

Programozás I. Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2023. tavasz

Általános követelmények, tudnivalók

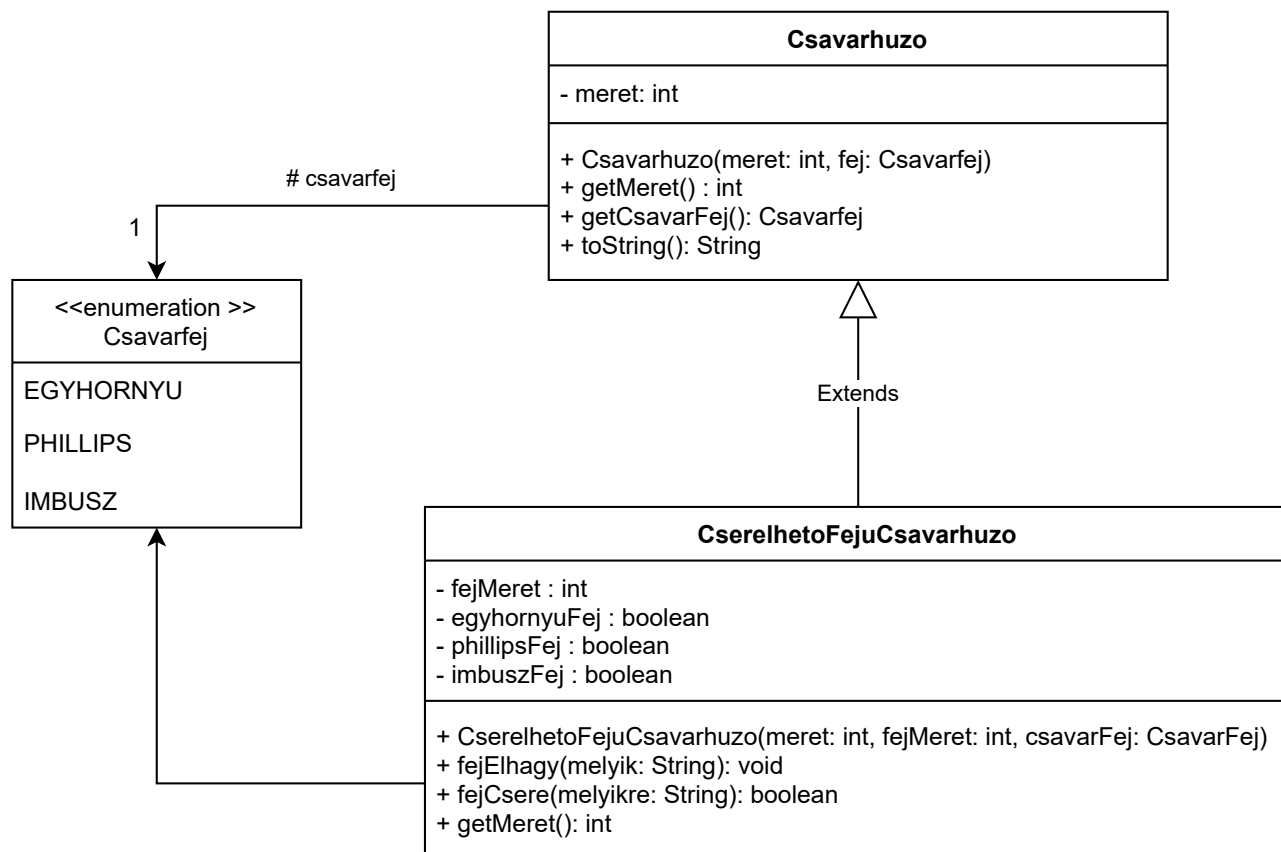
- A feladat elkészítési határideje: **vasárnap 23:59:59**. Ez szigorú határidő, a Bíró előre megadott időben zár, pótlásra nincs lehetőség.
- A feladatokat számítógép előtt kell megoldani, tetszőleges fejlesztői környezetben, tetszőleges operációs rendszer segítségével.
- Az elkészült programot **20** 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.
- Az órán tanult 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 bemutat 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 az 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).

Csavarhúzó feladat

Készíts egy `Csavarhuzo` nevű osztályt, ami csavarhúzót reprezentál. Az adattagokat, valamint az elvárt metódusokat az alábbi ábrán láthatjuk. Ügyelj a megfelelő láthatóságok használatára!

1. ábra. Csavarhúzó osztálydiagram



Csavarfej

Készíts egy `Csavarfej` nevű publikus felsorolást, az UML alapján!

Csavarhúzó

A `Csavarhuzo` és `Csavarfej` közötti asszociáció az ábráról is leolvashatóan adattag formájában valósul meg.

Konstruktor

- Kettő paramétert váró konstruktor: várja a csavarhúzó méretét, valamint egy csavarfejet, ezeket az azonos nevű adattagok értékének állítsd be.

Metódusok

A kért getterek és setterek az elvárásuknak megfelelően működjenek!

A `toString` metódus térjen vissza egy szöveggel, az alábbiak szerint:

- Amennyiben egyhornyú a csavarfej, térjen vissza a "hagyományos feju csavarhuzo" szöveggel.
- Amennyiben phillips fejű a csavarhúzó, térjen vissza a "csillagfeju csavarhuzo" szöveggel.
- Amennyiben imbusz fejű a csavarhúzó, térjen vissza a "imbuszfeju csavarhuzo" szöveggel.

Cserélhető fejű csavarhúzó

A cserélhető fejű csavarhúzókhöz alapvetően jár mindhárom fejtípus (ám ezeket el is hagyhatjuk), melyeket a `egyhornyuFej`, `phillipsFej`, `imbuszFej` reprezentál (a változó igaz, ha az adott fej még rendelkezésre áll, hamis, ha már elhagytuk). Mivel alapból jár mindhárom fej a cserélhető fejű csavarhúzókhöz, ezért az adattagok értéke kezdetben legyen igaz.

Konstruktor

- Három paramétert váró konstruktor: várja a csavarhúzó méretét (a fejjel együtt), a cserélhető fej méretét, valamint egy csavarfejet, ezeket az azonos nevű adattagok értékének állítsd be.

Metódusok

A `fejElhagy` metódus működése: a kapott szöveg alapján (amely az alábbiak egyike: "egyhornyu", "phillips", "imbusz") az adott fejhez tartozó logikai változót állítsa hamisra. Arra nem kell felkészülni, hogy olyan csavarfejet hagyunk el, ami éppen a csavarhúzón található, ez a valóságban sem fordul elő.

Az `fejCsere` metódus működése: paraméterben kapunk egy szöveget (amely az alábbiak egyike: "egyhornyu", "phillips", "imbusz"), majd a beérkező szöveg alapján cseréljük le a fejet. Amire mindenképpen figyelni kell:

- Amennyiben a jelenlegi és az új fej megegyezik, ne történjen semmi, térjünk vissza hamissal.
- Amennyiben a beállítani kívánt fejet már elhagytuk (az ezt reprezentáló logikai változó hamis), ne történjen semmi, térjünk vissza hamissal.
- Amennyiben a jelenlegi fej nem egyezik meg az új fejjel, és az új fej rendelkezésre áll, cseréljük le azt (az örökölt `csavarfej` adattag beállításával).

Definiáld felül a `getMeret` metódust. Mivel a csavarfej cserélhető, ezért annak mérete nem számít bele a csavarhúzó méretébe. Ennek megfelelően működjön a méretet visszaadó függvény. A konstruktorban érkező méret a csavarhúzó teljes mérete.

Jó munkát!