

OODB VIZSGA 2020-01-14

FIGYELEM! A vizsga megkezdése előtt ezt olvassa át.

- A különböző szintekhez (2,3,4,5) a `#define` makrókat kell állítani 0-ról 1-re.
- Ahol kiemeleten jelezve van, azokat a kódrészeket tilos megváltoztatni. Megváltoztatás esetén a vizsga sikertelen.
- A vizsga megvalósítható a *main.cpp* jelzett részein, illetve külön *.cpp*, *.h* és *.hpp* fájlokban.
- Memóriaszivárgás léte esetén a vizsga elégtelen (**delete** részek megváltoztatásával).
- Online (internet, stackoverflow, diasorok stb.) és offline (jegyzetek, könyvek) segédeszközök használhatóak. Ugyanakkor a munka nem kollaboratív, így más ember (padtárs, Skype, Viber, Facebook stb.) segítsége kizárva.
- A *main.cpp*-ben sok **assert** függvényhívást láthat. Ez egy makró, ami egy feltételt vizsgál és annak nem teljesülése esetén futásidejű hibát dob. Az Ön implementációjának tesztelésére és a jegy meghatározására szolgál. Az **assert** feltételek megváltoztatása **TILOS**. Az **assert** részek ideiglenesen az `#define ELLENORZES` makró 0-ra állításával kapcsolhatók ki, ami ajánlott is a fejlesztés során.
- **A szabályok megsértése a vizsga automatikus sikertelenségét vonja maga után.**

Specifikáció

A Galaktikus Birodalom új csillagrombolókat készül építtetni az új kísérleti üzemében. Vader nagyúr szeretné nyilvántartásba venni az eddig elkészült rombolókat, erre biztosított egy C++ programleírást.

2-es jegyért (SZINT_2)

- Definiálja a **CsillagRombolo** osztályt! Az osztálynak legyen egy konstans neve. Az **Imperial** és az **Executor** osztályokat ebből származtassa. Konstansként legyen definiálva a maximális/minimális legénység és maximális/minimális turbólézerszám. Az osztály jegyezze mennyi az aktuális legénység és turbólézer. A **CsillagRombolo** ne legyen példányosítható.
 - o Implementálja a következő absztrakt metódust:
 - **int tuzero()**
 - o Implementálja a következő segédmetódusokat a **CsillagRombolo** osztályhoz
 - **unsigned int getLegenyseg()**: visszatér az aktuális legénység számával
 - **std:string getName()**: visszatér a hajó nevével
 - **void addLegenyseg(unsigned int legenyseg)**: hozzáad **legenyseg** számú legénységet
 - **void removeLegenyseg(unsigned int legenyseg)**: eltávolít **legenyseg** számú legénységet
 - **bool checkUzemkepess()**: Vagyis: `aktualis_legenyseg >= minimalis_legenyseg`?
 - **unsigned int getMaxLegenyseg()**: visszatér a hajóhoz rendelhető maximális legénységszámmal
 - **unsigned int getMinLegenyseg()**: visszatér a hajóhoz rendelhető minimális legénységszámmal
- Készítse el a két származtatott csillagromboló típus osztálydefinícióját: **Imperial**, **Executor**!

- **Imperial-osztály:** Konstruktor megadja a nevet, aktuális legénységszámot és aktuális turbólézerszámot.
 - Konstruktorban állítsa be a minimális (29000) és maximális (45000) legénységszámot.
 - A konstruktorban állítsa be a minimális (40) és maximális (80) turbólézerszámot.
 - Megfelelő számú legénység jelenléte esetén: **Tűzerő** $100 * \text{turbolézerszam}$
- **Executor-osztály:** rendelkezik ion ágyúkkal is, ezt jelezze konstans attribútumokkal. Konstruktor megadja a nevet, az aktuális legénységszámot, aktuális turbólézerszámot és aktuális ionágyúszámot.
 - A konstruktorban állítsa be a minimális (1500) és maximális (2500) turbólézerszámot.
 - A konstruktorban állítsa be a minimális (50000) és maximális (350000) legénységszámot.
 - A konstruktorban állítsa be a minimális (200) és maximális (300) ionágyú számot.
 - Megfelelő számú legénység jelenléte esetén a **Tűzerő** $150 * \text{turbolézerszam} + 1000 * \text{ionagyuszam}$
- Készítsen egy flotta osztályt, ami tartalmaz csillagrombolókat! A **Flotta** csak egyszer példányosulhat, legyen ez az osztály singleton (ld. **Singleton** design pattern). A flotta tartsa számon a tartalék legénységet is, ami kezdetben 0.
 - Konstruktor nem hívható meg kívülről. Copy konstruktor és assignment legyen letiltva.
 - A **getInstance()** nevű statikus metóduson keresztül érhető el az egyetlen példányra mutató pointer. A **freeInstance** szabadítsa fel az egyetlen példányt
 - Az **addCsillagrombolo** metódussal lehet új csillagrombolót hozzáadni, egy **CsillagRombolo** mutatót várva argumentumként.
 - A flottából lekérdezhető egy csillagromboló a nevével (**getCsillagrombolo**). Ekkor térjen vissza az adott csillagrombolóra mutató pointerrel.
 - A flotta adja vissza, hogy aktuálisan hány darab csillagromboló van a flottában. (**getQuantity**)
 - A flottához lehessen hozzárendelni tartalék legénységet, aminek bemenő paramétere egy pozitív egész szám (**rendelTartalekLegenyseg**). Lehessen lekérdezni az aktuális tartaléklegénységet!

3-as jegyért (SZINT_3)

- Legénységet lehessen hozzáadni a flotta egy csillagrombolójához a Flotta **addLegenyseg** metódusával a flottához rendelt tartalék legénység számából. A metódus a csillagromboló nevét és egy darabszámot vár!
- Ugyanezt a függvényt valósítsa meg úgy is, hogy egy csillagrombolóra mutató pointert vár!
- Minden hozzárendelés csökkentse a flottához rendelt tartalék legénység számát!
- Ennek inverzét is valósítsa meg, a **Flotta tartalekbaHelyez** metódusával. A metódus várja a csillagromboló nevét és a darabszámot!
- Számolja azt is ki, hogy mekkora a flotta teljes legénysége! (**getPersonnel**) Ezt a csillagrombolókhoz aktuálisan rendelt legénység és a tartalék legénység összegével kapja meg.

4-es jegyért (SZINT_4)

- Legyen a flottának egy **checkImmobility** metódusa, ami ellenőrzi, hogy a flotta üzemképes-e (minden hajón rendelkezésre áll-e minimális legénység). Térjen vissza az első olyan hajónak a mutatójával, ami nem üzemképes, vagy nullpointerrel, ha nincs ilyen!
- A **addLegenysegChecked** metódus ugyanazt valósítsa meg, mint a **Flotta::addLegenyseg** (pointer argumentum), viszont dobjon kivételt, ha a hozzárendeléssel túllépnénk egy csillagromboló kapacitását (**ErrorLegenyseg**), vagy 0 alá csökkenne a flotta tartalékos legénysége (**ErrorTartalekos**).
- A **tartalekbaHelyezChecked** metódus ugyanazt valósítsa meg, mint a **tartalekbaHelyez** (pointer argumentum) metódus, viszont dobjon kivételt (**ErrorLegenyseg**), ha a csillagromboló minimális legénységkapacitása alá mennénk!
- A flottától elvonható tartalék legénység (**removeTartalekosChecked**), ellenőrizze, hogy ne csökkenjen 0 alá a legénység - ha 0 alá csökkenne, dobjon kivételt (**ErrorTartalekos**)!

5-ös jegyért (SZINT_5)

- A Flotta **addCsillagromboloChecked** metódusa ellenőrizze, hogy a hozzáadandó csillagromboló szerepel-e a flottában!
- Amennyiben szerepel, a legénységet rendelje hozzá a flottában szereplő csillagrombolóhoz! Más attribútum ne változzon!
- Ehhez definiáljon felül egy **operator==**-t, amivel ellenőrzi két csillagromboló azonosságát! Két csillagromboló azonos, ha ugyanaz a nevük!
- A csillagrombolókat lehessen std::ostream-re kiíratni! Ugyanígy, egy flotta listázható legyen, az összes csillagromboló felsorolásával! A kiíratásban legyen benne a név, aktuális legénység, min. legénység, max. legénység!

Jó munkát!