

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1 String kiegészítés
 - ◆ 1.2 Jegyek
 - ◆ 1.3 Rendezett jegyek

Feladatok

Minden feladathoz nyiss új projektet IDE-ben vagy írd új file-ba ha parancssorból dolgozol!

Mostantól .cpp kiterjesztésű file-okban dolgozzunk!

String kiegészítés

Egészítsük ki a múltórai **String** osztályt az **operator=** és **operator==** operátorokkal. Az első új értéket ad egy már létező String-nek, a második összehasonlít két String-et.

Pl:

```
String s1 = String("batman");
String s2 = String("catman");
cout << (s1 == s2) << endl;
s1 = s2;
cout << (s1 == s2) << endl;
```

Ne felejtjük el, hogy vigyázni kell az **operator=-**nél, a következő ne okozzon hibát:

```
String s1 = String("batman");
s1 = s1;
cout << s1 << endl;
```

Megoldhatjuk azt is, hogy ne csak String-ekkel működjenek, hanem a szokásos C string-ekkel is:

```
String s1 = String("batman");
String s2 = String("catman");
cout << (s1 == "batman") << endl;
s1 = "catman";
cout << (s1 == s2) << endl;
```

Jegyek

Írjunk **Pontszamok** nevű osztályt, mely hallgatók pontszámait tartalmazza egy adott tárgyból. Tároljon nevet (vagy neptunkód ekvivalenst) és a pontszámot, ami 0 és 100 közötti egész szám.

- Minden hallgatóhoz 1 pontszám tartozik.
- Feltételezhetjük, hogy 100-nál nem lesz több hallgató.
- Amikor létrehozunk egy **Pontszamok** objektumot, az legyen üres.
- Legyen metódus ami új hallgatót és a hozzá tartozó pontszámot veszi fel.
- Legyen metódus ami visszaadja, hogy jelenleg hány pontszámot tárolunk.
- Legyen metódus ami kiszámolja és visszaadja a tárolt pontszámok átlagát (float-ként).

- Lehessen **operator**[-] -al elkérni az i-edik hallgató nevét, ha i egész szám (mindegy milyen sorrendben, berakási sorrend is jó). Ezt String-ként adja vissza.
- Lehessen **operator**[-] -al elkérni a kapott s String nevű hallgató pontszámát.

Pl:

```
Pontszamok pontok;
pontok.add("Andras", 56);
pontok.add("Aladar", 22);
pontok.add("Anita", 71);
pontok.add("Andrea", 34);
pontok.add("Aniko", 64);
for(int i = 0; i < pontok.hossz(); i++) {
    cout << pontok[i] << " : " << pontok[pontok[i]] << endl;
}
cout << "atlag: " << pontok.atlag() << endl;
```

Rendezett jegyek

Módosítsuk a fenti osztályt, hogy amikor elkérjük **operator**[-] -al és egy egész számmal az elemeket, akkor pontszámok szerinti csökkenő sorrendben adja vissza (azaz pontok[0] a legmagasabb pontút adja vissza).

- Írjuk még meg a median metódust ami a pontszámok mediánját adja vissza.
- Írjuk meg továbbá, hogy cout-al kényelmesen ki lehessen írni. Ekkor sorrol sorra pontszám szerinti csökkenő sorrendben kiírja a hallgató nevét és pontszámát.
- Egészítsük ki egy új adattaggal az osztályt: minimum. Azt tárolja, hogy hány pont kell ahhoz, hogy sikeresen teljesítse a tárgyat a hallgató. Ezt lehessen beállítani amikor létrehozzuk az Pontszamok objektumot. Alapvetően állítsuk 40-re.
- Írjunk egy **atmentek** metódust, ami visszaadja egy String tömbben azon hallgatók nevét akik átmentek a tárgyon (magasabb vagy egyenlő a pontjuk a minimummal).
- Módosítsuk a cout-os kiírást úgy, hogy tegyen egy vonalat az átmentek és nem átmentek közé (vonal pl: "http://wiki.math.bme.hu-----"http://wiki.math.bme.hu).