

# Programozás I. Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2024. tavasz

## Általános követelmények, tudnivalók

- A feladat elkészítési határideje: **vasárnap 23:59:59**. Ez szigorú határidő, a Bíró előre megadott időben zár, pótlásra nincs lehetőség.
- 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.
- 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://biro2.inf.u-szeged.hu/Hallg/IB204L/FELADAT/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).

# Leánykérés

## 1. feladat (3 pont)

Készítsd el a **VisszautasitasException** nevű publikus kivétel osztályt. Készíts default konstruktort, ami minden esetben a „A leanykerest visszautasítottak” üzenettel inicializálja őst.

## 2. feladat (3 pont)

Készíts egy **Ember** absztrakt osztályt, amely rendelkezik

- egy publikus, szöveges név adattaggal, melynek neve *nev* legyen

Az osztálynak legyen egy absztrakt *velemeney* metódusa, amely egy másik embert vár paraméterként és egy egész számot ad vissza.

## 3. feladat (10 pont)

Készíts egy **Kisasszony** osztályt ami öröklődik az *Emberből* és rendelkezik

- kedvesség (egész), ennek neve legyen *kedvesseg*
- szépség (egész), ennek neve legyen *szepseg*
- örökség (egész), ennek neve legyen *orokseg*

adattagokkal. Az adattagok csak csomagon belül láthatóak, nem szükséges hozzájuk sem getter, sem setter.

Készíts egy *paraméteres konstruktort*, amely az alábbi sorrendben várja az adattagok értékeit: *név*, *kedvesség*, *szépség*, *örökség*. A kérők listáját pedig egy üres, tömbbel megvalósított listával inicializálja.

Definiáld a *velemeney* metódust: a kapott emberről döntse el, hogy Agglegény típusú-e, ha nem akkor 0-t, ellenkező esetben az Agglegény három számbeli tulajdonságából (a következő feladatban szerepelnek) számoljon átlagot (egészosztással, nem kell lebegőpontos osztást végezni).

Legyen egy *kerotFelvesz* metódus, amely nem ad vissza értéket, és egy *Agglegényt* vár paraméterként, amelyet hozzáadja a kérők listájához.

Készíts egy *dontes* metódust, amely *Agglegényt* ad vissza és nem vár paramétert. Amennyiben nem volt kérője, null értékkel térjen vissza. Egyébként a kérők listájából kiveszi az utolsó olyan *Agglegényt*, amelyhez a legmagasabb vélemény értéke tartozik az adott kisasszonynak. Ezután üríti a listát és visszatér ezzel az Agglegénnyel.

## 4. feladat (10 pont)

Készíts egy **Agglegeny** nevű osztályt, ami az Emberből származik, rendelkezik:

- kedvesség (egész), melynek neve legyen *kedvesseg*
- jóképűség (egész), melynek neve legyen *jokepuseg*
- vagyon (egész), melynek neve legyen *vagyon*

adattagokkal. Az adattagok csak csomagon belül legyenek láthatóak. Legyen egy *paraméteres konstruktor*, amely az alábbi sorrendben várja az adattagokat: *név*, *kedvesség*, *jóképűség*, *vagyon*.

Definiáld felül a *velemeney* metódusát, hogy nem Kisasszony típusú emberre nullát adjon, ellenkező esetben pedig térjen vissza a kisasszony számbeli tulajdonságainak abszolút összegével.

Legyen egy *leanykeres* metódus, amely egy Kisasszony típusú objektumot vár paraméterül és nem tér vissza semmivel, illetve dobhat *VisszautasitasException* kivételt. Ez a metódus ellenőrizze az adott Kisasszony véleményét róla, amennyiben ez kisebb mint 5, dobjon egy új *VisszautasitasException* kivételt. Amennyiben ez nem következik be, vegyük fel az adott Kisasszony kéréi közé a *kerotFelvesz* metódussal.

Készíts egy *tanchaz* nevű metódust, amely egy Kisasszonyokat tároló listát kap paraméterként. Ez a metódus válassza ki a legjobban tetsző lányt (azt a lányt, amelyik véleménye a legnagyobb). Ennek a lánynak próbálja meg megkérni a kezét! Ha a listában nincs egy kisasszony sem, akkor ne történjen semmi. Ha több olyan lány is van, akiről a legjobb a véleménye az agglegénynek, akkor az első lány legyen kiválasztva.

## 5. feladat (9 pont)

Készítsd el a **LeanykeresFuggvenyCsomag** nevű osztályt. Ebből az osztályból ne szármasson további osztály!

Az osztályban csak egy darab statikus függvény legyen, aminek a neve legyen *beolvas*, amely egy paramétert vár (a fájl nevét, amiből olvasni szeretnénk), és egy *Ember* objektumokat tároló, tömbbel megvalósított listával tér vissza.

A metódus a megadott fájlból olvas be embereket. A fájlban minden sor egy ember adatait tartalmazza, ezek *#* karakterrel vannak tagolva. Az első elem mindig „Kisasszony” vagy „Agglegény”. Ezután egy név és három számérték következik, amelyek az adott ember tulajdonságait írják le (abban a sorrendben, ahogy az adott osztály konstruktora várja a paramétereket).

A létrehozott objektumokat tedd be egy listába, amivel a függvény a beolvasás után térjen vissza.

Amennyiben a fájlból olvasás során kivétel keletkezik, térj vissza egy üres listával, és írd ki az alapértelmezett hibakimenetre az alábbi szöveget: `"Fajlolvadas sikertelen"`.

Jó munkát!