

# Programozás II. Gyakorló Feladat

SZTE Szoftverfejlesztés Tanszék

2025. ősz

## Ismertető

- A programot C++ nyelven kell megírni.
- **A benyújtandó fájl neve kötelezően feladat.cpp.**
- A megoldást a *Bíró* fogja kiértékelni.
  - A Feladat beadása felületen a Feltöltés gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a *Bíró* a programot g++ fordítóval és a  
-std=c++20 -static -O2 -DTEST\_BIRO=1 -Wall -Werror  
paraméterezéssel fordítja és különböző tesztesetekre futtatja.
- **A Bíró fordítási hibával nulla pontot fog adni minden -Wall kapcsoló által jelzett warningért!**
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 5 másodpercnél és hiba nélkül (0 hibakóddal) fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:
  - Futási hiba 6: Memória- vagy időkorlát túllépés.
  - Futási hiba 8: Lebegőpontos hiba, például nullával való osztás.
  - Futási hiba 11: Memória-hozzáférési probléma, pl. tömb-túlinde克斯, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők az alábbi módon:
  1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
  2. A kapott url formátuma:  
`https://biro2.inf.u-szeged.hu/Hallg/IBL302g-1/1/hXXXXXX/4/riport.txt`
  3. Az url-ből visszatörölve a 4-esig (`riport.txt` törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 20 alkalommal lehet benyújtani, a megadott határidőig.
- A programban szerepelhet `main` függvény, amely a pontszámításkor nem lesz figyelembe véve. Azonban ha fordítási hibát okozó kód van benne az egész feladatsor 0 pontos lesz.
- A megvalósított függvények semmit se írnak ki a standard outputra!

# Feladatsor

Néhány feladat megoldásához szükséges megvalósítani **az előző gyakorló feladatsor Telepes** osztályát vagy legalábbis annak egy minimalizált verzióját: legyenek benne a kívánt adattagok (1-es táblázat), getterek, setterek és a default konstruktor.

Adattag neve	Típusa	Jelentése	Getter neve	Setter neve
nev	std::string	A telepes neve	getNev	setNev
szulBolygo	std::string	Születési bolygó	getSzulBolygo	setSzulBolygo
bolygo	std::string	Jelenlegi bolygó	getBolygo	setBolygo
ero	unsigned	Munkavégző képesség	getEro	setEro

1. táblázat. Telepes adattagok. Default értékek: üres sztringek illetve az ero adattag esetén 1

## Bemelegítő feladat

Bővítsd ki a Telepes osztályt egy új, privát láthatóságú adattaggal! A default értéke false legyen. Inicializálja ezt is a default konstruktor!

Adattag neve	Típusa	Jelentése	Getter neve	Setter neve
vegan	bool	Vegán-e a telepes	isVegan	setVegan

2. táblázat. Új telepes adattag

## 1. feladat

Készíts egy Kolonia nevű osztályt, mely a telepesek által létesített kommunákat reprezentálja. Gazdasági okok miatt minden kolónia csak 25 telepest enged befogadni. Minden kolóniáról tárold el:

- A benne lakó telepeseket
- A kolónia nevét (sztring, legyen lekérdezhető a getNev getter segítségével)
- A kolóniának helyet adó bolygó nevét (sztring, legyen lekérdezhető a getBolygo getter segítségével)
- Tudjuk lekérdezni a kolóniában lakó telepesek számát a getLetszam, unsigned értéket visszaadó függvénnyel!

Készítsd el a Kolonia osztály konstruktorait is!

- Legyen egy konstruktora, mely két string paraméterrel rendelkezik. Az első sztring alapján a kolónia neve, a második alapján a kolónia bolygója legyen beállítva.
- Legyen lehetőség default konstruálásra is, mely során a kolónia nevét és a bolygóját üres sztringre kell állítani.

## 2. feladat

Valósítsd meg a `letelepszik()` függvényt, mely egy telepest vár paraméterül és letárolja azt a kolóniában, ha még elfér

```
Kolonia k;  
Telepes t;  
k.letelepszik(t);
```

A telepes úgy legyen letárolva, hogy a `bolygo` adattagja legyen beállítva a kolónia `bolygo` adattagjára. Ha a 25 fős limit miatt már nem lehet újabb telepest a kolóniához adni, akkor a módszer pontosan ezt a sztringet írja ki a standard outputra: „A kolónia megtelt”. A kiíratást sortörés kövesse.

## 3. feladat

Készíts egy globális függvényt a `Kolonia` osztályon kívül, amely megvizsgálja, hogy a kolónia vegán-e, azaz többségben vannak-e a vegán életfelfogású lakók. A neve legyen `veganE`, egy `Kolonia` objektumot vár paraméterül. A függvény akkor ad vissza igaz értéket, ha kolónia lakóinak több, mint 50%-a vegán (a `vegan` adattag értéke igaz). Ha nincs egy lakó sem a kolóniában, akkor alapértelmezetten nem vegán (azaz `false` a függvény visszatérési értéke).

## 4. feladat

Fejleszd tovább a 2-es feladatban létrehozott `letelepszik()` függvényt! Az új telepes letárolása előtt azt is vizsgáld meg, hogy a telepes vegán orientációja egyezik-e a kolónia vegán orientációjával. Ha a kolónia vegán, akkor az új telepes letárolása csak akkor lehetséges, ha ő is vegán. Amennyiben nem vegán az alábbi hibaüzenet kerüljön a standard outputra (utána sortörés következzen): „A kolónia vegán”.

Figyelem, a `veganE` használatához a függvény és/vagy a `Kolonia` osztály forward deklarációja szükséges. Linkek itt és itt. Alternatíva még a `friend` függvény használat.



## 5. feladat

Készíts egy `nevsor()` nevű függvényt a Kolónia osztályba, amely lexikografikusan rendezve kiírja a standard outputra a kolóniában élő telepesek neveit. Minden telepes neve után legyen egy sortörés. Ha névismétlődés lenne, akkor nem kell duplán feltüntetni a nevet. Példa:

A kolóniábanlévő telepesek: *Jutka, Juli, Julia, Janos, Jutka*

Elvárt kimenet:

Janos

Juli

Julia

Jutka

## 6. feladat

Keressük meg a legerősebb telepest a kolóniában! Legyen egy `legerosebb` függvény a kolónia osztályon belül, mely visszaad egy unsigned értéket: a legerősebb telepes erő értékét vagy 0-át ha nincs telepes még a kolóniában.

## 7. feladat

Készíts egy `szozol()` metódust a Kolónia osztályon belül, amely egy sztringet vár paraméterül és visszaad egy egész számot. A sztring egy bolygó nevét tartalmazza és a függvénynek vissza kell adnia, hogy hány telepesnek a szülőbolygója egyenlő vele. Két bolygó neve egyenlőnek számít, ha csak kisbetű/nagybetű eltérés van a nevükben (pl. „Mars” és „mars” egyenlők). A bolygók nevei ascii karakterek (tehát nem tartalmazznak pl. ékezetes betűket). Ha mindkét adattag értéke üres sztring egyenlőnek tekinthető.

## 8. feladat

Készíts egy `eroHisztogram()` nevű függvényt, ami egy `map<unsigned, unsigned>` típusal tér vissza. A map-ben össze van számolva, hogy a kolóniában előforduló erőértékekkel hány telepes rendelkezik. Példa:

Telepesek erőértékei a kolóniában: 5, 2, 1, 1, 5, 5

A visszaadott map tartalma: 1: 2, 2: 1, 5: 3