

Programozás II. Gyakorló Feladat

SZTE Szoftverfejlesztés Tanszék

2023. 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-túlinde克斯, 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!

Feladatsor

1. feladat (4 pont)

Készíts egy `Etel` nevű osztályt!

- Az ételnek legyen neve (string), leírása (string) és ára ami egy pozitív egész szám.
- Legyen egy default konstruktor, ami az árat 0-ra inicializálja, a név "`amint kész megtudod`" és leírás "`ami sikerul`".
- Legyen egy 3 paraméteres konstruktor is, aminek megadhatjuk a nevet, leírást és az árat (ebben a sorrendben).
- Legyen egy `get_nev()` metódusa, amelyik visszaadja az étel nevét string formátumban.
- Legyen egy `get_leiras()` metódusa is, amelyik visszaadja az étel leírását string formátumban.
- Valósítsd meg azt az operátort, ami az ételt számmá (unsigned) konvertálja.

2. feladat (4 pont)

Készíts egy `Menu` nevű osztályt, amelyik az `Etel` osztályból származik publikusan!

- A menü célja, hogy az ételekből össze tudjunk állítani egy menüt, ami jelen esetben 3 `Etel`-ből áll. Valósítsd meg ennek az eltárolását.
- Legyen az osztálynak default konstruktor, ami nem csinál semmit sem, és az `Etel`-t is a default konstruktorral inicializálja.
- Legyen egy kétparaméteres konstruktor, aminek megadhatjuk a menü nevét és leírását, de az árat ez is állítsa 0-ra.
- Legyen egy olyan konstruktor is, ahol a név és leírás mellett további 3 paraméter segítségével a menübe tartozó ételeket megadhatjuk.
- Definiáld felül a `get_nev()` metódust úgy, hogy a menü neve elé írja oda, hogy "`Menu:`" és azzal térjen vissza.
- Definiáld felül az egész számmá konvertáló operátort úgy is: a menü ára a 3 étel árának az összege, azonban jár 10% kedvezmény. Például, ha a 3 étel ára 800, 1000 és 1200, akkor a menü ára 2700.
- Valósítsd meg az indexer operátort úgy, hogy helyes index esetén a menü i-edik ételét adja vissza. Rossz index esetén `std::exception`-t kell dobni.

3. feladat (4 pont)

Legyen egy `Vendeglo` osztály, aminek a következőket kell tudnia:

- Pontosan 7 `Etel`-t képes eltárolni.
- Legyen default konstruktora.
- Legyen egy olyan konstruktora, ami egy 7 elemű ételekből álló tömböt kap, és azzal inicializálja a vendéglőt.
- Az indexer operatora (`size_t` paraméter esetén) úgy adja vissza az *i*-edik ételt, hogy azt lehessen módosítani. Ha rossz indexet kap, akkor dobjon `std::exception`-t.

4. feladat (4 pont)

A `Gyorsetterem` osztály az `Vendeglo` osztályból származzon publikusan.

- A gyorsétteremben vehetünk menüket is, és pontosan 3 féle menüből választhatunk.
- Legyen egy default konstruktora.
- Legyen egy olyan konstruktora, amelyik 3 menüt vár (3 elemű `Menu` tömböt), és ezzel inicializálja a választható menüket.
- Legyen egy olyan konstruktora, amelyik 7 ételt kaphat (7 elemű `Etel` tömb), és ezzel inicializálja a választható ételeket.
- Legyen egy olyan konstruktora is, amelyiknek meg lehet adni az ételt és a menüt is az előzőekhez hasonlóan.
- Definiáld felül az indexer operátorát, hogy (`size_t` paraméter esetén) adja vissza az *i*-edik menüt úgy, hogy azt lehessen módosítani.

