Programozás II. 1. ZH

SZTE Szoftverfejlesztés Tanszék

2022. 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++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-túlindexelés, null pointer használat.
- 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://biro.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.

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 bemutat példákban a string-ek 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 kiíratása.
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).

Karácsonyi készülődés

Ügyelj rá, hogy minden olyan metódus konstans legyen, ami nem módosít az adattagok értékein! Ha egy metódus nem változtat a paraméterén, akkor az a paraméter legyen konstans!

1. feladat: Tél (5 pont)

A `Tel` osztály amely a telet reprezentálja. Ismert az átlagos hőmérséklete, valamint hogy melegnek számít-e. Az osztálynak van egy 1 paraméteres konstruktora.

Módosítsd az osztály konstruktorát! Ha a konstruktor hőmérsékletként 10-nél nagyobb értéket kap, akkor a létrejövő tél melegnek számít, különben nem. **(0+1 pont)**

Az osztály szintén tartalmaz egy számláló adattagot, aminek az lenne a feladata, hogy számontartsa, hogy a getter metódust hányszor hívtuk meg.
Oldd meg a **getter fejlécének megváltoztatása nélkül**, hogy a számláló értéke növekedjen 1-el, amikor a gettert meghívjuk. **(0+2 pont)**

Oldd meg, hogy amikor az objektum megszűnik, akkor ellenőrízzük, hogy a számláló értéke nagyobb-e mint 3, és ha igen, akkor írjuk ki a képernyőre, hogy *"Itt a tavasz"* idézőjelek nélkül, sortöréssel a végén.
Figyelem! Ennek a metódusnak a helyes megvalósítása szükséges számos másik teszteset megfelelő futtatásához!

(0+1 pont)

Előfordulhat, hogy a télből egy osztály öröklődni fog, és a gyerekosztályban más lesz az elvárt viselkedés az objektum megszűnésekor. Oldd meg, hogy ilyen esetekben a gyerekosztályban definiált viselkedés történjen meg. **(0+1 pont)**

2. feladat: Apó (3 pont)

A `Apo` osztály egy öreg bácsit reprezentál, és csak a nevét ismerjük. Az osztály egy 1 paraméteres konstruktorral rendelkezik.

A konstruktor nem elég alaposan lett megírva, hiszen rosszul gazdálkodik a memóriával. Optimalizálj a konstruktor memóriahasználatán.
Ügyelj rá, hogy az alábbi példányosítás továbbra se dobjon fordítási hibát:

```
int main() {
    const string nev = "Bela";
    Apo apo(nev);
}
```

(0+2 pont)

Oldd meg, hogy ha az apó objektumot polimorfikusan hozunk létre, akkor is a gyerekosztály destruktora hívódjon meg, ne csak az ősosztályé. **(0+1 pont)**

3. feladat: Télapó (6 pont)

Készítsd el a `TelApo` osztályt, amely egy télapót fog reprezentálni, aki rendelkezik egy tél és egy apó tulajdonságaival is. Az öröklődést ez alapján készítsd el. **(0+2 pont)**

Adattagként eltároljuk, hogy hány ajándék van a télapónál. Ehhez az adattaghoz tartozik egy getter metódus is, ami ugyan működik, de nem minden objektum esetén. Oldd meg, hogy a getter működjön a következő kódrészletben is.

```
int main() {
    const TelApo telapo;
    telapo.get_ajandekok();
}
```

A programot csak 1 helyen írd át. **(0+1 pont)**

Készíts az osztályhoz egy default konstruktorát, amely a télapó hőmérsékletét 36-ra állítja, tehát a télapó meleg lesz. A neve legyen "Miklos" (idézőjelek nélkül), és nincs nála ajándék.
Hint:
A gyerekosztály konstruktorában mit szoktunk először csinálni? **(0+1 pont)**

Készíts az osztályhoz egy 3 paraméteres konstruktort, amely a télapó 3 tulajdonságát várja (az alábbi sorrendben): a hőmérsékletét, a nevét, illetve az ajándékainak a számát. A konstruktor állítsa be az adattagokat ezen értékek alapján.

(1+1 pont)

4. feladat: Mikulás találkozó (22 pont)

Készítsd el a `MikulasTalalkozo` osztályt, amely egy találkozót reprezentál, ahol a leendő télapók találkoznak.

Az osztályban tároljuk el a részt vevő **télapókat** egy dinamikus tömbként. Az adattag neve legyen *telapok*. **(0+1 pont)**

A lent leírt feladatok megoldása érdekében tetszőleges további adattagok létrehozhatóak!

Készítsd el az osztály konstruktorát, ami paraméterben egy egész számot vár, hogy legfeljebb hány tépaló vehet részt a találkozón. A konstruktorban ez alapján foglaljuk le a megfelelő mennyiségű memóriát a *telapok* tömbnek. **(0+1 pont)**

Definiáld felül az osztályban a *+= operátor*t, amely segítségével egy új télapót lehet hozzáadni a találkozóhoz. Tegyük be a kapott télapót a tömb a legelső üres pozíciójára! Amennyiben a találkozó már megtelt (tehát nem fér több elem a tömbbe), akkor a tömb méretét növeljük meg 2-vel, és ezután tegyük bele a télapót a tömbbe. **(1+4 pont)**

Figyelem! Ennek a metódusnak a helyes megvalósítása szükséges számos másik teszteset megfelelő futtatásához! Legalább az alap esetet készítsük el.

Hozd létre a *felez* metódust, ami a találkozó létszámát - így a télapók tömböt is - a felére fogja csökkenteni. A találkozón jelen levő télapók közül csak **minden második** fog a tömbben maradni, valamint a tömb hossza is a felére csökken. Feltételezhetjük, hogy a tömb hossza páros szám.

Például, ha a tömbben a következő mikulások voltak (pszeudokód, csak szemléletelésnek):
`["Bela", "Miklos", "Ferenc", "Elemer"]`
akkor a változtatás után az alábbi módon fog kinézni a tömb:
`["Miklos", "Elemer"]` **(0+5 pont)**

Mivel az osztály tartalmaz dinamikusan foglalt memóriát, így néhány egyéb metódust is meg kell valósítanunk ahhoz, hogy a megfelelő viselkedést kapjuk, azaz egyrészt minden **lefoglalt memória fel legyen szabadítva**, másrészt az **objektumok lemásolása** megfelelően működjön, azaz egymástól független másolatokat tudjunk készíteni. Valósítsd meg az összes olyan metódust az osztályban, amely ehhez a működéshez szükséges! **(0+10 pont)**

5. feladat: Statisztika (2 pont)

Készítsd el a *statisztika* nevű függvényt a globális névtérben, amely fejléce az alábbi módon néz ki:

```
float statisztika(TelApo** telapok);
```

A paramétere egy télapó **pointerekből** álló tömb. A tömb utolsó elemét nullpointer jelzi.

A függvény számolja ki a kapott télapók ajándékainak számának az átlagát, és térjen vissza vele. Ügyelj a megfelelő típusokra.

(0+2 pont)