

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).

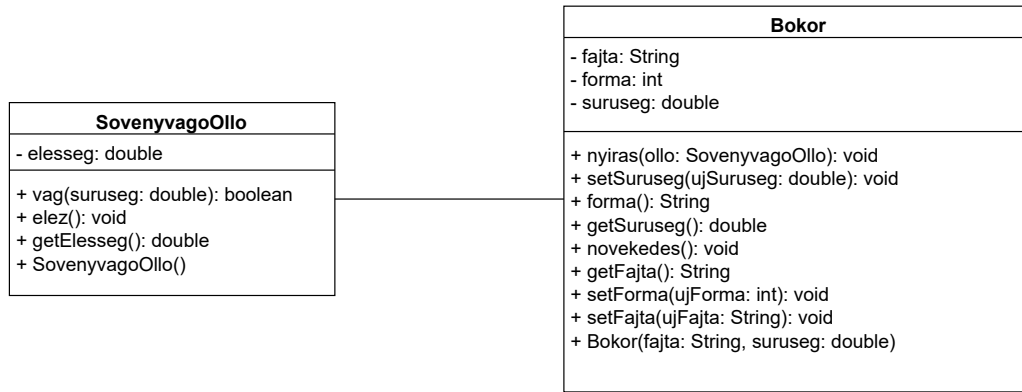
SovenyvagoOllo osztály (8 pont)

Írj egy `SovenyvagoOllo` nevű osztályt, ami egy sövényvágó ollót reprezentál, egyetlen adattaggal, ami az olló élessége, ami egy 0-100 közötti lebegőpontos szám. Minden olló legfeljebb ilyen sűrűségű bokrot megformázni.

Az adattagokat, valamint a metódusokat az alábbi ábrán láthatjuk. Ügyelj a megfelelő láthatóságok használatára.

Az osztály egyetlen konstruktora az olló élességét alaptól 100.0-ra állítsa.

1. ábra. SovenyvagoOllo osztálydiagram



Az `elez` metódus élezzze meg az adott ollót. Minden élezés 15-tel javítja az olló élességét. Ügyelj rá, hogy 100 felett nem lehet az élesség.

A `vag` metódus paraméterben egy bokor sűrűségét kapja. A metódus térjen vissza igazzal, ha sikerül az adott sűrűségű bokrot megformáznia. Amennyiben az élesség miatt erre nincs lehetőség, térjen vissza a metódus hamissal. Amennyiben sikerül a bokor megnyírása, az olló élessége csökkenjen a formára vágott bokor sűrűségével.

Bokor osztály (13 pont)

Az adattagokat, valamint a metódusokat az ábrán láthatjuk. Ügyelj a megfelelő láthatóságok használatára.

Az osztályban definiált sűrűség adattag a bokor sűrűségét adja meg, 0-100 közötti lebegőpontos szám, a fajta a bokor fajtája.

A forma pedig csak 0 és 4 közötti értéket vehet fel, melyek jelentése:

- **0:** a bokor még csak cserje
- **1:** a bokor egy normál, kicsi bokor
- **2:** formára nyírt bokor
- **3:** a bokor kezd elburjánzani
- **4:** teljesen elburjánzott formájú bokor

A Bokor osztály konstruktora állítsa be a fajtát, valamint a sűrűséget.

A `forma` metódus szövegesen visszaadja a bokor formáját, melyek az alábbiak lehetnek:

- **0:** "cserje"
- **1:** "bokor"
- **2:** "formara nyirt"

- 3: "kisse elburjanzott"
- 4: "teljesen elburjanzott"

A `setForma` ne csináljon semmit, ha nem megfelelő értéket próbálunk beállítani, egyébként pedig az elvárható módon működjön.

A `setSuruseg` a sűrűséget állítja be, ha a minimumnál kisebb lenne, akkor a minimumra, ha a maximumnál nagyobb lenne, akkor a maximumra állítja be.

A `nyiras` metódus egy sövényvágó ollót kap paraméterül, és az azzal kapott ollóval megpróbálja elvágni az adott sűrűségű bokrot. Ha a vágás sikeres, állítsuk be a bokor formáját formára nyírtra.

A `novekedes` metódus reprezentálja a bokor növekedését, ami annyit fog jelenteni, hogy a formája változik: cserjéből bokor lesz, bokorból formás bokor, majd kissé elburjanzott bokor, végül pedig teljesen elburjanzott bokor. Ügyelj a forma adattag lehetséges értékeire!

Jó munkát!