

Objektumorientált programozás - Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2025. 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!

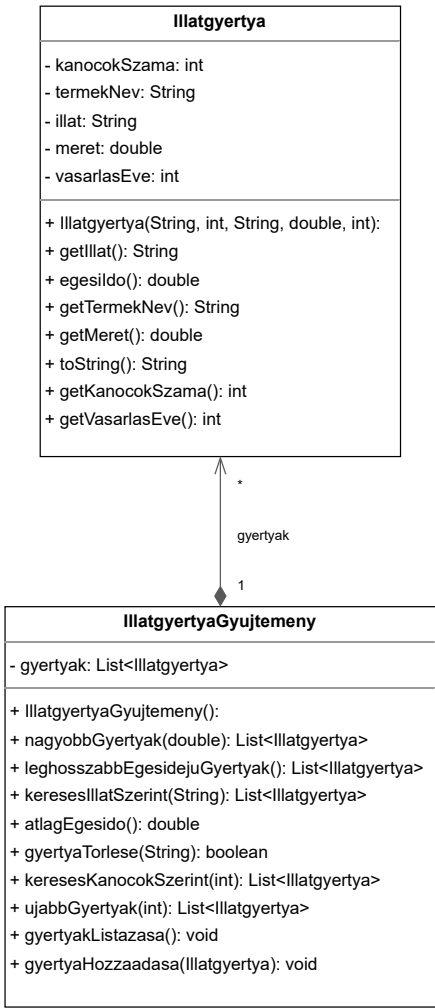
- Zip készítése: Windowson és Linuxon is lehet a GUI-ban jobb klikkes módszerrel tömörített állományt létrehozni (Windowsban pl. a 7-Zip nevű ingyenes program használatával).
 - Linux terminálon belül például a "zip feladat.zip *.java" paranccsal is elkészíthető a megfelelő állomány.
- A feladatokban 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/IBNa1002L-0/1/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).

Illatgyertya feladat

Egyre népszerűbbek az illatgyertyák napjainkban. Készíts egy gyűjteményt, ahol nyilván tudod tartani az illatgyertyáidat!

Az adattagokat, valamint az elvárt metódusokat az alábbiakban ismertetjük. Ügyelj a megfelelő láthatóságok használatára!

1. ábra. Osztálydiagram



Illatgyertya osztály

Az Illatgyertya osztály reprezentál egy illatgyertyát a gyűjteményben.

Adattagok

- `termekNev (String)`: az illatgyertya termékneve (pl.: "Cozy Home"). Létrehozás után nem módosítható.
- `vasarlasEve (int)`: az illatgyertya vásárlásának éve (pl.: 2025). Létrehozás után nem módosítható.
- `illat (String)`: az illatgyertya illata (pl. "fahéj és alma"). Létrehozás után nem módosítható.
- `meret (double)`: az illatgyertya mérete centiméterben (pl.: 10.5).
- `kanocokSzama (int)`: az illatgyertyában található kanócok száma (pl.: 1).

Konstruktor

(3 pont)

- Öt paramétert váró konstruktor: várja a termék nevét, vásárlásának évét, illatát, méretét és kanócainak számát. Ezeket az azonos nevű adattagok értékének állítsd be.

Metódusok

Getterek

(5 pont)

- `getTermekNev()`: Visszaadja a gyertya terméknevét.
- `getVasarlasEve()`: Visszaadja a gyertya vásárlásának évét.
- `getIllat()`: Visszaadja a gyertya illatát.
- `getMeret()`: Visszaadja a gyertya méretét.
- `getKanocokSzama()`: Visszaadja a gyertya kanócainak számát.

`egesido()`

(5 pont)

Ez a metódus kiszámítja a gyertya várható égésidejét órában. A következő szabályok vonatkoznak a számításra:

- A térfogat kiszámítása: feltételezzük, hogy a méret az átmérő cm-ben, és a gyertya henger alakú.
- A térfogat számítási képlete: $\pi \cdot (meret/2)^2 \cdot (meret/2)$ (ahol a magasság a méret felét vesszük).
- Alapidő: minden köbcentiméter 0.5 óra égésidőt biztosít.

- A kanócok száma arányosan növeli az égés sebességét, ezért az alapidőt el kell osztani a kanócok számával.

toString() (2 pont)

A `toString` metódus térjen vissza egy olyan szöveggel, amely tartalmazza a gyertya összes adatát, beleértve a számolt égésidőt is, az alábbi formátumban:

"[termekNev] ([illat]), meret: [meret] cm, kanocok: [kanocokSzama], vasarlas eve: [vasarlasEve], varhato egesido: [egesido] ora"

IllatgyertyaGyujtemeny osztály

Az `IllatgyertyaGyujtemeny` osztály egy illatgyertya gyűjteményt reprezentál.

Adattagok

- `gyertyak` (`List<Illatgyertya>`): az illatgyertyák listája.

Konstruktor

(1 pont)

- Paraméter nélküli konstruktor: inicializálja a gyertyák listáját egy üres `ArrayList`-tel.

Metódusok

gyertyaHozzaadasa(Illatgyertya gyertya) (2 pont)

Ez a metódus egy új illatgyertyát ad hozzá a gyűjteményhez. A következő lépéseket hajtja végre:

- Hozzáadja a gyertyát a gyertyák listájához.
- Kiír egy üzenetet a konzolra: "[illat] illatu gyertya kerult a gyujtemenybe"

gyertyaTorlese(String termékNev) (2 pont)

Ez a metódus törli az adott terméknevű illatgyertyát a gyűjteményből. A következők vonatkoznak rá:

- Visszatérési értéke logikai érték: igaz, ha sikeres volt a törlés, hamis, ha nem.
- Ha nincs ilyen nevű illatgyertya, kiír egy üzenetet a konzolra: "Nincs ilyen nevu illatgyertya: [termekNev]" és hamissal tér vissza.

keresesIllatSzerint(String illat) (2 pont)

Ez a metódus visszaadja azon illatgyertyák listáját, amelyek illata tartalmazza a keresett szöveget. A keresés legyen kisbetű-nagybetű független, és részleges egyezés is találatnak számít.

keresesKanocokSzerint(int kanocokSzama) (2 pont)

Ez a metódus visszaadja azon illatgyertyák listáját, amelyekben pontosan a megadott számú kanóc található.

nagyobbGyertyak(double meret)

(2 pont)

Ez a metódus visszaadja azon illatgyertyák listáját, amelyek mérete nagyobb a megadott értéknél.

ujabbGyertyak(int ev)

(2 pont)

Ez a metódus visszaadja azon illatgyertyák listáját, amelyeket a megadott évnél később vásároltak.

gyertyakListazasa()

(1 pont)

Ez a metódus kilistázza a gyűjteményben található összes illatgyertyát. Ha a gyűjtemény üres, kiírja, hogy "A gyujtemeny ures.". Minden más esetben pedig először kiírja, hogy "A gyujtemenyben talalhato illatgyertyak:", majd pedig új sorban, sorszámozva kiírja az aktuális gyertya indexét, majd pedig meghívja az adott gyertya toString metódusát. Például: "0. <0. gyertya toString>", új sorban ezt követi az 1. index és a hozzá tartozó toString, stb.

atlagEgesido()

(2 pont)

Ez a metódus kiszámítja és visszaadja a gyűjteményben található illatgyertyák átlagos égésidőjét. Ha a gyűjtemény üres, 0.0-val tér vissza.

leghosszabbEgesidejuGyertyak() (2 pont) Ez a metódus meghatározza és visszaadja a gyűjteményben található legnagyobb égésidőjű illatgyertyákat. A működése:

- A visszatérési érték egy lista, amely tartalmazza az összes olyan gyertyát, amelynek égésidője a legnagyobb a gyűjteményben (a pontosság érdekében 0.001 hibahatárral, tehát egy gyertya, amelyik 102.333 percig ég és egy másik, amelyik 102.334 percig ég, azonosnak számít).
- Ha több gyertya is azonos leghosszabb égésidővel rendelkezik, akkor mindegyiket tartalmaznia kell a visszaadott listának.
- Ha a gyűjtemény üres, a metódus egy üres listát ad vissza.

Példák:

- Ha a gyűjteményben több gyertya is azonos égésidővel rendelkezik, és ez a leghosszabb, akkor a metódus mindegyiket visszaadja.
- Ha csak egy gyertya rendelkezik a leghosszabb égésidővel, akkor a metódus csak ezt az egy gyertyát adja vissza (egy egy elemű listában).
- Ha a gyűjtemény üres, a metódus egy üres listát ad vissza.

Jó munkát!