

11. Hőmérséklet osztály jegyzőkönyv

Korszerű számítástechnikai módszerek a fizikában

Szabó Marcell

2021.05.05

1 Bevezetés

Ebben a dokumentumban ismertetem a címben említett tárgy keretein belül készített projekttemet. A beadandót C++ nyelven írtam, témája ugyancsak a címben található. Céлом az volt, hogy egy olyan hőmérséklet osztályt írjak, amely tudja tárolni és kezelni a különböző (Celsius, Fahrenheit, Kelvin, Rankine, Delisle, Römer) mértékegységekben megadott hőmérsékletetket. Ezen kívül képes összeadni, kivonni egymásból, valamint sorba rendezni őket.

2 A program funkciói, a kódok magyarázata

2.1 A main.cpp fájl

Az elkészített hőmérséklet osztályomat más szükséges headerekkel egyetemben beimportáltam. A "main" függvényben létrehoztam kettő változót, amelyek segítségével megmutattam osztályom működését.

Kiírtattam őket, először átváltottam az egyiket és vissza, majd egy harmadik változót is kiírtattam, ami az összegüket, később a különbségüket tartalmazta.

Ezek után az összeadás értékadás és a különbségképzés értékadás műveleteket teszteltem.

Végül létrehoztam egy vektort, amely az osztályom elemeit tartalmazta. A sorbarendező függvényemmel sorbarendeztam az elemeit egy új vektorba, majd kiírtattam az eredményt. Az az algorithm könyvtár segítségével is megoldottam a sorbarendezés feladatát. Ez a módszer a meglévő vektort rendezi sorba.

2.2 A temperature.h fájl

Ez egy úgynevezett header fájl. Az itt megírt (és beimportált) osztályt, függvényeket a main programban úgy tudtam használni, hogy ezt a headert töltöttem be oda. Ezzel a módszerrel átláthatóbb, struktúráltabbá válik a kód.

Az elején betöltöttem a vector-t, hiszen szükségem volt rá a későbbiekben.

2.2.1 Temperature osztály alapok

Egy hőmérséklet adatot két változóval célszerű leírni. Az egyik az értéke, ez egy szám. A másik a mértékegysége, ez pedig szöveg típusú. Számból sok fajta van, egész, tört, stb., ezért ezen változót sablonosítottam, tehát nem adtam meg konkrét típust neki, hanem ez egy objektum létrehozásakor megadott érték szerint fog eldőlni. A string típusú változót meghagytam stringnek.

2.2.2 Temperature osztály saját függvényei

Az programnak át kellett tudnia váltani a hőmérsékleteket egymásba (a műveletek elvégzéséhez meg amúgy is). Ezt két lépésben oldottam meg.

1. `convert_to_C` függvény segítségével Celsius fokba váltottam át az összes hőmérsékletet. Egy `if-else` módszerrel végimentem az osztály által számontartott mértékegységeken, és az objektum mértékegységének megfelelő átváltással megkaptam a hőmérséklet C belı értékét.
 2. A `convert_from_C` függvénnyel a C-beli értéket átváltottam a kívánt mértékegységűre hasonló módszerrel.
- A `convert` függvényben egyesíttem a fentieket. Így összesen a 6 mértékegységre 12 lehetőségből megvalósult az átváltás, különben jóval bonyolultabban 21 lépébből lehetett volna.

2.2.3 Temperature osztály saját műveletei

Ezek olyan speciális függvények, amelyek egyszerű műveleteket végeznek.

- Az `operator+=` az összeadás és értékadásért felelős, a hozzáadandó értéket (pontosabban egy vele egyenlő segédváltozót) átváltja a művelet bal oldalán szereplő érték mértékegységére, majd hozzáadja. Ezzel az eredeti objektumot módosítottuk, nem egy újat hoztunk létre. Ez különbözteti meg a "sima" összeadástól, ezért is az osztályon belülre került.
- Az `operator=` kivonással csinálja meg ugyan azt, mint a felette lévő.
- Az `operator<` a hőmérséklet osztályba tartozó objektumokat tartalmazó vektor sorbarendezéséhez szükséges, így tudja eldönteni a program két hőmérséklet értékről, hogy melyik a nagyobb. Mindkét hőmérsékletet Celsius skálára konvertálva hasonlítja őket össze.
- Az `operator>` ugyanazt a funkciót látja el, mint az őt megelőző operátor, csak ellentétes esetben. Attól függően használja a program az egyiket vagy a másikat, hogy a vektort milyen sorrendbe kívánjuk rendezni.

2.2.4 Osztályon kívüli függvények

Itt olyan függvényeket definiáltam, amelyek ugyan kötődnek az osztályhoz, olyan objektumok a bemenő és kimenő értékeik is, viszont nem magát az objektumot módosítják, hanem egy újat hoznak létre.

- A `convert_to_C1` szinte identikus a felül említett társával, csak abban különbözik, amiben az osztályon kívüli függvények különböznek a belül definiáltaktól.
- A `convert_from_C1` és a
- `convert1` függvényre is ugyanez elmondható

Azért volt fontos definiálni ezeket osztályon kívül is, mert így is működnie kell a programnak, hogy nem az eredeti értéket változtatja.

- A `sort` függvény egy kicsivel komplexebb a többinél. Egy hőmérséklet objektumokból álló vektort kap meg, amit sorbarendez. A sorrendet egy `bool` típusú `order` argumentummal lehet szabályozni, igaz értékre növekvőbe, hamis értékre csökkenő sorrendbe rendezi az elemeket. Magát a sorba állítást egy egyszerű rendezéssel oldottam meg, felhasználva a fentebb definiált `<`, `>` műveleteket. A függvény az eredeti vektor másolatát adja vissza a kért sorrendbe rendezve.

2.2.5 Osztályon kívüli műveletek

Ezek az operátorok hasonlóképp működnek az osztályon belüliekhez, viszont itt a visszatérési érték egy új objektum, nem az egyik meglévőhöz adja hozzá/vonja ki a másikat. Így az osztályon kívülre kerültek. Ebben az esetben nem egyértelmű, hogy milyen mértékegységű legyen a kimenet. Én aszerint jártam el, hogy az első érték mértékegysége a domináns, de lehetett volna a másodiké, vagy akár egy fix default, mondjuk Celsius fok is.

- Az operator+ hasonlóan az eddigiekhez, átváltja a második objektum másolatát az első mértékegységére, összeadja a kettőt és ezúttal egy új objektumban tárolja és adja vissza az értéket.
- Az operator- kivonással hajtja végre a felette lévőnél leírtakat.

3 Diszkusszió

A jegyzőkönyvben bemutatott programmal a hőmérséklet osztályt és használatát igyekeztem megvalósítani, remélhetőleg sikeresen. Az osztály segítségével lehetőség nyílik hőmérsékleteket tárolni, összeadni, kivonni illetve több hőmérsékletet sorba rendezni. A rendezést kétféleképpen is végre lehet hajtani, természetesen mindkét irányba. Még lehetne továbbfejleszteni a programot attól függően, hogy mi a célunk vele. Például lehetne átlagot számolni, vagy akármi mást.

Összességében egy érdekes, hasznos projektnek mondanám, hatására valamelyest tisztázódtak bennem az osztályokkal kapcsolatos ismertek alapjai, tanultam belőle új dolgokat.