

Programozás II. Házi Feladat

SZTE Szoftverfejlesztés Tanszék

2022. ősz

Ismertető

- A programot C++ nyelven kell megírni.
- **A benyújtandó fájl neve kötelezően feladat.cpp.**
- A megoldást a *Bíró* fogja kiértékelni.
 - A Feladat beadása felületen a Feltöltés gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a *Bíró* a programot g++ fordítóval és a
`-std=c++1y -static -O2 -DTEST_BIRO=1`
paraméterezéssel fordítja és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 5 másodpercnél és hiba nélkül (0 hibakóddal) fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:
 - Futási hiba 6: Memória- vagy időkorlát túllépés.
 - Futási hiba 8: Lebegőpontos hiba, például nullával való osztás.
 - Futási hiba 11: Memória-hozzáférési probléma, pl. tömb-túindexelés, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők 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/IBL302g-1/1/hXXXXXX/4/riport.txt`
 3. Az url-ből visszatörölve a 4-esig (`riport.txt` törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 20 alkalommal lehet benyújtani, a megadott határidőig.
- A programban szerepelhet `main` függvény, amely a pontszámításkor nem lesz figyelembe véve. Azonban ha fordítási hibát okozó kód van benne az egész feladatsor 0 pontos lesz.
- A megvalósított függvények semmit se írnak ki a standard outputra!



VilagitoDisz

1. feladat (4 pont)

Készíts egy VilagitoDisz nevű osztályt, mely egy karácsonyfa díszítésére alkalmas! Adattagjai (az 1. táblázat) legyenek private láthatóságúak!

Adattag neve	Típusa	Jelentése	Getter neve	Default érték
fenyesség	unsigned	A dísz fényessége	get_fenyesség	-
bekapcsolva	bool	A dísz állapota, világít-e vagy sem	is_bekapcsolva	false

1. táblázat. VilagitoDisz adattagok

A `get_fenyesség` metódus adja vissza a fényességet, ha a dísz be van kapcsolva! Ha nincsen, akkor nullát!

Készítsd el a VilagitoDisz konstruktorát is, mely a fényességet várja és állítja be. Default konstruktorként (default paraméter érték) a 0 értéket állítsa be!

2. feladat (3 pont)

Definiáld felül a VilagitoDisz `pre ++` és `pre --` operátorát! A `++` operátor kapcsolja be a világítást, a `--` operátor kapcsolja ki azt. Az operátorok `pre` verzióként működjenek, azaz a módosított értékkel az eredeti objektumot adják vissza!

KisKaracsonyfa

3. feladat (4 pont)

Készíts egy KisKaracsonyfa nevű osztályt! Adattagjai (a 2. táblázat) legyenek privátok!

Adattag neve	Típusa	Jelentése	Getter neve
csucs_disz	VilagitoDisz*	KisKarácsonyfára helyezhető egyetlen dísz	get_csucs_disz
fa_tipus	std::string	A fa típusa	get_fa_tipus

2. táblázat. Kolónia adattagok

Valósítsd meg a KisKaracsonyfa konstruktorát, mely a fa típusát várja! Default paraméter értéként a "luc" értéket add meg! A csucs_disz paramétert nullptr -el inicializálja!

4. feladat (4 pont)

Írj egy **disz_felhelyezese** metódust, mely egy VilagitoDisz-t vár paraméterben, figyelj annak pontos típusára! A metódus törölje a fán lévő díszet (ha van) és foglaljon dinamikusan egy új díszet, lemásolva a paraméterben kapottat. A metódus ne térjen vissza semmivel.

Tipp: másoló konstruktor hívás.

Mivel az osztályban dinamikusan kezelünk memóriát, gondoskodj a helyes memóriafelszabadításról is az objektum megszűnésekor!

5. feladat (4 pont)

Legyen egy **void bekapcsol()** és egy **void kikapcsol()** metódus, melyek rendre be vagy kikapcsolják a csúcsdíszet, ha van a fán!

6. feladat (2 pont)

Legyen egy **get_fenyesség** metódus, mely a fa aktuális fényességét határozza meg! Ha nincsen csúcsdísz a fán, adjon vissza nullát, különben a dísz fényességét! Visszatérési típusát a csúcsdísz metódusához igazítsd! (Egyezzen meg vele!)

7. feladat (3 pont)

Valósítsd meg a KisKaracsonyfa másoló konstruktorát, mely az adattagokat lemásolja! Figyelj a dinamikusan foglalt adattagok helyes kezelésére!

8. fealdat (4 pont)

Valósítsd meg a értékadás (assignment) operátort! Az operátor kezelje a lehetséges hibákat és hasonlóan működjön a másoló konstruktorhoz! Figyelj a dinamikusan foglalt adattagok helyes kezelésére!

NagyKaracsonyfa

9. feladat (4 pont)

Valósítsd meg a NagyKaracsonyfa osztályt! Adattagjai (a 3. táblázat) legyenek private láthatóságúak!

Adattag neve	Típusa	Jelentése	Default érték
diszek	VilagitoDisz*	NagyKaracsonyfára helyezhető díszek tömbje	get_diszek
diszek_szama	unsigned	A fára helyezhető díszek száma	-
act	unsigned	A tömb üres pozícióját jelző változó	0

3. táblázat. NagyKaracsonyfa adattagok

Legyen egy konstruktora, mely a díszek számát várja! Állítsa be az értékeket és a megadott elemszámú memóriát foglaljon VilagitoDisz-eknek.

A dinamikus memória használat miatt ügyelj az objektum életének végekor a megfelelő memória felszabadításra!

10. feladat (2 pont)

Legyen egy void **disz_felhelyezese** metódus, mely az aktuális pozícióra a paraméterben kapott VilagitoDisz-t helyezi el! Az üres hely pozícióját frissítse! Ha nem fér el több dísz a fán, ne tegyen semmit!

11. feladat (4 pont)

Legyen egy **void bekapcsol()** és egy **void kikapcsol()** metódus, melyek rendre be- és kikapcsolják az éppen fán lévő díszeket (mindet)! Csak azokat, melyeket már felraktunk a fára!

12. feladat (2 pont)

Legyen egy unsigned **get_fenyesség** metódus, mely a fa fényességét adja meg! A fa fényessége az éppen rajta lévő díszek fényességének összege.

13. feladat (4 pont)

Valósítsd meg a NagyKaracsonyfa másoló konstruktorát! Figyelj oda a dinamikus elemek helyes másolására!

14. feladat (4 pont)

Valósítsd meg az értékadás (assignment) operátort! Figyelj a lehetséges hibákra, és az operátor hasonlóan működjön a másoló konstruktorhoz! Figyelj oda a dinamikus elemek helyes másolására!