

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!

Feladatsor

1. feladat (3 pont)

Készíts egy `Etel` nevű **struktúrát**! Két **publikus** adattagja van:

Adattag neve	Típusa	Jelentése	Getter neve	Setter neve
nev	string	Az étel neve	-	-
hutes	int	Az étel hűtési igénye	-	-

1. táblázat. Étél adattagok

Legyenek konstruktorai is!

- Legyen egy paraméteres konstruktor, melynek első paramétere egy string, a második egy int és beállítja a két adattagot.
- Legyen egy default konstruktor, amely a sztringet üresre, az int-et 0-ra állítja be

2. feladat (2 pont)

Készíts egy `string print() const;` publikus metódust az `Etel` struktúrához. A visszaadott sztring tartalma az alábbi legyen: `"nev:<nev>,hofok:<hutes>"`. A `<nev>` és a `<hutes>` helyettesítődjenek az adattagokkal, a vessző után **nincs** szóköz.



3. feladat (6 pont)

Készítsd el a program hibaosztályait! Minden osztályban az adattagok, ha vannak, legyenek privát láthatóságúak. A metódusok public-ok legyenek.

- class **Hiba**. Ősosztály. Az alábbiakkal rendelkezik:
 - Egy virtuális, stringet visszaadó metódus (`string hiba() const;`), mely ezt a sztringet adja vissza: `"Hiba történt"`.

- class **Megromlik**. Publikusan öröklődik a Hiba őszosztályból. Akkor kell dobni, ha az ételt rossz hőfokon tároljuk le. Az alábbiakkal rendelkezik:
 - Egy privát, Etel típusú adattag (neve `etel`)
 - Egy konstruktor, amely egy Etel-re mutató konstans referenciát vár paraméterül és inicializálja az adattagot.
 - Felüldefiniálja az őse `hiba()` metódusát úgy, hogy az alábbi sztringet adja vissza:
 - * "A <etel.nev> tarolasahoz legalabb <etel.hutes> fok kell"
 - * Ahol a kacsacsőrök közötti részek helyettesítődnek a hiba objektumban letárolt etel megfelelő adattagjaival
- class **Megtelt**. Publikusan öröklődik a Hiba őszosztályból. Az alábbiakkal rendelkezik:
 - Egy privát, string típusú adattag (neve `mit`)
 - Egy konstruktor, amely egy string-et vár paraméterül és beállítja az adattagot.
 - Felüldefiniálja az őse `hiba()` metódusát úgy, hogy az alábbi sztringet adja vissza:
 - * "Nem sikerult letarolni a <mit>-t"
 - * Ahol a kacsacsőrök közötti rész helyettesítődik az adattaggal

4. feladat (4 pont)

Készíts egy Hutoegyseg nevű osztályt! A hűtő adattagjait a 2. táblázat foglalja össze. Az adattagok legyenek privát láthatóságúak. Mindegyik rendelkezzen a táblázat szerinti getterrel és setterrel. Az osztály minden gettere legyen használható konstans objektumon is!

Adattag neve	Típusa	Jelentése	Getter neve	Setter neve
hofok	int	Hűtési hőfok	<code>get_hofok</code>	-
tartalom	Etel[10]	Az egységben tárolt ételek, max 10	<code>get_tartalom</code>	-
aktualis	unsigned	Aktuálisan tárolt étel mennyiség	<code>get_aktualis</code>	-

2. táblázat. Hűtőegység adattagok

Készítsd el a Hutoegyseg osztály konstruktorát is! Egy **string** paramétert vár, amely a hűtőegység hűtési hőfokát reprezentálja sztringes formában. Konvertáld intté a kapott értéket, de figyelj oda rá, hogy hiba esetén a konstruktor ne dobjon kivételt, hanem állítsa be -1-re a hőfokot! Az aktuális méret kezdetben 0.

5. feladat (5 pont)

Valósítsd meg a `+=` operátort a Hutoegyseg osztályban. Paramétere egy Etel legyen! Az operátor feladata, hogy hozzáadja a paraméterben kapott ételt a tartalom tömbhöz. Legyen benne hibakezelés is! Először a hőmérséklet megfeleltetése legyen vizsgálva. Az étel csak akkor tehető be a hűtőegységbe, ha `hutes` értéke nagyobb vagy egyenlő, mint a hűtőegység által biztosított `hofok`. Ha ez nem teljesül, akkor legyen egy Megromlik kivétel dobva, mely kivétel

a paraméterben kapott étellel van inicializálva. Ha a hők feltétel teljesül, akkor az legyen ellenőrizve másodikként, hogy van-e még hely a tömbben. Ha nincs, akkor legyen egy Megtelt kivétel dobva az étel nevével inicializálva.

6. feladat (3 pont)

Készíts egy `print()` publikus metódust a `Hutoegyseg` osztályhoz. Feladata, hogy végigmenjen az aktuálisan tárolt ételeken és a standard outputra kiírja, amit az ételek `print()` metódusa visszaad. Az egyes `print` hívások után legyen mindig egy-egy soremelés is a standard outputra írva. Legyen Hiba típusú kivétel dobva, ha a hűtőegységben nincs étel tárolva.

7. feladat (4 pont)

Készíts egy globális (osztályokon kívüli)

```
bool feltolt(Hutoegyseg& h, Etel etelek[], unsigned etelszam)
```

függvényt, amelynek feladata, hogy feltöltse a hűtőt a `+=` operátor meghívogatásával. Menjen végig a függvény a paraméterben kapott tömb elemein. A hosszt a harmadik paraméter adja meg. Mivel a `Hutoegyseg` interfészéből nem kérdezhető le a kapacitás, így előzetes méretellenőrzést nem kell/nem lehet belerakni. Fontos, hogy a `feltolt` függvényből ne dobódjon kivétel. Ha hiba történne fejeződjön be a hűtő feltöltése és legyen `false` érték visszaadva. Ha minden étel letárolása sikerült, akkor legyen `true` visszaadva.