

C++ gyakorló feladat – fájlkezelés/tárolók használata

- Készítsünk egy Exam osztályt, ami egy megírt dolgozat eredményeit fogja tárolni:
 - A tantárgyat (string)
 - A dolgozat megírásának helye (string)
 - A dolgozat eredményei (sorrendi tároló int típussal)

Legyen getter a tárgyhoz és a helyhez.

Egy addGrade függvény segítségével lehessen új jegyet adni az osztályban lévő tárolóhoz,

- Készítsünk az osztályhoz alapvető statisztikai függvényeket, amik bizonyos értékeket számolnak ki és adnak vissza:
 - átlag (avg)
 - medián (median)
 - módusz (modus)
 - szórás (deviation),
- Készítsünk egy Date osztályt, ami egy dátumot tárol (év, hónap, nap). Minden adathoz legyen getter,
- Készítsünk egy tároló osztályt (Calendar), ami Exam példányokat tárol. A tárolás ne sorrendi tárolóval történjen. Az osztály dátum-vizsga párokat tárol, a megírás napja (Date) alapján azonosítva. Feltételezzük, hogy minden napon csak egy dolgozatot írtak. Legyen egy addExam függvénye, amely megkap egy dátumot és egy vizsgát, és eltárolja,
- Legyen az Calendar osztálynak egy print függvénye, ami megjeleníti a következő adatokat minden dolgozatról, dátum szerint növekvő sorrendbe rendezve: Dátum, tantárgy, helyszín, átlag, medián, módusz, szórás,
- Legyen az Calendar osztálynak egy printAvgOrder függvénye, ami szintén megjeleníti a dolgozatok adatait, de ezúttal az átlag szerint növekvő sorrendbe rendezve,
 - Nem szabad feltételezni, hogy minden átlag különböző, lehetnek egyformák is.
 - Tipp: tárold a belső tároló iterátorait egy új listába, írd egy külön < operátort hozzá, ami két iterátor összehasonlítja az átlag alapján, majd sort(),
- Az Exam objektumokat egy szöveges fájlból olvassuk be, amelynek minden sora egy-egy vizsga adatait tárolja a következő formátumban (az adatok formátumát nem kell ellenőrizni):
 - subject place year month day grad1 grade2 ... gradeN -1
 - pl.: korszprog PC0 2016 9 26 4 5 5 4 3 4 4 4 3 5 4 5 4 4 -1