott kamionokat.

Amennyiben egy, már lejelentett cég újra megjelenik, akkor a vámos nem írhatja felül az eredeti listáját. **(1 pont)**

Időnként a vámosnak le kell jelentenie, hogy milyen cégek fognak áthaladni a határon. Készíts egy metódust, amely visszaadja azoknak a cégeknek a nevét, amelyekről a vámos tud. Az irodai dolgozók viszont nem hajlandóak az így kapott adat rendezésével foglalkozni, így csak megmondják, hogy éppen ABC-szerint növekvő vagy csökkenő sorrendben szeretnék-e megkapni a cégek listáját. Ha a paraméter *false*, a metódus ABC-szerint csökkenőben, *true* esetén pedig növekvő sorrendben legyen, tehát a paraméter azt mondja meg, hogy növekvő legyen-e az ABC-szerinti sorrend. **(3 pont)**

A nap végén statisztikát várnak el a vámostól, leginkább arra kíváncsiak, hogy melyik termékből szállítottak az adott nap a legtöbbet (feltehetőleg azért, hogy a jövőben külön adót vethessenek ki rá). Készíts el egy olyan metódust, amely nem vár paramétert, visszaadja a legtöbbet szállított termék nevét. Amennyiben nincs egy termék sem, a metódus térjen vissza üres sztringgel. **(5 pont)**

Bizonyos termékek be vannak tiltva, készíts egy metódust, ami megkapja egy tiltott termék nevét és a következő lépéseket hajtja végre: **(5 pont)**

- Eltávolítja az összes tiltott terméket szállító céget a rendszerből.
- Visszatér a még whitelist-en lévő cégek nevével ABC sorrendben, egy listában.

Jó munkát!