

10. heti feladatok

1 Készítsen egy **OrderedItemsHandler** osztályt! Az osztály adattagként olyan elemek tömbjét tárolja el, amelyek megvalósítják az **IComparable** interfészt. A tömböt az osztály konstruktorán keresztül lehessen inicializálni, és a konstruálást követően ne lehessen az osztályon kívülről módosítani.

Az osztály publikus metódusai valósítsák meg a következő funkcionálisokat:

- Lehessen megvizsgálni, hogy rendezett-e a tömb a `bool IsOrdered(bool isAscending = true)` metódus használatával. Az `isAscending` „flag” értékétől függően növekvő, illetve csökkenő rendezettséget vizsgáljon a metódus.
- Lehessen rendezni a tömböt a `void Sort(SortingMethod sortingMethod, bool isAscending = true)` metódus használatával. A `SortingMethod` egy olyan enum típus legyen, amely három értékkel rendelkezik: `Selection`¹, `Bubble`² és `Insertion`³. Ha a rendezéseket csak növekvő módon rendezett tömbökre kívánja implementálni, akkor érdemes készíteni egy privát `void Reverse()` metódust, amely képes megfordítani a tömbbeli elemek sorrendjét.
- Bináris kereséssel lehessen eldönteni, hogy egy adott érték benne van-e a tömbben és ha igen, akkor adja vissza a keresett elemet. Ez a funkcionalitás is működjön növekvő, illetve csökkenő módon rendezett tömbök esetén is. Ha a tömb nem rendezett az elvárt módon, akkor a metódus dobjon el egy **NotOrderedItems** kivétel objektumot, ami tartalmazza a tömböt is.
- A bináris keresést implementálja iteratív és rekurzív módon is!
- Növekvően rendezett tömbök esetén lehessen megkeresni a legkisebb olyan indexet a tömbben, amely indexnél található elem értéke *nem kisebb* egy keresett értéknél. Ha a tömb nem növekvő módon rendezett, akkor dobjon el egy **NotOrderedItems** kivételt.
- Növekvően rendezett tömbök esetén lehessen megkeresni a legkisebb olyan indexet a tömbben, amely indexnél található elem értéke *nagyobb* egy keresett értéknél. Ha a tömb nem növekvő módon rendezett, akkor dobjon el egy **NotOrderedItems** kivételt.
- Növekvően rendezett tömbben lehessen meghatározni azon elemek darabszámát, amelyek egy adott értékkel egyenlőek.
- Növekvően rendezett tömbben lehessen meghatározni azon elemek darabszámát, amelyek egy megadott tartományba esnek.
- Növekvően rendezett tömbből lehessen kinyerni az összes olyan elemet, amelyek egy megadott tartományba esnek.

Minden publikus metódushoz készítsen unit teszteket, amelyekkel az összes lehetséges határeset le tudja fedni.

Készítsen egy **PhoneBookItem** osztályt, mely publikus auto-property-k segítségével eltárolja emberek nevét és telefonszámát. Az osztály valósítsa meg az **IComparable** interfészt. A `int CompareTo(object other)`

¹Minimumkiválasztásos rendezés

²Javított buborékredezés

³Javított beillesztéses rendezés

metódust implementálja úgy, hogy az osztály elemeit lehessen név szerint rendezni. A metódus `other` paramétere fogadjon **PhoneBookItem** és **string** típusú bemenetet is, egyéb esetben dobjon el **ArgumentException**-t.

A főprogramban készítsen egy „telefonkönyv” tömböt, amellyel példányosítson egy **OrderedItemsHandler** objektumot. Ellenőrizze le, hogy az implementált funkcionalitások helyesen működnek-e a telefonkönyv elemeivel.