

6. LABOR

OPERÁTOROK TÚLTERHELÉSE

Általános információk

1 IMSC pont jár a 2. és 3. feladatok helyes megoldására

Kötelező feladatok

1. Dinamikus tömb (*Vector*) osztály: közös értelmezés

Közösen értelmezzétek a megadott *Vector* solution állományait!

1. Hogyan lehetne leegyszerűsíteni a másoló konstruktor definiálását?
2. Miért (konstans) referenciatípus a paraméter és a visszatérési érték az egyes függvényeknél?
3. Mi a helyzet a $v=v$ kifejezés esetén?
4. Mivel térjen vissza a `[]` operátor?
5. Milyen változtatásokra lenne szükség, ha az `int` típusú változók helyett saját osztályt példányosító objektumokat tárolna a tömb?
6. Miért vannak külön névtérben az *ascend* és a *descend* predikátum függvények?
7. Miért nem statikus függvény a *sort*?
8. Miért *friend* a kiírató operátor?
9. Szükség van az *at* függvényekre?
10. Értelmezzétek a *sort* függvény pointer paraméterét!

2. Dinamikus tömb (*Vector*) osztály: tagfüggvények definiálása – első rész

Az alábbi függvényeket mindenkinek kötelező definiálnia:

- `ostream& operator<<(ostream& os, const Vector& v)`
- `Vector& Vector::operator=(const Vector& theOther)`
- `Vector::Vector(const Vector& theOther)`
- `Vector::~~Vector()`
- `unsigned Vector::size() const`
- `unsigned Vector::getMaxElementsNum() const;`
- `void Vector::clear()`
- `bool Vector::erase(unsigned position)`
- `int& Vector::at(unsigned position)`
- `const int& Vector::at(unsigned position) const`
- `bool Vector::insert(unsigned position, int element)`
- `int& Vector::operator[](unsigned position)`
- `const int& Vector::operator[](unsigned position) const`

javaslatok:

- A másoló konstruktor definiálásakor érdemes az operator=()-t felhasználni
- Az operator[]-nél pedig az at() függvényt célszerű használni
- sort()-nál egyszerű és elegendően hatékony a minimum kiválasztásos rendezés, de az insertion sort is alkalmas választás
- insert()-nél ügyelj a megfelelő szintű hibakezelésre
 - elemek maximális száma
 - mi történjen akkor, ha olyan indexre szúrunk be elemet, ami nagyobb, mint az aktuális elemszám?
- erase()-nél is fontos a hibakezelés
 - ha olyan indexről törölünk, amely nagyobb, mint az aktuális elemszám
 - ha csak egy elem van a vektorban

Opcionális feladat

3. Dinamikus tömb (Vector) osztály: tagfüggvények definiálása – második rész

Az alábbi függvények definiálása opcionális (de ajánlott):

- `void Vector::operator*=(unsigned right)`
- `void Vector::operator+=(unsigned right)`
- `istream& operator>>(istream& is, Vector& v)`
- `bool Sorters::ascend(const int& a, const int& b)`
- `bool Sorters::descend(const int& a, const int& b)`
- `void Vector::sort(bool(*predicate)(const int& a, const int& b))`