

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!

1. Feladat: MediaTartalom (12 pont)

Készítsd el a **MediaTartalom** absztrakt osztályt. Az osztály adattagjait az alábbi táblázat mutatja be:

Adattag neve	Típusa	Jelentése	Getter neve
nev utvonal	szöveg szöveg	a médiatartalom (fájl) neve a fájl útvonala	get_nev -

Az adattagok legyenek `protected` láthatóságúak. A getter metódus működjön konstans objektumon is!

Készíts egy konstruktort, ami az alábbi sorrendben várja a paramétereket: név, útvonal. Az útvonal legyen opcionális, ha nem adjuk meg, az értéke egy üres szöveg legyen.

Készíts egy tisztán virtuális metódust `get_meret` néven, ami konstans objektumon is működjön, és `double` értékkel térjen vissza.

Készítsd el a `fajl_utvonala` nevű metódust, ami a fájl teljes elérési útvonalával tér vissza (szöveg): `"utvonal/nev"` (tehát a két adattag egy `"/"` jellel legyen összefűzve).

Valósítsd meg a `/ =` operátort, ami nem tér vissza semmivel, és egy szöveg vár paraméterben. A kapott szöveget fűzze hozzá az eddigi útvonalhoz, `"/"` jellel elválasztva.

Készítsd el a `hibas` metódust, ami nem vár paramétert, logikai értékkel tér vissza, konstans objektumon is működik. A metódus adja vissza, hogy hibás-e a médiatartalom. A médiatartalom akkor hibás, ha a neve és az útvonala is üres sztring.

2. Feladat: Fenykep (23 pont)

Készítsd el a **Fenykep** nevű osztályt, ami publikusan öröklődik a *MediaTartalom* osztályból.

Az osztály adattagjait az alábbi táblázat mutatja be:

Adattag neve	Típusa	Jelentése	Getter neve
meret_x meret_y	egész egész	a kép szélessége (pixelben) a kép magassága (pixelben)	get_meret_x get_meret_y

Az adattagok legyenek `privát` láthatóságúak.

Készíts két darab konstruktort az osztályhoz:

- **Paraméteres konstruktor:** a paramétereket az alábbi sorrendben fogadja a függvény: név, szélesség, magasság, útvonal. Az útvonal természetesen itt is opcionális legyen, ha nem adjuk meg, üres szöveg legyen. A konstruktor inicializálja az ősét megfelelő módon, majd pedig a saját adattag értékeket is megfelelően állítsa be.
- **Default konstruktor:** az ős adattagjait üres szöveggel inicializálja, a saját adattagokat (a két méret) pedig 0 értékkel.

Definiáld felül az örökölt `get_meret` metódust, ami az értékeket konvertálja kilobyte-ra. A méret a pixelek száma hárommal megszorozva, majd kilobyte-tá konvertálva:

$$\frac{(meret_x * meret_y) * 3}{1000}$$

Figyelj rá, hogy az eredmény megfelelő lebegőpontos legyen.

Definiáld felül az örökölt `hibas` metódust. Egy kép akkor hibás, ha a neve és az útvonala is üres illetve a szélessége és magassága is -1 .

Valósítsd meg az egyenlőség operátort (`==`). Paraméterben egy másik fénykép érkezik, és a függvény döntse el, hogy megegyeznek-e az objektumok. Két fénykép akkor egyezik meg, ha a méretük is azonos, valamint a teljes elérési útvonaluk is megegyezik (`fajl_utvonala`).

Valósítsd meg a nem egyenlő operátort (`!=`), ami egy másik *Fenykep* objektumot vár paraméterben. A függvény pontosan akkor adjon vissza igazat, ha a két objektum nem egyenlő.

Készítsd el a prefix `++` operátort. Az operátor növelje meg a szélességet és a magasságot is 10 pixellel, és térjen vissza az aktuális objektummal.

3. Feladat: Memoriakartya (20 pont)

Készítsd el a *Memoriakartya* nevű osztályt. Az osztály adatait az alábbi táblázat mutatja be:

Adattag neve	Típusa	Jelentése	Getter neve
jelenlegi tartalom	előjel nélküli egész <code>Fenykep[10]</code>	jelenleg hány kép van a kártyán a tárolt képek	- -

Az adattagok láthatósága `protected` legyen. **A jelenlegi adattag értéke kezdetben mindig 0 legyen.**

Valósítsd meg a `+=` operátort is, ami egy fényképez kap paraméterben, és az aktuális objektummal térjen vissza a függvény. A működése az alábbiak szerint alakuljon:

- Amennyiben betelt a memóriakártya, dobjunk `std::exception` hibát.
- Amennyiben van hely, tegyük bele a paraméterben érkező fényképet a tartalom tömbbe (a végére), növeljük a jelenlegi értéket.

Készítsd el a `szektorhiba` nevű függvényt, ami visszaadja, hogy szektorhibás-e a memóriakártya (értelemszerűen a visszatérési értéke `bool`). Egy memóriakártya csak akkor szektorhibás, ha van rajta legalább egy hibás kép (bejárás során használhatod a *Fenykep* osztály `hibas` metódusát).

Készítsd el az `ls` nevű függvényt, ami a memóriakártya tartalmát listázza és egy szöveggel tér vissza az alábbi módon: a szöveg tartalmazza a memóriakártyán lévő összes *Fenykep* objektum `fajl_utvonala` metódus által visszaadott szöveget. Az útvonalakat sortörés karakter (`\n`) válassza el egymástól. A metódus működjön konstans objektumon is.

Készítsd el a tömbindex operátor metódust is. A metódus egy előjel nélküli értéket várjon paraméterben, és egy fényképpel térjen vissza. Ez a metódus is működjön konstans objektumon is. A működése az alábbiak szerint alakuljon:

- Amennyiben nem megfelelő indexet adunk meg, dobjunk *std::exception* kivételt.
- Minden más esetben térjen vissza a függvény az adott indexen lévő elemmel.