

Programozás II. ZH

SZTE Szoftverfejlesztés Tanszék

2024. ősz

Technikai ismertető

- A programot C++ nyelven kell megírni.
- 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++20 -Wall -Werror -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:
- 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.

Általános követelmények, tudnivalók

- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz pont.
- Minden metódus, amelyik nem változtatja meg az objektumot, legyen konstans! Ha a paramétert nem változtatja a metódus, akkor a paraméter legyen konstans!
- string összehasonlításoknál 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 az 3 alma, de a szóköz szükséges!
 - A tesztesetekben nem lesz ékezetes szöveg kiírása.
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).
- Ha az objektum másolása nem triviális (azaz a fordító által generált másolás nem elegendő), akkor a megfelelő másolást is meg kell valósítani.

Kiindulási projekt, megoldás feltöltése

- **A megoldáshoz az előre kiadott osztályok módosítása szükséges lehet.**
 - Nem minden ZH esetében van kiindulási projekt.
- Feltöltéskor ezeket az osztályokat is fel kell tölteni és a módosításokat is pontozhatja a bíró!
- Egyes tesztesetekben a bíró módosított osztályt is használhat ezen kiinduló osztályok helyett, ezzel tesztelve a valóban helyes működést!

Zh alatt használható segédanyag

- A ZH során használható segédanyag elérhető bíróban.
 - <https://biro.inf.u-szeged.hu/kozos/prog2/>

Kiindulás

A kiindulási feladatban kapott osztály módosítása vagy bővítése szükséges lehet!

Gyerek

A Gyerek osztály egy gyereket reprezentál.

- Legyen default konstruktora, ami a gyerek nevét "gyerek" szóval inicializálja.
- Legyen egy olyan konstruktora is, amelyikben meg lehet adni a gyerek nevét.
- A gyerek nevét le lehessen kérdezni.
- A gyereknek van boldogság szintje, ami létrehozáskor 0, és legyen hozzá getter és setter.

Ajándék

Készíts egy `Ajandek` nevű osztályt, mely későbbi ajándékok ősosztálya lesz!

- Legyen egy konstruktora az Ajándéknak, mely egy üzenetet vár (`std::string`)
- Legyen egy `ajandekoz` metódus!
 - A metódus egy Gyereket vár és kiírja a képernyőre az üzenetet sortöréssel a végén.

Vonat

Készíts egy `Vonat` osztályt, mely egy Ajándék!

- Lehessen példányosítani a Vonatot üzenet és boldogságérték (`int`) megadásával! A boldogság legalább 1 értéket vesz fel.
- Az `ajandekoz` metódus az őshöz hasonlóan kiírja az üzenetet, MAJD a kapott gyereknek beállítja a vonat boldogságértékét.

Virgács

Készíts egy `Virgacs` nevű osztályt, mely egy Ajándék!

- Lehessen default módon példányosítani a Virgácsot. Minden virgács üzenete: "jaj"!
- Az `ajandekoz` metódus kiírja az üzenetet az őshöz hasonlóan MAJD minden pozitív boldogsággal rendelkező gyerek boldogságát felére állítja be. Egyébként kétszeresére módosítja a boldogságértékét.

Feladat

Mikulás

Készíts egy `Mikulas` nevű osztályt!

- Legyen default konstruktora.
- A `Mikulás` a `<<` operátor segítségével kap egy ajándékot, melyet lemásol és letárolja azt.
 - A tárolás a hozzáadás sorrendjében történjen!
 - Legyen láncba fűzhető az operátor.
- Legyen egy `>>` operátor, mely egy gyereket kap és megajándékozza azt a gyereket.
 - A `Mikulás` a nála lévő ajándékok közül az utolsót adja oda a gyereknek.
 - * Mivel odaadja a gyereknek, így nála már nem lesz ott, így **kikerül a hátizsákjából**.
 - A megajándékozás azt jelenti, hogy az `Ajandek ajandekoz` metódusát kell meghívni.
 - A `Mikulás` „mindenkinek” csak egyszer ad ajándékot.
 - * A `Mikulás`nak csak a neveket tudja megjegyezni, így adott névhez csak egyszer ad ajándékot, akkor is, ha két különböző gyereket hívnak ugyanúgy.
 - Ha nincs ajándéka vagy ilyen nevű gyereknek már adott ajándékot, akkor **nem csináljon semmit**.
 - Az operátor legyen láncba fűzhető.
- A `<<` operátor legyen megvalósítva `Mikulás`ra is úgy, hogy átveszi a másik `Mikulás` ajándékait (a saját ajándékkészletét bővítve), valamint a gyerekek listáját is bővíti.
- Legyen helyesen megoldva a memóriakezelés!