

4. LABOR

DINAMIKUS TAGVÁLTOZÓK, DINAMIKUS OSZTÁLYOK

Általános információk

1 iMSc pont jár arra, ha írásban válaszolsz az 1. és 2. feladat kérdéseire. A Person-nál fejtsd ki részletesen, hogy a konzolon kiírt log adott sora melyik utasítás hatására hívódott meg, és melyik objektumhoz tartozik.

Kötelező feladatok

1. Áttekintés: Person osztály

Vizsgáld meg a mellékelt *Person solution*-ön belüli *Person* osztályt.

1. Hogyan használ dinamikus tagváltozókat?
2. Hogyan lehet lemásolni egy már létező *Person* objektumot?
3. Mikor hívódik meg a konstruktor és a destruktork? Miért pont ez a sorrend?

2. VendingMachine osztály

Vesd össze a *VendingMachine solution* header-jét az alábbi specifikációval, azonosítsd melyik függvénynek mi a feladata:

- Az osztály egy italautomata működését szimulálja. Ingyen vásárolhatunk, és végtelen mennyiségű üdítővel fel lehet tölteni ☺
- Feltöltéskor adott nevű és mennyiségű ital kerül a gépbe, a sor végére
- Vásárláskor mindig a legrégebben berakott üdítőt adja ki a gép
- Egyszerre többet is lehet vásárolni, ilyenkor ugyanazt a műveletsort hajtja végre a gép ciklikusan

Az italokat egy *string* tömb reprezentálja. Ez már egy osztályként van megírva a C++ nyelvben.

- `std::string` néven tudunk rá hivatkozni, innentől ugyanúgy viselkedik, mint a megismert típusok
- Dinamikusan foglalunk megfelelő méretű tömböt az italok betárazásához
- A kapott kódrészletben keresd meg amikor C++ osztályra jellemző módon használjuk a `string`-et, és keresd meg hogyan lehet „kompatibilissé” tenni a hagyományos C nyelvben való használathoz (iMSc kérdés)
- Miért nem hatékony ez a megvalósítás, hogyan lehetne rajta javítani? (iMSc kérdés)

Feladatok:

- A *VendingMachine.cpp*-ben a meglévő kódok alapján írd meg a hiányzó *refill* függvény törzsét és a másoló konstruktort
- A *TestFile.cpp*-ben írd tesztkódot az automata különböző funkcióinak tesztelésére:
 - Hozz létre másfajta italokat is a Coca-cola alapján
 - Töltsd fel a gépet többféle itallal
 - Vásárolj belőle
 - Teszteld le funkcionálisan a másoló konstruktort

Gyakorló feladatok

- [Verem osztály alapvető funkcióinak elkészítése](#)
- [Optimalizált verem osztály](#)
- [Mátrix osztály](#)
- [Intervallumhalmazok](#)
- [Multihalmaz osztály tetszőleges számú elemmel](#)