

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 pontosan a bírónak!
- 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!

Verseny osztály

- A **Verseny** osztály egy versenyt reprezentál, aminek van neve.
- A nevet a konstruktorban megkapja és eltárolja.
- Legyen egy **leiras** metódusa, ami string formában adja vissza a verseny leírását, ami a nevét jelenti.

FutoVerseny osztály

- A **FutoVerseny** egy olyan verseny, ami a futás távolságát (egész szám) is eltárolja.
- A konstruktorban a nevet és a távolságot kell megadni.
- A futóverseny leírása a nevéből és a távolságból áll, amiket szóközők és egy "-" jel választ el. Például: "Szeged Maraton - 42"
 - Definiálja felül az **ősosztályból** származó metódust.

HegymaszoVerseny osztály

- A **HegymaszoVerseny** egy olyan verseny, ami tárolja a megmászkodó hegy nevét is.
- A konstruktor a verseny nevét és a hegy nevét várja paraméterként.
- A leírása a verseny és a hegy nevét tartalmazza, amiket szóközők és egy "-" jel választ el. Például: "Himalája kihívás - Himalája"
 - Definiálja felül az **ősosztályból** származó metódust.

Feladatok

Versenyek osztály

- A **Versenyek** osztály összegyűjti a különböző versenyeket és eltárolja.
- Rendelkezzen default konstruktorral.
- A **<<** operátor egy **Verseny** pointert kap paraméterben, és a pointer által mutatott objektumot el kell tárolnia.
 - A pointer által mutatott objektum biztosan dinamikusán foglalt.
 - A **Versenyek** osztály felel a paraméterben kapott objektum felszabadításáért.
 - A versenyeket eredeti sorrendjükben kell eltárolni.
 - Legyen láncba fűzhető.

- Valósítsd meg a `kiir` metódust, ami a versenyek leírását írja ki, mindegyiket külön sorba (és az utolsó sor után is kell sortörést tenni).
- Lehesse másolni az objektumot.
 - Valósítsd meg a másoló konstruktort és az értékadó operátort.

UltraFutoVersenyek osztály

- Az `UltraFutoVersenyek` osztály a `Versenyek` osztályból származik.
- Legyen default konstruktora.
- Definiálja felül a `<<` operátort úgy, hogy az csak olyan futóversenyeket adjon hozzá, amelyeknek a távolsága legalább 42.
 - Ha ennek a kritériumnak nem felel meg, akkor ne adj hozzá, de **nem kell** semmi hibajelzést sem adni.
 - Egyébként a működése feleljen meg az `ősosztály` operátorának.
- Definiáld felül a `<<` operátort úgy, hogy `Versenyek` paramétert is elfogadjon. Ilyenkor a versenyekből csak azokat vegye át, amik futóversenyek és legalább 42 a hosszuk.
 - A versenyeket le kell másolni és az osztály felel a felszabadításért.